Une approche statistique multi-échelle au recalage rigide de surfaces : Application à l'implantologie dentaire

THÈSE

présentée et soutenue publiquement le 7 avril 2003

pour l'obtention du

Doctorat de l'École des Mines de Paris

(spécialité Informatique temps-réel, robotique, automatique)

par

Sébastien GRANGER

Directeur de thèse : Nicholas AYACHE

Co-Directeur de thèse : Xavier PENNEC

Composition du jury

Directeur : Nicholas Ayache

Rapporteurs : Isabelle Bloch
Radu Horaud

Examinateurs : Xavier Pennec
Raymond Derycke
Yves Rouchaleau

Institut National de Recherche en Informatique et Automatique

Mis en page avec la classe thloria.

Remerciements

Pour leur aide professionnelle, ...

AREALL (Derycke, Stankoff, Ju, Francis & Eric, les Loos, Fred & Jean-Mi, Franck, Richard, ...)

EPIDAURE & CHIR (Xav, Alexis, Ronan, JoJo, Herve, Janet, Gerald Bianchi, Celine, ...)

LES AUTRES INRIA (David CS, Kornprobst, Morvan, le SEMIR et le self)

LES MINEURS (Rouchaleau, Laurgeau, ...)

LES AUTRES (Médioni, Mi-Suen Lee, Zhang, ...)

LES MACHINES (pasteur, sabin & banting)

LES RELECTEURS

LES MEMBRES DU JURY

Pour leur aide personnelle, ...

Max,

La famille, les proches, Jessica pour leur amour,

Mes guitares,

Mes cartes,

Merci a Fred de m'avoir fait pensé a remercier mon chat,

L'octo, Steph et les Nicois, Alexis, Nico, Bingo et les Kinetou, Seb & Natacha, JoJo, Julien, David, Yan et Laurent pour leur amitie,

Laetitia, Coco, Natacha, Flo, Frède, Emma, Isabelle, Michaela, Alix et Céline pour leur joli sourire,

Babar, Fred & Loic pour leurs bières,

Les zicos, les grimpeurs, les skieurs, les randonneurs et les rencontres d'un soir,

L'AGOS, Fabrice, Fanny et Martine pour leur soutien pendant ma difficile deuxième année,

TMC & Maguelone Hédon,

et tous les autres (pardon de les avoir oublié) qui m'ont aidé à devenir docteur[Nicolau, 2001].

Table des matières

Reste à faire :

Introduction : source des images - Complément sur les projets concurrents

Divers : finir notations, glossaire et index

Planning :

Délais imposés par les Mines :

Proposition de jury : 3 semaines avant la soutenance

Remise des rapports à tous les membres du jury : 12 jours avant la soutenance

Calendrier prévisionnel :

Semaine 4-5 (20-31 janv) : AREALL - Envoi aux rapporteurs

Semaine 6-7 (3-14 fev) : Finition manuscrit (chapitre 8, annexes, index)

Semaines 8-9 (17-28 fev) : AREALL

Semaine 9 (24-28 fev) : Reception rapports - Préparation soutenance - Proposition de jury

Semaines 14 (3-7 mars) : Préparation soutenance - Correction manuscrit

Mi mars : Soutenance

Pages couleurs :
Ch1:
4, 5, 6, 9, 10, 12, 15, 16, 17
Ch3:
60, 68
Ch4:
79, 80
Ch5:
116
Ch6:
123, 129, 130
Ch7:
161, 163, 180, 188, 189, 198,
Ch 8:
227, 231, 236, 237, 239, 241, 257
Annexe D:
299, 311
Annexe F:
340, 342, 343, 344, 344,

Chapitre 1

Introduction

Sommaire

1.1 Introduction

L'objet de ce chapitre est de motiver le travail de cette thèse. Il commence par présenter l'environnement de travail (Epidaure et AREALL, section 1.2). Vient ensuite une brève présentation de l'implantologie dentaire, et notamment de l'implantologie assistée par ordinateur (section 1.3), qui permettra au lecteur de se familiariser avec ce domaine. Nous nous focaliserons ensuite sur le projet VirtualScope, système de guidage per-opératoire pou le percement des axes des implants dentaires développé par AREALL, et plus précisément sur la partie recalage rigide qui est le coeur de ce travail de thèse (section 1.4). La section 1.5 présentera alors les données sur lesquelles sont basées ce recalage. Nous justifierons enfin le choix de l'ICP, algorithme de recalage basé sur les points, comme algorithme de base pour notre travail (section 1.6). Nous donnerons enfin une vue d'ensemble du contenu des différents chapitres de cette thèse (section 1.7) et des différentes contributions que l'on peut y trouver (section 1.8).

1.2 Epidaure et AREALL

Cette thèse s'est déroulée dans le cadre d'une bourse CIFRE. Ces bourses, subventionnées par l'ANRT, favorisent le transfert technologique, par le biais du travail du doctorant, salarié d'une entreprise mais détaché au sein d'un laboratoire de recherche. Ceci est particulièrement intéressant pour le doctorant, qui en côtoyant ces deux mondes, apprend leurs logiques très différentes et reçoit une double motivation. Ainsi, ma thèse s'est déroulée au sein du projet Epidaure, laboratoire de l'INRIA, pour le compte d'AREALL.

Epidaure & l'INRIA Le projet EPIDAURE de l'INRIA a pour vocation, en temps que laboratoire de recherche, de créer, développer et étudier de nouvelles méthodologies dans le domaine du traitement informatisé de l'imagerie médicale, indépendamment de toute logique commerciale. Ses domaines de prédilection sont la segmentation et le recalage d'images, l'analyse de séquences d'images et la simulation d'organes.

Ses membres m'ont aidé à développer des méthodes innovantes, à m'intéresser aux aspects théoriques. Xavier Pennec, qui m'a encadré, m'a ainsi encouragé à poursuivre son travail de thèse et m'a communiqué son savoir et ses points de vues sur le travail de chercheur.

AREALL La société AREALL, basée à Paris et Caen, est une petite entreprise (environ 20 employés, en majorité des ingénieurs) qui développe le système VirtualScope que nous présenterons un peu plus loin. Le développement de ce projet, commencé peu avant le début de ma thèse devrait s'achever par la validation clinique avec la fin de cette thèse.

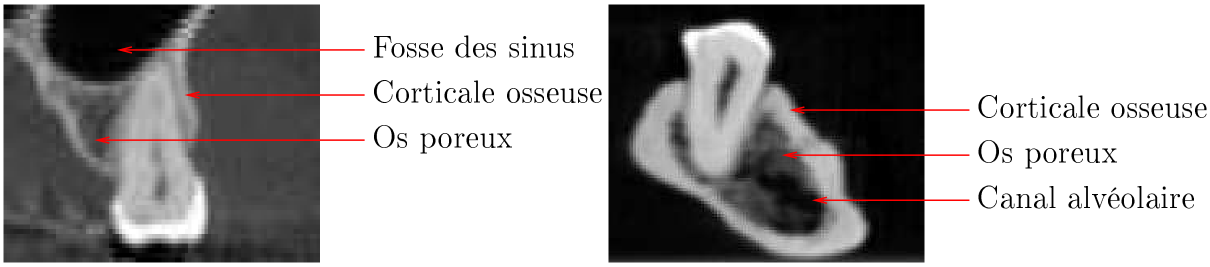


Fig. 1.1 – *Coupe sur un scanner X d'un maxillaire (à gauche) et d'une mandibule sèche (à droite)*

J'aurai ainsi pu me familiariser avec la développement d'un tel projet, et participé au travail des ingénieurs avec ses contraintes de délais et de parfaite fiabilité.

1.3 Implantologie dentaire assistée par ordinateur

Le projet VirtualScope est un système d'aide à la chirurgie pour la pose d'implants dentaires. Nous allons donc présenter ici rapidement l'anatomie dentaire, les différentes formes de prothèses dentaires, les implants et la chirurgie qu'ils impliquent. Nous nous contenterons des notions indispensables pour bien comprendre la problématique de cette thèse. Le lecteur souhaitant rentrer plus dans les détails pourra se reporter à [Dutreuil, 2001].

1.3.1 Éléments d'anatomie dentaire

L'homme possède deux dentitions (voir [Lautrou, 1998] pour une présentation détaillée) : la *maxillaire*, i.e. la mâchoire supérieure, qui fait partie du crane et la *mandibule*, i.e. la mâchoire inférieure, qui est indépendante. Chacune est formée de l'os de la mâchoire proprement dite et des dents. L'os est formé d'un pourtour extrêmement dur, la corticale osseuse, et d'un intérieur poreux, friable (voir figure 1.1). Les dents y sont fixées par leur racine. Les dents inférieures sont innervées par le nerf alvéolaire, qui passe dans le canal alvéolaire situé à proximité des racines des dents, à l'intérieur de l'os poreux. Les dents supérieures sont innervées par le nerf maxillaire, qui passe à proximité du nez, avant de se ramifier pour atteindre les racines des dents. Enfin, les sinus sont situés à proximité directe des dents supérieures.

1.3.2 Les prothèses dentaires

Les prothèse dentaires forment le type le plus ancien, le plus simple et le plus connu de dents artificielles. Ces systèmes reproduisent la partie visible de la dent et permettent

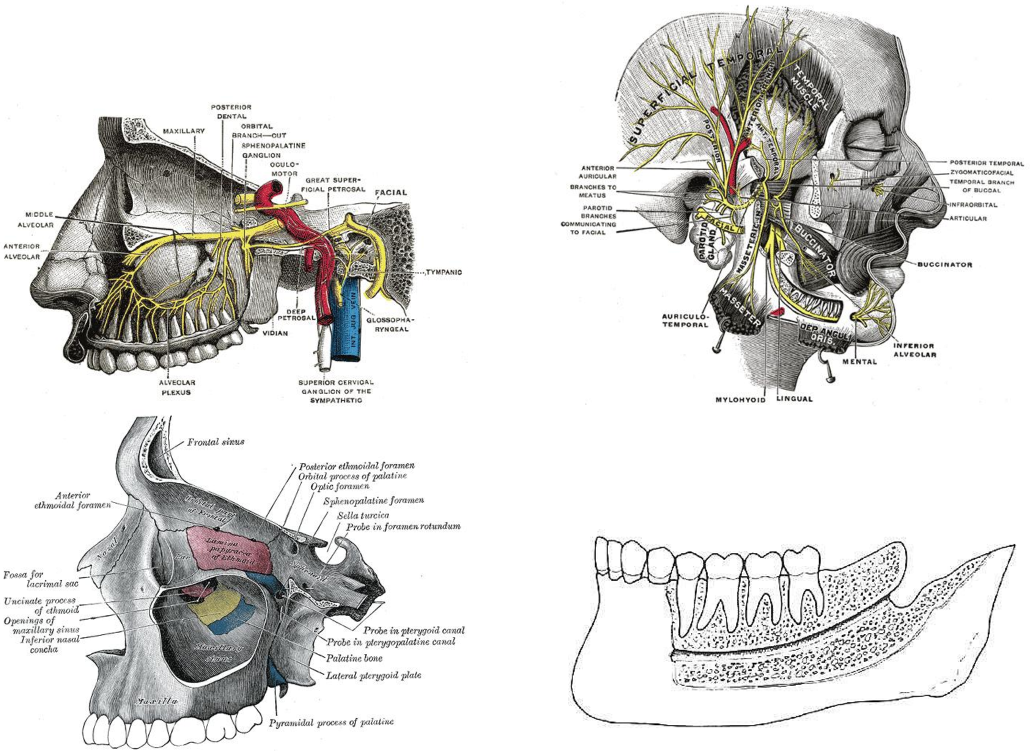


FIG. 1.2 – *Les zones à risques pour l'implantologie dentaire : les nerfs et sinus au-dessus du maxillaire (à gauche) et le nerf et son canal dans la mandibule (à droite)*

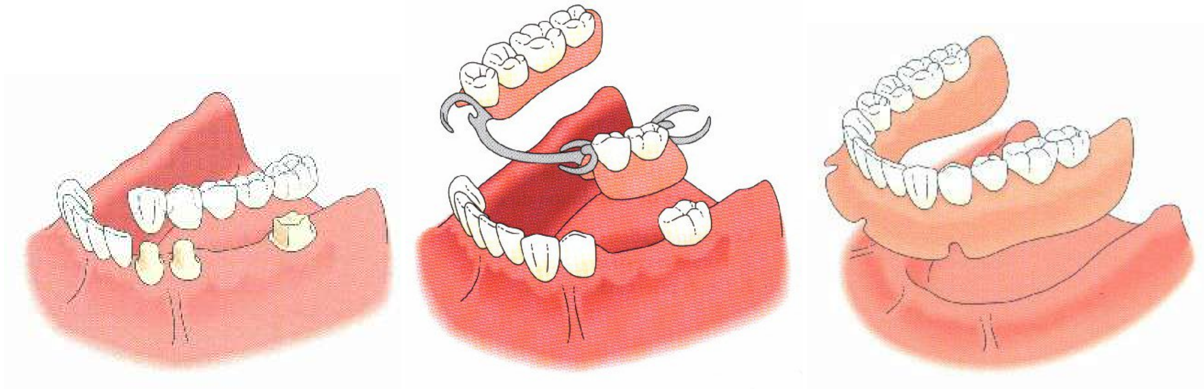


Fig. 1.3 – *Les 3 grands types de prothèses dentaires : de gauche à droite, Bridge, Stellite et Dentier*

ainsi de retrouver les fonctions de mastication et le sourire. On en distingue plusieurs grands types [Assémat-Tessandier and Amzalag, 1993] (voir figure 1.3) :

- Les bridges, qui reposent sur les dents restantes.
- Les stellites qui reposent sur la gencive et sont calées sur les dents restantes.
- Les dentiers, qui reposent directement sur la gencive, et conviennent aux édentations totales.

Ces prothèses sont relativement peu onéreuses, simples à poser, mais présentent les défauts suivants :

- Nécessité d'un entretien : ces prothèses doivent être entretenues avec soin. Il faut en particulier les enlever et les nettoyer tous les jours.
- Problème de la fonte osseuse : l'os de la mâchoire a tendance à fondre lorsqu'il n'est plus stimulé, entraînant certaines complications [Schnebelen, 1995]. Ceci arrive très souvent avec les prothèses, car elles ne reproduisent pas la racine de la dent et ne communiquent donc pas les efforts de mastication à l'os de la mâchoire.
- Manque de confort : le dentier est le moins confortable. En effet, il repose sur la gencive, qui est un tissu relativement mou, et n'est donc pas parfaitement fixé, entraînant une gêne lors de la mastication. Cette gêne s'aggrave lorsque l'épaisseur de la gencive augmente, suite à la fonte osseuse.

Pour pallier à ces défauts, un autre type de dents artificielles, basé sur un principe totalement différent, est apparu.

1.3.3 Les implants dentaires

Les implants dentaires modernes ont été mis au point par Bränemark, en Suède, à la fin des années 60 [Branemark *et al.*, 1969]. Ils ont permis, en reproduisant la structure complète de la dent, d'éliminer les défauts des prothèses. Ils sont cependant nettement

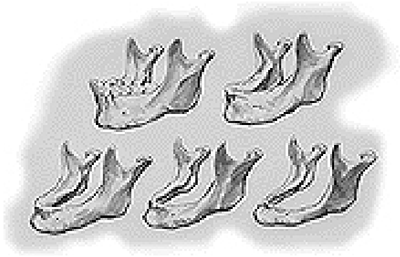


FIG. 1.4 – *Illustration de la fonte osseuse chez un patient totalement édenté*

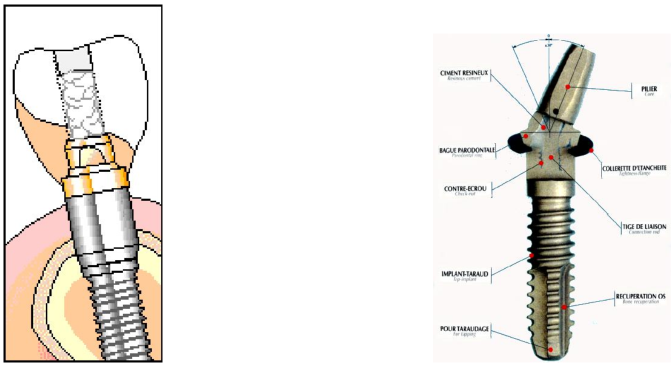


FIG. 1.5 – *Coupe schématique d'un implant dentaire (à gauche) et vue de la racine d'un implant (à droite)*

plus coûteux et plus difficiles à poser. Ils n'ont donc pas remplacé les prothèses, mais sont plutôt une alternative réservée à une population aisée.

L'implant est constitué, à l'image de la dent réelle, d'une racine solidement ancrée dans la mâchoire (voir figure 1.5), sur laquelle repose la partie apparente. Cette racine est une vis de titane, matériau qui possède une propriété fondamentale : l'osséo-intégration [Branemark *et al.*, 1985]. Ceci signifie que, une fois posée, l'os va se reformer autour de la vis pour épouser parfaitement ses contours. Ainsi, les risques de rejets sont pratiquement inexistants, et la solidité de l'implant est optimale.

1.3.4 Pose d'un implant

La racine de l'implant doit être vissée dans la mâchoire. Il faut donc au préalable creuser le trou ad-hoc. Le choix de l'axe n'est cependant pas chose facile [Assémat-Tessandier and Amzalag, 1993; Hobkirk and Havthoulas, 1998; Taylor, 1998]. Il doit en effet reposer sur deux galettes d'os dur, pour encaisser correctement les importants efforts de la mastication. Or l'os de la mâchoire est poreux à l'intérieur, et ne peut donc offrir une assise suffisante que lorsqu'il est traversé de part en part. Il faut de plus obligatoirement éviter le nerf dentaire qui passe à l'intérieur de l'os de la mâchoire inférieure, et les sinus situés juste au dessus de l'os de la mâchoire supérieure.

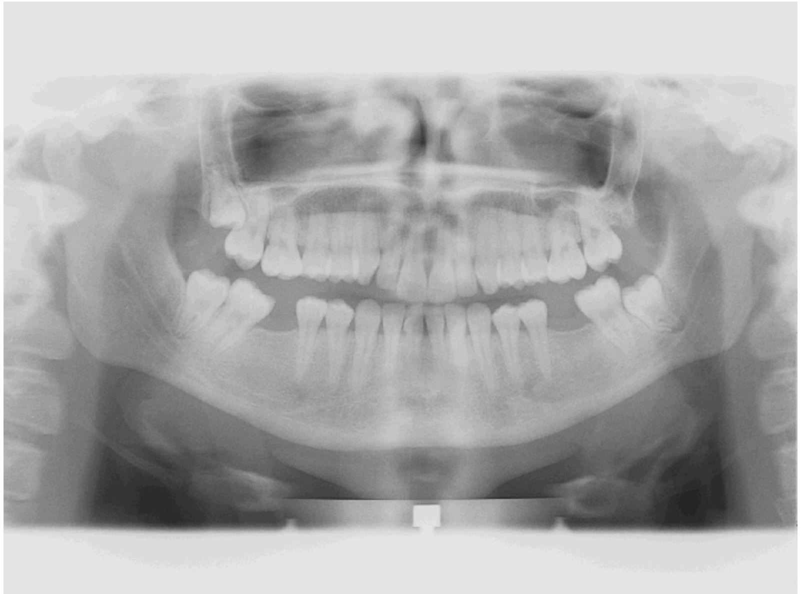


FIG. 1.6 – *Radio panoramique*

Le chirurgien-dentiste doit donc pouvoir situer l'os dur et les zones à risque avant d'opérer. Pour cela il utilise une radio panoramique (voir figure 1.6) dans les cas les plus simples, ou, en général, un scanner X (i.e une radio en 3 dimensions, voir figure 1.7) réalisé avant l'opération, sur lequel il peut préparer son geste en trouvant le meilleur axe pour l'implant [Abrahams and Kalyanpur, 1995; Bellaiche and Doyon, 1992; Rothman *et al.*, 1987; Vannier *et al.*, 1997; Jacobs and van Steenberghe, 1998; Quirynen *et al.*, 1990]. La pose de l'implant nécessite alors plusieurs étapes réparties sur plusieurs mois : la première, la seule véritablement lourde et difficile, est celle du forage de l'axe et de la pose de la racine de l'implant. Après cette opération et quelques visites de routines [1], il faut attendre quelques mois, temps nécessaire à l'os pour intégrer parfaitement la racine en titane, et le chirurgien peut enfin ré-ouvrir la gencive et remplacer la vis fixée provisoirement au sommet de la racine par la fausse dent (ou le dentier) définitive. Le patient doit au final effectuer en moyenne 8 visites réparties sur un peu moins d'un an.

1.3.5 Forage de l'axe

Pour pouvoir se situer sur la mâchoire du patient par rapport au scanner, l'oeil du chirurgien-dentiste est son seul secours, et il est pratiquement toujours obligé d'ouvrir la gencive pour voir affleurer l'os, et pouvoir ainsi ajuster son placement. L'axe est alors percé, puis la racine de l'implant est posée, et munie d'une vis provisoire. La gencive est suturée, et après une heure passée en salle stérile, le patient peut se relever. L'opération,

1. pour enlever les points de sutures de la gencive, réaliser des moulages des dents restantes, préparer les futures dents artificielles, essayer celles-ci, ...

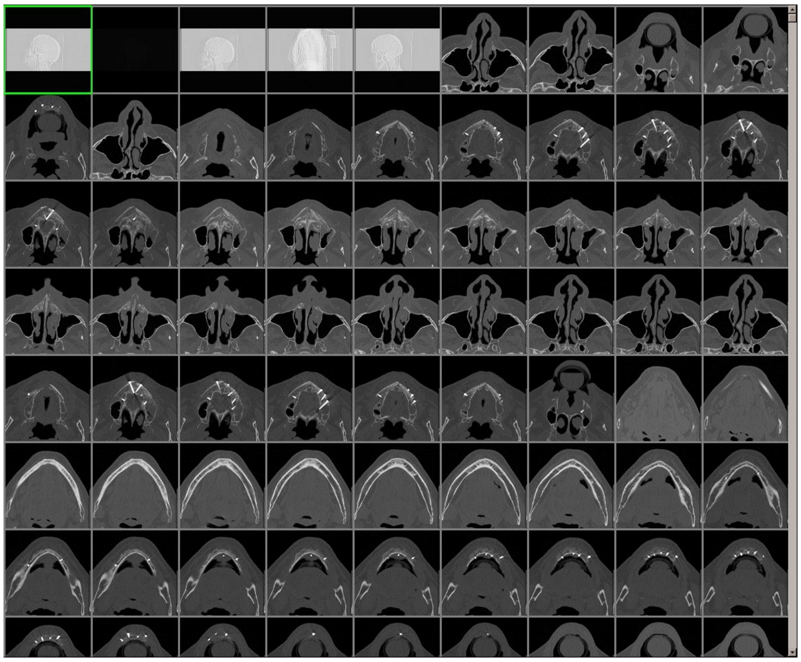


Fig. 1.7 – *Scanner d'un maxillaire (partie haute) et d'une mandibule (partie basse)*

relativement courte et simple, peut considérablement se compliquer en cas d'erreur de visée. Si un trou est mal percé, il faut en général compter plusieurs heures de travail...

Certains chirurgiens utilisent un guide chirurgical appelé *gouttière* [Hussaini and Canela Pichardo, 1997; Sicilia *et al.*, 1998; Fortin *et al.*, 1995] (voir figure 1.8). Celle-ci est réalisée de la manière suivante : on effectue dans un premier temps un moulage de la mâchoire du patient. On tire de ce moulage une réplique en plâtre de la mâchoire et son complémentaire, un moule en plastique qui s'adapte parfaitement sur la mâchoire. Le percement des axes se fait alors dans le moule en plastique replacé sur le plâtre. On place et on fixe des tubes dans les trous ainsi percés pour obtenir le guide chirurgical proprement-dit. Le chirurgien dentiste n'a alors plus qu'à le replacer sur la mâchoire du patient, et percer à l'intérieur des tubes. Une telle approche permet de ne pas avoir à ouvrir la gencive, et donc de raccourcir le temps de l'opération, et ainsi la quantité d'anesthésiants et le traumatisme opératoire. Elle ne peut en revanche être appliquée que lorsque le guide chirurgical peut être construit avec précision (i.e. dans les cas les plus simples, car il est moins facile de se repérer sur le plâtre que sur la véritable mâchoire) et lorsqu'elle ne rend pas le site opératoire inaccessible (ce qui est souvent le cas avec les molaires).

A l'heure actuelle, il semble que les taux de réussites de opérations d'implantologie soit de l'ordre de 95%, mais les résultats sont très variables en fonction de la position de la zone traitée et de la qualité de l'os de la mâchoire [Branemark *et al.*, 1977; Bass, 1991; Eckert and Wollan, 1998].

FIG. 1.8 – *Gouttière (guide chirurgical utilisé pour le forage de l'axe des implants dentaires) sur la copie plâtre d'une mâchoire*

1.3.6 Implantologie assistée par ordinateur

L'implantologie dentaire assistée par ordinateur [Abrahams and Kalyanpur, 1995] est apparue au début des années 90[2], et s'est focalisée sur le forage de l'axe. Elle a, dans un premier temps, offert des logiciels permettant de visualiser les scanners à l'écran. Les logiciels se sont alors enrichis de nombreuses fonctionnalités concernant :

- la visualisation des scanners en 2D : à la simple visualisation suivant les plans de coupes xy (les coupes horizontales, appelées coupes axiales), yz et xz dans le scanner, se sont ajoutés des fonctions plus évoluées, permettant par exemple l'interpolation de coupes ou de radios suivant un plan quelconque. Les dentistes utilisent en particulier le "plan panoramique" qui est vertical et longe "l'axe panoramique" (i.e. l'axe[3] médian de la mâchoire dans une coupe axiale) et les coupes sagittales définies comme les coupes verticales et perpendiculaires à l'axe panoramique[4].
- la visualisation des scanners en 3D : elle a commencé par le rendu surfacique, dans des logiciels permettant de segmenter la surface des différents tissus visibles dans le scanner, et de les visualiser en 3D. Les logiciels les plus récents permettent aussi un rendu volumique. Précisons cependant que les chirurgiens préfèrent le travail en 2D, car il permet un placement précis, et n'utilisent en général la 3D que pour la vérification ou pour le choix de plan de coupes virtuels pour leur travail en 2D.
- l'implantologie virtuelle : elle est vite apparue pour répondre au véritable besoin du

2. Il existe à l'heure actuelle une demi-douzaine de logiciels différents, parmis lesquels Simplant et DentaPC [Dutreuil, 2001, Section 3.2.2.1].

3. Le terme axe est bien sûr abusif, car il ne s'agit pas d'une ligne droite. Dans les logiciels, il est en général modélisé par une spline plane.

4. Le terme de coupe sagittale est lui aussi un peu abusif, car il ne correspond pas à la notion classique de coupe sagittale, i.e. de coupe verticale d'avant en arrière.

FIG. 1.9 – *Captures d'écrans du logiciel de planification de VirtualScope, illustrant différentes fonctionnalités sur un scanner contenant beaucoup d'artefacts (d'où les traces informes en haut des dents).*
A gauche : visualisation des coupes panoramiques (en bas), des coupes sagittales (à droite) et du modèle surfacique 3D (en jaune la surface des dents, en rouge le canal du nerf dentaire).
A droite : implantologie virtuelle, avec en particulier la sélection de l'implant et la visualisation en 3D et 2D au voisinage de l'axe.

chirurgien : pouvoir placer efficacement son implant. On a donc rajouté des fonctions permettant le choix et le placement des implants, et on a fait apparaître ces implants virtuels dans la visualisation. On a ensuite adapté les fonctions d'interpolations de vues 2D pour examiner ce qui se passe au voisinage de l'implant. Enfin l'ordinateur s'est permis de donner son avis sur le placement de l'implant : soit en vérifiant que les zones à risques ne sont pas touchées (ce qu'un chirurgien sait déjà très bien faire), soit en effectuant un calcul d'effort par une méthode de type éléments finis, et vérifier ainsi la solidité de l'implant (ce qu'un chirurgien a beaucoup plus de mal à faire).

– la prothèse virtuelle : elle est venue s'ajouter à tous les éléments précédents pour permettre la conception assistée par ordinateur (CAO) des dents prothétiques, et leur fabrication par des machines outils numériques. Notons que la difficulté dans la conception de ces dents est le respect des contraintes d'occlusion, i.e. de l'emboîtement des deux mâchoires lorsque la bouche se ferme, qui exige un ajustement à 50 microns .

Mais toutes ces améliorations n'ont pas résolu le plus gros problème des chirurgiens : respecter lors de l'acte chirurgical le placement prévu sur le scanner. Ce problème de guidage chirurgical a reçu une première réponse avec des systèmes robotisés permettant la réalisation automatique des gouttières à partir de l'implantologie virtuelle. Plusieurs ont été mis au point [5], mais sont limités par les problèmes spécifiques des gouttières (nécessité d'avoir encore des dents, impossibilité d'atteindre les molaires) et par un certain manque de précision (dû à la fois aux méthodes de recalages employées et aux imperfections du replacement de la gouttière sur la mâchoire).

Aucun système de guidage chirurgical direct n'a encore été validé. Le projet VirtualScope est peut-être la tentative la plus avancée dans ce sens [6].

1.4 Le projet VirtualScope

Le projet VirtualScope de la société AREALL [Etienne and al., 2000] se donne pour but de guider le chirurgien dentiste pendant l'opération, sans utiliser de gouttière. Pour cela, la perceuse est fixée à un système capable de suivre la trajectoire de l'outil (un système optique, voir figure 1.10), et un ordinateur doit indiquer au praticien quelle est sa position par rapport à l'axe prévu, en faisant le lien entre les coordonnées des points de l'os de la mâchoire dans l'image scanner (où le chirurgien a placé son axe avant l'opération) et dans le repère du robot. L'os de la mâchoire étant indéformable, les deux systèmes de coordonnées sont reliés par une transformation rigide (une translation et une rotation), que nous devons estimer. Il s'agit donc d'un problème de recalage [7] rigide per-opératoire. Nous allons préciser ici les attentes auxquelles il doit répondre, et résumer ainsi le "cahier des charges" qui a guidé ce travail de thèse.

1.4.1 Résultat cherché

Il s'agit de la close la plus importante de notre cahier des charges. Le but du projet VirtualScope est d'atteindre une précision finale de l'ordre de 1 mm. Les sources d'imprécisions étant nombreuses et cumulatives (imprécision du robot passif, imprécision du système de suivi du mouvement du patient, ...), nous devons fournir une transformation rigide laissant une imprécision de l'ordre de 0.3 mm dans la zone de l'opération.

Cette précision dépendra surtout des données per-opératoires. L'idéal serait d'être capable de prévoir "en ligne" la précision du recalage en fonction de celle-ci. Ceci permettrait, si elle est insuffisante, de demander de refaire ou de continuer l'acquisition des données per-opératoires. La question se pose aussi de la liberté laissée au chirurgien dans la répartition des données per-opératoires : celles-ci doivent-elles avoir une forme particulière?

5. Notamment par Praxim (issu du TIMC à l'université de Grenoble) et Materialise [Dutreuil, 2001, Section 3.2.2.2].

6. Nous n'avons entendu parlé que d'un seul projet comparable, DenX [Shapira, 2002]. Celui-ci est basée sur un recalage de marqueurs et est donc intrinsèquement moins précis.

7. De manière générale, le recalage est la recherche de la transformation séparant deux acquisitions d'un même objet. On parlera de recalage multiple lorsque l'on dispose d'un nombre supérieur d'images [Bergevin *et al.*, 1996; Stoddart and Hilton, 1996]. Nous n'aborderons pas ce sujet dans cette thèse, et conseillons [Ben-Jemaa, 1998; Cunnington and Stoddart, 1999] au lecteur qui s'y intéresserait.

FIG. 1.10 – *Le système VirtualScope pour le guidage per-opératoire tel que présenté au salon de l'Association Dentaire Française. On distingue le système de suivi optique de l'outil (avec les caméras en haut de la potence et les cibles fixées à l'outil) et du patient (avec les cibles fixées à la mâchoire), et l'écran de contrôle.*

Si oui, cette forme doit-elle être donnée par une heuristique (e.g. “il vaut mieux prendre des points sur les zones caractéristiques que sur les zones plates”) ou par un calcul de prédiction “en ligne” ?

1.4.2 Travail pré-opératoire

Le travail avant l’opération pose peu de contraintes : peu ou pas de contraintes de temps de calcul, la possibilité de mettre le chirurgien à partie pour aider à initialiser le recalage et pour vérifier la qualité du résultat. Autrement dit, on s’autorise une certaine interactivité, limitée par les talents informatiques du praticien, et un certain manque de robustesse. Les contraintes sur la précision reste, quant-à-elles, toujours d’actualités.

1.4.3 Travail per-opératoire

Pendant l’opération, les contraintes deviennent importantes. Les mains du chirurgien ne peuvent toucher que des objets stérilisés. Il ne lui reste que les pieds pour commander, à l’aide de pédales, les appareils (perceuse, mais aussi fauteuil, et dans notre cas, ordinateur). Ceci limite grandement les possibilités d’interactivité.

Il sera aussi difficile d’effectuer une vérification du résultat. Il faut donc être parfaitement robuste : être capable d’indiquer au chirurgien lorsque le recalage a pu poser problème et peut se révéler mauvais, tout en s’efforçant de rendre ces problèmes aussi rares que possible.

Dernière contrainte, importante, celle du temps de calcul : il ne doit pas dépasser quelques minutes.

1.5 Données

Nous allons ici présenter les données utilisées pour faire les recalages. Toutes ces données ont été fournies par AREALL. Précisons que nous avons utilisé pour certaines expériences des données acquises sur des mâchoires sèches.

Il existe trois types de données : deux en pré-opératoire (le scanner bien-sûr, mais aussi une image 2D 1/2), et un en per-opératoire (des nuages de points).

1.5.1 Image 3D : Scanner X et segmentation

Le premier type d’image disponible, avant l’opération, est le scanner X sur lequel est réalisé la planification. Il s’agit d’une image volumique précise (la taille des voxels dans les images habituelles est de 0.25x0.25x0.5 mm) et peu bruitée.

Comme nous le verrons, le recalage est basé sur une technique purement géométrique, qui nécessite de segmenter l’os de la mâchoire dans la scanner. Nous avons donc mis au point une telle méthode de segmentation. Cette technique étant très classique, nous ne donnerons que son principe général.

Elle est basée sur l'algorithme du "Marching-Cube" [Lorensen and Cline, 1987], qui permet d'extraire une iso-surface dans une image volumique. On extrait en l'occurrence la surface où le laplacien de l'intensité est nul, i.e. la surface où la variation d'intensité est la plus grande. Il s'agit bien entendu de l'interface entre plusieurs tissus, et on isole la surface de l'os en se restreignant aux zones de l'image où l'intensité est compatible avec celle de l'os. Enfin, on ne garde que la plus grande partie connexe de l'ensemble ainsi obtenu pour se débarrasser des morceaux de surfaces parasites.

Les nuages ainsi obtenus contiennent 50.000 à 100.000 points reliés par des triangles, et on peut de plus calculer directement la normale en chaque point en utilisant le gradient de l'intensité dans l'image, qui indique la direction où l'intensité augmente le plus vite, qui est la direction perpendiculaire à la surface (voir figure 1.11).

Ces nuages sont en temps normal précis (l'écart-type sur la position des points est inférieur au voxel et il y a peu de points aberrants), mais les choses se compliquent lorsque le scanner contient des artefacts. Ceux-ci sont générés par les métaux utilisés dans certains plombages et couronnes[8], et se manifestent par un "rayonnement" autour de ces métaux dans chaque coupe (voir figure 1.11 bas-gauche). Ils sont donc limités aux coupes situées au-dessus de la gencive (là où on trouve des plombages et couronnes), et ont pour conséquence de déformer (cf la base de la canine gauche, figure 1.11 bas-droite) voire de faire disparaître la surface des dents (cf la base des prémolaires gauche), et de générer des surfaces aberrantes, en général plus torturées que la vrai surface des dents (cf la partie supérieure de la molaire gauche), ce qui pourrait permettre de les reconnaître et les supprimer par un algorithme de nettoyage ad-hoc. En attendant, il faudra prendre en compte ces défauts dans le recalage.

1.5.2 Image 2D 1/2 : Picza

C'est précisément pour ne pas être tributaires des artefacts que l'on a parfois utilisé une deuxième image pré-opératoire pour la fusionner avec la première : une acquisition 2D 1/2 d'un moulage de la mâchoire. Cette acquisition peut se faire à l'aide d'un capteur mécanique (Picza) ou optique (télémètre laser), qui vient mesurer la hauteur du moulage sur une grille horizontale régulière avec un pas d'échantillonnage de l'ordre de 0.2mm. Les nuages de points ainsi obtenus sont structurés, mais ne sont assorti d'aucune information sur les normales. Le nombre de points est de l'ordre de 50.000 (voir figure 1.12).

Ces nuages sont de bonne qualité (la précision est de l'ordre de $0.1mm$, il n'y a aucun point aberrant), mais présentent deux défauts qu'il faudra prendre en compte dans les

8. Les racines des implants, en titane, ne produisent pas d'artefacts.

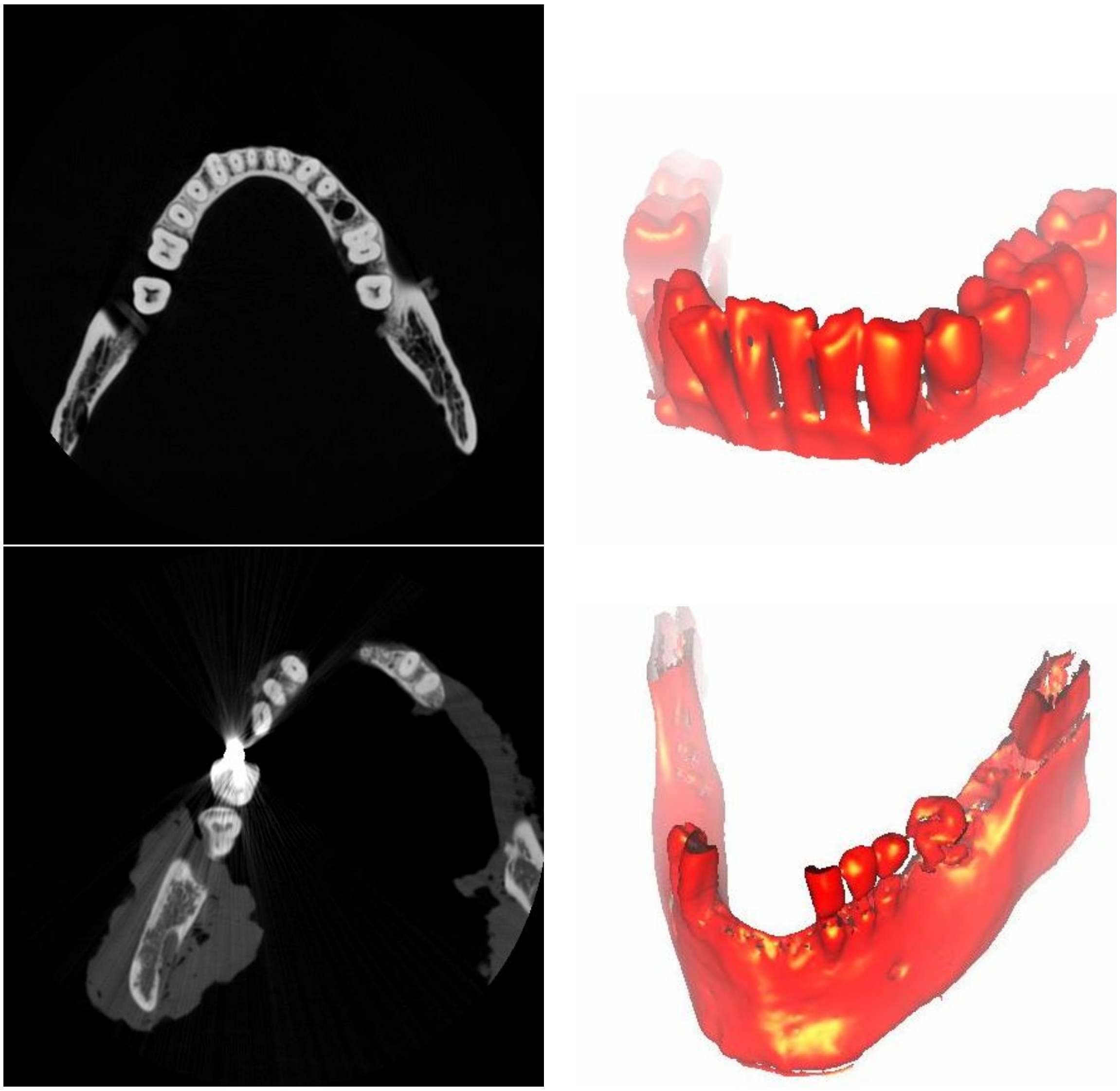


FIG. 1.11 – *Scanner (à gauche) et os segmenté dans ce scanner (à droite) : cas propre (en haut), cas avec artefacts (en bas)*

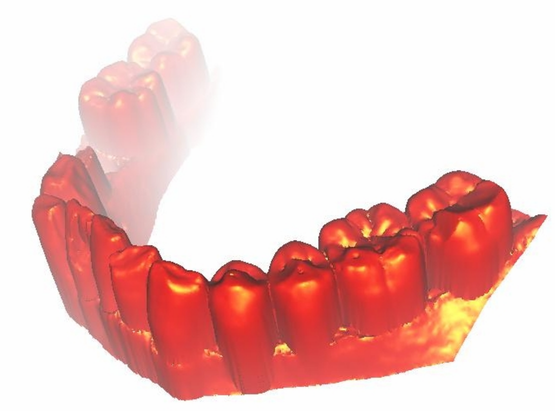


FIG. 1.12 – *Image 2D1/2 de la mâchoire, obtenue avec un capteur Picza*

algorithmes :

- L'absence de points dans les surplombs : comme on mesure les points par le haut suivant une grille régulière dans un plan horizontal, on ne mesurera que peu de points dans les zones presque verticales, et aucun point dans les surplombs.
- La présence de la gencive : comme l'acquisition est faite sur un moulage de la mâchoire, elle correspond à la surface apparente des dents et la surface de la gencive. Elle ne correspond donc que partiellement aux autres acquisitions qui se font sur l'os, i.e. sur la surface apparente des dents et la surface cachée de l'os de la mâchoire.

1.5.3 Pointés per-opératoires

Enfin, pendant l'opération, nous disposons d'une sonde ultra-sonore fixée à un système de suivi (robot passif, capteur magnétique ou capteur optique) et permettant d'aller mesurer la profondeur de l'os. Ce système permet donc d'acquérir des coordonnées de points à la surface de l'os. Il existe deux modes de fonctionnement (voir figure 1.13) :

- En mode contact : l'acquisition se fait point à point, à chaque fois que la sonde est en contact avec l'os ou la gencive. On obtient ainsi des nuages contenant, en fonction de la patience du manipulateur, quelques dizaines à quelques centaines de points.
- En mode rafale : l'acquisition se fait alors suivant un pas de temps fixé. On obtient ainsi des rafales de points.

La sonde doit être à peu près perpendiculaire à la surface de l'os pour fonctionner, et on peut ainsi considérer l'orientation de la sonde comme une approximation grossière de la normale à la surface.

En revanche, ces nuages de points, pourtant pris sur une surface, ne sont pas structurés (dans le cas des rafales, il s'agit tout de même de courbes. Mais la finesse du pas d'échantillonnage par rapport au bruit de mesure leur donne plutôt une apparence de

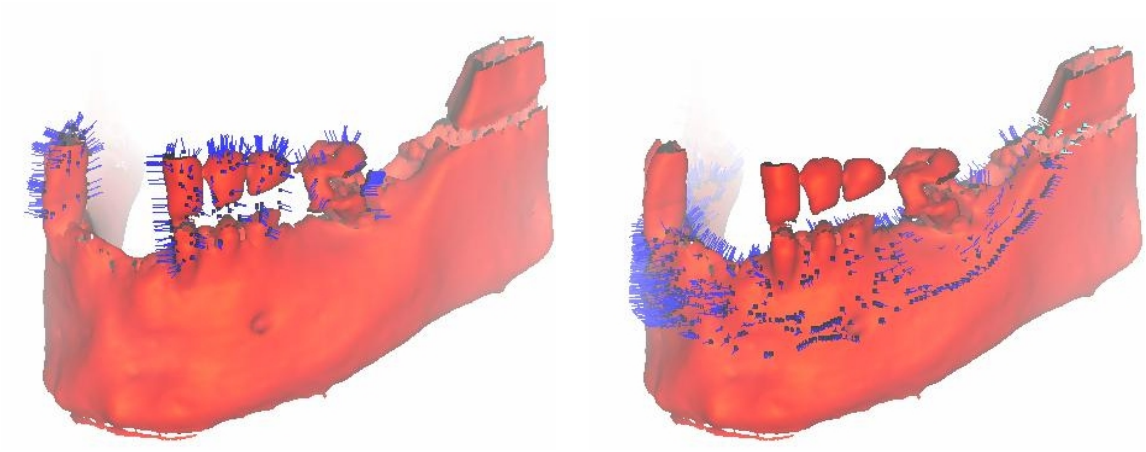


FIG. 1.13 – *Nuages de points per-opératoires recalés sur l'os, en mode contact sur les dents (à gauche) et en mode rafale sur l'os (à droite)*

pâtés). Il est difficile de calculer des informations différentielles (normale, courbures de la surface) avec de tels nuages.

Enfin, ces nuages sont de bonnes qualités (précision de l'ordre de $0.2mm$, parfois quelques points aberrants correspondant à des déclenchements intempestifs de la sonde), mais l'échantillonnage peut-être très irrégulier en mode rafale lorsque la vitesse de déplacement de la sonde est elle-même irrégulière, ce qu'il faudra prendre en compte dans les algorithmes.

1.6 Choix de la méthode de recalage

Il existe plusieurs stratégies pour effectuer un recalage rigide, basées sur différentes manières d'utiliser l'information disponible dans les données. Chaque stratégie est cependant lourde à mettre en place, nécessitant plusieurs mois de développements et de mise au point. Nous avons donc décidé de privilégier une seule stratégie durant cette thèse. Nous expliquons ici pourquoi cette stratégie a été choisie, mais présentons aussi d'autres pistes, basées sur des idées totalement différentes, qu'il faudra peut-être étudier par la suite. Le lecteur pourra se reporter à [Audette *et al.*, 2000] ou [Maintz and Viergever, 1998] pour un état de l'art plus complet.

1.6.1 Recalage basé sur les points

La méthode retenue s'appuie sur un recalage purement géométrique, dont la caractéristique principale est de n'utiliser que l'information sur la position (et éventuellement l'orientation de la normale) des points. On distingue deux grandes classes d'algorithmes de ce type :

- Les algorithmes de recalage d'amers : les points sont peu nombreux et se correspondent, et on peut alors traiter le problème du recalage avec des algorithmes spécifiques [Zhang, 1993].
- Les algorithmes de recalages de formes libres : les points ne se correspondent plus nécessairement comme dans notre cas, où on travaille sur des surfaces échantillonnées.

Le principal algorithme permettant de faire ce dernier type de recalage est l'ICP [Besl and McKay, 1992; Zhang, 1992; 1994; Chen and Medioni, 1992], qui présente de nombreux avantages : il s'agit d'un algorithme simple, peu exigeant en terme de mémoire, rapide, et se prêtant bien à de multiples variantes (voir chapitre 3) permettant de l'adapter à chaque cas particulier. Il faut de plus noter que d'intéressants travaux théoriques ont été réalisées, mettant en évidence de bonnes propriétés de convergence, et abordant le difficile problème de la prédiction de l'incertitude sur le résultat final. Seul point noir au tableau, l'ICP est peu robuste, il exige une bonne initialisation pour converger correctement, et il est parfois difficile de reconnaître une mauvaise d'une bonne convergence.

Enfin, on peut remarquer que si cet algorithme est parfaitement adapté au deux derniers types de données, il ne correspond en revanche pas au premier type de données, le scanner. Pour l'utiliser, on peut envisager deux stratégies différentes. La première passe par une segmentation préalable de la surface des dents dans le scanner. Il s'agit de la technique utilisée actuellement. La seconde technique, plus lourde, consisterait à utiliser l'image telle quelle pendant le recalage, cherchant au coup par coup l'interface des dents à proximité d'un point donné, en examinant, par exemple, le profil d'intensité dans l'axe de la normale. De telles techniques sont justement utilisées pour la segmentation du foie [Delingette, 1999]. Elles nous ont semblé inadaptées dans notre cas, car la segmentation est aisée, et permet de condenser considérablement l'information disponible dans l'image scanner, en ne gardant que l'information véritablement intéressante, i.e. la surface des dents.

1.6.2 Recalage basé sur les invariants géométriques

Il s'agit, pour ce genre de méthodes, d'utiliser l'information invariante pour mettre en correspondance différents points ou parties de l'image. Ces informations peuvent être disponibles directement dans l'image, comme les courbures de la surface sous-jacente en un point [Duncan *et al.*, 1991; Guéziec and Ayache, 1991; 1994; Feldmar and Ayache, 1996], que nous pouvons calculer lors de la segmentation du scanner. Nous ne pouvons en revanche calculer que très grossièrement ces informations pour le Picza, et ne pouvons du tout le faire pour les pointés. Cette méthode n'est donc pas adaptée.

On peut envisager de fabriquer des invariants de manière artificielle, en recherchant par exemple les droites qui sont tangentes ou normales à la surface en deux points (appelées bi-normales ou bi-tangentes) [Feldmar *et al.*, 1997a]. Cette recherche est cependant difficile, et nous n'avons donc pas exploré cette voie, inapplicable de toute façon au recalage avec les pointés per-opératoires.

1.6.3 Recalage basé sur la forme globale

Il existe des techniques de visions par ordinateur permettant de recaler deux surfaces ou volumes en prenant en compte leur forme globale [Johnson and Hebert, 1998; Campbell and Flynn, 1999]. Elles seraient dans notre cas intrinsèquement peu précises, car la forme globale de nos nuages sont différentes à cause des points aberrants et des occlusions.

1.6.4 Recalage basé sur un modèle géométrique

On peut enfin envisager de recaler chacun des deux nuages de points issus de la mâchoire sur un même modèle géométrique de cette mâchoire [McInerney and Terzopoulos, 1996; Sethian, 1996; Montagnat and Delingette, 2001], puis en déduire la transformation séparant les deux nuages de points. Elles exigent de plus de mettre en place un modèle, ce qui parait difficile dans le cas des mâchoires, extrêmement variables d'un sujet à l'autre.

1.6.5 Recalage iconique

Il s'agit de méthodes plus récentes, basées sur l'information d'intensité présente dans les images [Roche, 2001]. On pourrait envisager d'utiliser de telles techniques, à condition de créer une image volumique à partir des nuages de points issus du Picza ou de notre modalité per-opératoire. Ceci reviendrait à augmenter considérablement la taille des données, et mènerait à des algorithmes plus lourds et donc plus lents.

1.7 Problématique de la thèse et organisation du mémoire

Pour résumer, la principale problématique de cette thèse est la mise au point d'un algorithme de recalage rigide de surfaces échantillonnées qui sera utilisé dans des conditions cliniques. On doit donc maîtriser sa robustesse (i.e. sa capacité à trouver une transformation pertinente en fonction de la qualité de la transformation initiale), sa précision et sa vitesse d'exécution. Les problématiques secondaires sont :

- la prédiction théorique de l'incertitude des recalages et son application au guidage de l'acquisition des points per-opératoires.

– le nettoyage ("lissage") des nuages de points (qui pourrait permettre d'éliminer les artefacts dans les scanners), la reconstruction (au sens structuration) de nuages de points provenant de courbes ou de surfaces (qui pourra être utile pour les rafales de points per-opératoires) et la fusion de surfaces (qui permettra de fusionner les surfaces du Picza et du scanner en pré-opératoire).

Dans tous les cas, il faut tenir compte des différents types de données, de leur aspect bruité, et des problèmes spécifiques qu'elles peuvent poser (présence de points aberrants sur certains scanners segmentés, présence de la gencive et absence d'information dans les surplombs sur les Piczas, absence de structuration sur les pointés per-opératoires, échantillonnage très irrégulier sur les pointés en mode rafale).

Cette omni-présence des problèmes de bruit, de précision et de robustesse nous a orienté vers deux grands types de méthodes :

– Les méthodes de statistiques bayésiennes : elles permettent de prendre intrinsèquement en compte la présence de bruit. Elles permettent de plus de manier les notions de précision.
– Les méthodes multi-échelles utilisées sur les images volumiques : elles se sont montrées particulièrement efficaces pour améliorer la robustesse et le temps de calcul des algorithmes.

Or il est souvent possible de mêler ces deux approches : une fois la méthode bayésienne mise au point, il suffit d'appliquer un coefficient d'échelle aux variances des modèles de bruits qu'elle fait intervenir pour obtenir une méthode qui peut s'apparenter aux méthodes multi-échelles (mais n'en n'est pas forcément une). Notre approche générale sera donc la suivante :

– Formaliser le problème en terme probabilistes.
– Le résoudre par une approche bayésienne (Maximum A Posteriori, Maximum de Vraisemblance ou EM).
– Appliquer un coefficient d'échelle aux modèles de bruit, et voir si l'algorithme ainsi adapté présente effectivement les propriétés du multi-échelle.

Pour le recalage, l'algorithme ICP semble être le meilleur point de départ, mais présente de gros problèmes de robustesse, et une imprécision et un temps de calcul un peu trop élevé. Nous lui appliquerons donc notre approche en nous basant dans un premier temps sur des modèles de bruits gaussiens, qui permettent une formulation simple :

– Le chapitre 3 présente l'état de l'art sur l'ICP et ses nombreuses variantes.
– Le chapitre 4 applique notre démarche à l'ICP, ce qui permet d'obtenir l'ICP/EM multi-échelle pour le recalage de points identifiés.
– Le chapitre 5 présente quelques variantes simples de l'ICP/EM spécialement dédiées aux surfaces. Il se consacre aussi à la prédiction théorique de la précision de ces algorithmes.

Une fois les algorithmes de recalage mis au point, il faut pouvoir comparer expérimentalement les différentes variantes, régler les paramètres, et valider les méthodes en vue de leur application clinique. Le chapitre 6 présente donc les techniques (elles-aussi de nature statistiques) d'évaluation expérimentale de la robustesse et la précision des algorithmes de recalage rigide, et leurs applications (optimisation et validation du recalage pour une application clinique donnée).

Bien que les résultats ainsi obtenus soient satisfaisants pour le recalage dans VirtualScope, ils ne répondent pas totalement à notre problématique, pour deux raisons :

- Ils s'appuient sur un modèle de bruit gaussien trop simple pour les surfaces, et ne sont donc pas satisfaisants d'un point de vue théorique.
- Ils n'abordent absolument pas les problèmes du nettoyage, de la reconstruction et de la fusion des surfaces échantillonnées.

Nous avons donc poursuivi ce travail de thèse en attaquant le premier point, d'une part pour l'appliquer au recalage, d'autre part dans l'espoir (fondé sur les remarquables succès de travaux antérieurs) qu'il nous permette de dériver des algorithmes pour le second point. Nous avons donc étudié des modèles de bruits pour les surfaces échantillonnées et bruitées en nous basant sur l'élément élémentaire de ces surfaces, le point semi-orienté aléatoire (qui représente un point de la surface et la normale à la surface en ce point) :

- Le chapitre 7 présente les outils de statistiques exactes et approchées que nous utiliserons sur les normales aléatoires. Il trouve une première application dans l'interpolation des normales sur un nuage de points non-structuré et non-orienté issu d'une surface.
- Le chapitre 8 présente les points semi-orientés aléatoires et des modèles probabilistes d'éléments de surface. Il les applique au recalage, et met en perspective ces modèles en expliquant, dans les grandes lignes, comment ils peuvent ou pourraient être appliqués, sur la base d'un même formalisme (toujours bayésien et multi-échelle), à d'autres problèmes que le recalage, tels que le nettoyage, la reconstruction, la fusion et la complétion de courbes et de surfaces.

1.8 Contributions et publications

Les contributions de ce travail de thèse sont les suivantes :

- Le chapitre 4 regroupe et complète les travaux sur la formalisation statistique de l'ICP et la mise au point de l'ICP/EM [Chui and Rangarajan, 2000; Wells, 1997]. L'interprétation de l'ICP/EM comme un algorithme multi-échelle, la preuve de son comportement pour des échelles extrêmes et son association à une décimation spécialement adaptée sont originales. Ces travaux ont donné lieu à des publications dans les conférences MICCAI'01 [Granger *et al.*, 2001a] et ECCV'02 [Granger and Pennec, 2002b].

- Les variantes de l'ICP/EM présentées dans le chapitre 5 sont originales : l'utilisation de points orientés (qui est décrite dans le rapport de recherche [Granger *et al.*, 2001b], version étoffée de [Granger *et al.*, 2001a]), l'utilisation de points semi-anisotropes, la correction de décalage sur les données, les méthodes de prédiction de l'incertitude adaptées aux surfaces et les méthodes de guidage de l'acquisition (ces deux dernières devraient être publiées dans un article de revue).
- Le chapitre 6, consacré à l'évaluation expérimentale des algorithmes, regroupe des idées existantes pour créer un banc de test automatique, complet et efficace. Il permet l'optimisation automatique des paramètres des algorithmes et les systèmes de sécurité. Un article concernant l'étude expérimentale de la précision a été soumis à la conférence CARS'03 [Granger *et al.*, 2003].
- Le chapitre 7 résume les travaux de [Pennec, 1996], et les applique aux statistiques rigoureuses sur les normales. La seconde partie, consacrée aux statistiques approchées introduit un nouveau canevas théorique permettant d'expliquer certaines heuristiques utilisées dans le domaine de la vision par ordinateur. Elle justifie en particulier la méthode d'interpolation des normales par vote de tenseurs. Elle a donné lieu à un rapport de recherche [Granger and Pennec, 2002a] et un article de revue est en préparation.
- Le chapitre 8 présente une manière d'appréhender la notion d'élément de courbe ou de surface aléatoire, fortement inspirée des travaux sur la saillance [August, 2001] (études des mécanismes bas niveau de la vision humaine, et applications dans le filtrage, la segmentation d'images 2D et la complétion de courbes en 2D) et sur le vote de tenseurs [Médioni *et al.*, 2000] (regroupement et reconstruction de courbes et de surfaces bruitées en 2D et 3D). Nous présentons sous une forme commune 3 modèles basés sur ces travaux. Nous formalisons et complétons la notion de champs de vote d'un point bruité, et donnons pour la première fois des formules explicites bien qu'approximatives. Nous les confrontons aux votes empiriques observés sur nos surfaces. Nous présentons de nombreuses idées, encore incomplètes et inappliquées mais qui pourraient permettre de faire déboucher ces travaux sur une théorie unificatrice dans le domaine de la complétion, la segmentation, la reconstruction et la fusion de surfaces bruitées. Nous présentons enfin une nouvelle application, au recalage de surface bien-sûr !

Chapitre 2

Notations et glossaire

Sommaire

Nous présentons ici les notations et termes de vocabulaire utilisés dans cette thèse. La lecture de ce mémoire pourra en outre être facilitée par la lecture des annexes qui présentent certains outils mathématiques utilisés dans cette thèse :

- Le calcul tensoriel (annexe E).
- Le calcul des probabilités et les statistiques (annexe C).
- Les primitives (normales, transformations rigides) aléatoires (annexe D).

2.1 Vocabulaire

L'utilisation que nous faisons de certains termes est parfois inhabituelle. Nous préférons les préciser ici.

Concernant les variances, nous utilisons les termes suivants :

Covariance : La covariance $\Sigma_{\mathbf{xx}} = E\left[\overrightarrow{\overline{x}x}.\overrightarrow{\overline{x}x}^t\right]$

Variance totale : La trace de la covariance, i.e. l'espérance de la distance carré : $Tr(\Sigma_{\mathbf{xx}}) = E\left[d(\overline{x},x)^2\right]$

Écart-type total : La racine carré de la variance totale $\sqrt{Tr(\Sigma_{\mathbf{xx}})}$

Variance : La variance (sous entendu *par direction*), qui correspond au paramètre $\sigma_{\mathbf{x}}^2$ pour une covariance isotrope : $\Sigma_{\mathbf{xx}} = \sigma_{\mathbf{x}}^2.Id$. La variance totale est alors donnée par $Tr(\Sigma_{\mathbf{xx}}) = \mathfrak{D}.\sigma_{\mathbf{x}}^2$

Écart-type : La racine carré de la variance $\sigma_{\mathbf{x}} = \sqrt{\sigma_{\mathbf{x}}^2}$, l'écart-type total est alors donné par $\sqrt{Tr(\Sigma_{\mathbf{xx}})} = \mathfrak{D}.\sigma_{\mathbf{x}}$

Matrice de dispersion : La matrice $\Delta_{\mathbf{nn}}$ qui apparaît dans une loi de probabilité gaussienne $\frac{e^{-\frac{\overrightarrow{\overline{n}m}^t.\Delta_{\mathbf{nn}}^{-1}.\overrightarrow{\overline{n}m}}{2}}}{k}$, et qui dans le cas des primitives (normales, transformations rigides, ...) est différente de la covariance (voir section 7.2.8).

Dispersion totale : La trace de la matrice de dispersion.

Dispersion : La dispersion (sous entendu *par direction*), qui correspond au paramètre $\delta_{\mathbf{n}}^2$ pour une matrice de dispersion isotrope : $\Delta_{\mathbf{nn}} = \delta_{\mathbf{n}}^2.Id$. La dispersion totale est alors donnée par $Tr(\Delta_{\mathbf{nn}}) = \mathcal{D}.\delta_{\mathbf{n}}^2$

Dispersion-type : La racine carré de la dispersion $\delta_{\mathbf{n}} = \sqrt{\delta_{\mathbf{n}}^2}$

Concernant les primitives géométriques, nous utilisons les termes suivants :

Normale orientée (normale) : une normale, i.e. un vecteur unitaire

Normale non-orientée (direction) : une normale au sens près ($n = -n$)

Point non-orienté : un simple point

Point semi-orienté : un point + une normale non-orientée

Point orienté : un point + une normale orientée

2.2 Notations

On utilisera les notations suivante :

Géométrie

Espaces

$\mathfrak{D}$ dimension de l'espace euclidien de travail

$E = \mathbb{R}^{\mathfrak{D}}$ Espace euclidien de travail

x point de E

$\mathcal{D}$ dimension de la variété différentielle (1 pour les courbes, 2 pour les surfaces, l'espace des normales, 3 pour l'espace des rotations).

$\mathcal{M}$ Variété différentielle

$\mathcal{S}_{\mathfrak{D}-1}$ ensemble des normales orientées

$\mathcal{P}_{\mathfrak{D}-1}$ ensemble des normales non-orientées

$\mathcal{SO}_{\mathfrak{D}}$ ensemble des rotations

$\mathbb{R}^{\mathfrak{D}} \times \mathcal{S}_{\mathfrak{D}-1}$ ensemble des points orientés

$\mathbb{R}^{\mathfrak{D}} \times \mathcal{P}_{\mathfrak{D}-1}$ ensemble des points semi-orientées

$\mathbb{R}^{\mathfrak{D}} \times \mathcal{SO}_{\mathfrak{D}}$ ensemble des transformations rigides

$\mathtt{x}$ primitive

$\mathtt{o}$ primitive origine

$\overrightarrow{xy}$ représentation de $\mathtt{y}$ dans la carte exponentielle centrée en $\mathtt{x}$

Distances

$(.|.)$ Produit scalaire

$\|.\|$Norme euclidienne

$d(.,.)$ distance (en général euclidienne) entre deux points.

$D(.,.)$ distance (suivant contexte, en général carré de la distance euclidienne ou distance de Mahalanobis) entre deux primitives.

Normales

n Normales

Rotations et transformations rigides

R Rotation (ou sa matrice) autour de l'origine

$R_{a,\theta}$ Rotation autour de l'origine d'axe a et d'angle θ

r Vecteur de la rotation autour de l'origine

$T = (R,t)$ Transformation rigide $(T \star x = Rx + t)$

$t = (r,t)$ Vecteur d'une transformation rigide

Tenseurs

$a \otimes b = a.b^t$ Produit tensoriel

$a \overline{\otimes} b = a^t.b = Tr(a \otimes b)$ Produit tensoriel contracté, équivalent au produit scalaire $(a|b)$

Recalage

M (resp. S) Modèle (resp. scène)

n_M (resp. n_S) Nombre de points dans le modèle (resp. scène)

m_j (resp. s_i) Points du modèle (resp. scène)

n_{m_j} (resp. n_{s_i}) Normales aux points du modèle (resp. scène)

$T = (R,t)$ Transformation rigide appliquée à la scène ($T \star s_i = R.s_i + t$) pour la ramener sur le modèle

T_0 La bonne transformation

T^0 L'estimée initiale de la transformation

e_{m_j} (resp. e_{s_i}) Erreur de mesure sur le point

$e_{ij}(T)$ Vecteur d'erreur entre deux points appariés : $e_{ij}(T) = T \star s_i - m_j$

$A_{ij}(T)$ Matrice d'appariement

$\mathbf{A}$ Matrice d'appariements aléatoire

$\overline{A}$ L'espérance de $\mathbf{A}$ ($\overline{A_{ij}}$ Probabilité de l'appariement entre s_i et m_j dans le cas non-corrélé)

$\overline{\pi}$ L'espérance a-priori des appariements ($\overline{\pi_{ij}}$ Probabilité a-priori de l'appariement entre s_i et m_j)

Abréviations pour les calculs de dérivées

s_{Ti} Raccourci pour $T \star s_i$

A_T Raccourci pour $A(T)$

$\Phi_v C$ Gradient de C par rapport à v, raccourci de $\frac{dC}{dv}^t$

$H_v C$ Hessien de C par rapport à v, raccourci de $\frac{d\Phi_v C}{dv}$

Paramètres de l'algorithme

σ^2 Variance sur les points (variance dans l'axe de la normale pour les bruits semi-anisotropes)

σ_n^2 Variance sur la normale

σ_t^2 Variance dans la plan tangent pour les bruits semi-anisotropes

μ_{max}^2 Distance de Mahalanobis maximale (ou d_{max} distance maximale)

α Coefficient pour le rayon des sphères de décimation : $r = \alpha.\sigma$

Probabilités

$Pr(A)$ Probabilité de l'évènement aléatoire A

$Pr(A|B)$ Probabilité de l'évènement aléatoire A sachant B

$p(x)$ Densité de probabilité de l'évènement aléatoire x

$p(x|B)$ Densité de probabilité de l'évènement aléatoire x sachant B

$\overline{x} = E\left[x\right]$ Espérance du l'évènement aléatoire x

$\mu^2(x,y) = \overrightarrow{xy}^t.\Sigma_{xx}^{-1}.\overrightarrow{xy}$ Distance de Mahalanobis

$\chi^2_{\mathfrak{D}}$ Loi du χ^2 à $\mathfrak{D}$ dimensions

Première partie

Recalage de surfaces

Chapitre 3

ICP et variantes : état de l'art

Sommaire

3.1 Introduction

L'algorithme ICP, introduit au début des années 90 a connu depuis un immense succès dans le domaine du recalage de surfaces [Audette *et al.*, 2000], où il est utilisé dans 95% des cas. Il a surtout connu un grand nombre de variantes. Nous n'en présentons ici qu'une partie, celles qui seront utiles pour comprendre la suite de cette thèse. Le lecteur pourra se reporter à [Rusinkiewicz and Levoy, 2001] pour un état de l'art plus complet, et doublé d'une bonne comparaison expérimentale.

3.1.1 Problématique

Nous souhaitons recaler de manière rigide différents nuages de points. Ces points peuvent être acquis de plusieurs manières différentes (voir section 1.5) :

- A partir d'une image volumique : il peut s'agir du résultat d'une segmentation ou d'une recherche d'amers géométriques (comme les lignes de crêtes). Ce sont donc des nuages de points assortis de certaines informations différentielles (normale, courbures), et structurés. Dans notre cas, nous segmentons le scanner X et obtenons une triangulation comptant une cinquantaine à une centaine de milliers de points munis de leurs normales, et répartis sur les arrêtes des voxels de l'image, donc espacés assez régulièrement.
- A partir d'une image 2D 1/2 ("range data" en anglais) : il s'agit en général de l'acquisition par un appareil mécanique ou optique de la surface d'un objet (une copie plâtre d'une mâchoire dans notre cas). L'acquisition se fait en général suivant une grille régulière, et le nuage de points est donc facilement structurable. En revanche, l'information différentielle (normale, courbures) n'est pas directement disponible, et ne peut être que grossièrement interpolée. Nous utilisons pour l'instant un appareil nommé Picza, appareil mécanique qui réalise une carte des hauteurs de la mâchoire sur une grille régulière avec un pas variant entre 0.2 et 1mm. Le nuage obtenu compte une cinquantaine de milliers de points pour un pas de 0.2mm.
- A partir de points épars : il peut s'agir de points acquis à l'aide d'une sonde, un par un ou en rafales. Dans le premier cas, ils ne sont pas structurés et seule l'inclinaison de la sonde (qui doit être à peu près perpendiculaire à la surface pour fonctionner) peut nous donner une information grossière sur la normale à la surface. Dans le second cas, ils forment des courbes, dont on peut calculer la tangente avec une bonne précision.

On peut faire deux hypothèses différentes quant à la répartition des points.

Points homologues Nous supposerons ici que les points du modèle et de la scène sont, aux problèmes d'occultation et de points aberrants près, les mesures des même points physiques. Ce sera par exemple le cas lorsqu'on utilisera des amers. Un point de la scène s_i est alors soit un point aberrant (ou correspondre à un point occulté dans le modèle, ce que nous considérerons dans la suite comme équivalent) soit la transformée d'un point m_j du modèle, relié par la relation $T_0 \star s_i = m_j$, au bruit de mesure près. Nous dirons dans ce cas là que les points s_i et m_j sont homologues.

Points issus d'une même surface ou courbe Nous supposerons ici que les points sont issus d'une même variété différentielle (courbe ou surface), mais sont échantillonnés d'une manière différente. Ainsi, les points ne sont plus la mesure du même point physique et ne sont donc plus homologues.

3.1.2 Généralités sur l'ICP

Avant de rentrer dans les détails, nous allons donner les idées fondatrices des différents algorithmes de type ICP.

Ils s'appuient sur la minimisation d'un critère. Ce critère est sensé quantifier la distance entre la scène et le modèle en fonction de la transformation rigide appliquée à la scène. Pour écrire ce critère, on utilise une notion d'appariements, i.e. une façon d'identifier les points de la scène et du modèle, et on utilise, en guise de critère*, la somme des distances quadratiques entre les points (primitives) appariés.*

Ainsi l'optimisation du critère, et donc le calcul de la transformation rigide, s'appuie sur une estimation des appariements. Or le calcul des appariements nécessite en général de connaître une estimation de la transformation rigide cherchée. Cette remarque caractérise les algorithmes de type ICP qui se basent sur l'estimation alternée des appariements et de la transformation. Plusieurs itérations seront nécessaires pour affiner les deux estimations, et la première sera basée sur une première estimation de la transformation (dite transformation initiale).

Cette présentation succincte montre aussi que l'ICP laisse une grande liberté dans le choix de :

- la méthode d'initialisation : il s'agit de calculer la transformation initiale, en utilisant un algorithme spécifique. La qualité de cette transformation conditionne la robustesse du recalage.
- le type de transformation : nous nous restreindrons ici au cas rigide, mais L'ICP est aussi utilisé dans le cas non rigide [Feldmar, 1995; Feldmar and Ayache, 1996].
- la distance : il s'agira en général d'une distance quadratique, car les algorithmes d'optimisation sont particulièrement efficaces avec ce type de distance. On pourra l'adapter en fonction des informations dont l'on dispose : distance quadratique entre points dans le cas le plus simple, distance de Mahalanobis lorsqu'on dispose des covariances sur les points (ou de simples écart-types dans le cas isotrope), distance

quadratique entre primitives lorsqu'on dispose de primitives plus complexes (points orientés, ...). On verra aussi que l'on peut, en adaptant la distance, généraliser l'ICP au cas où l'on dispose d'invariants, et même au cas iconique.
- le lien entre la scène et le modèle : considère-t-on les points comme homologues, ou comme issus d'une même surface, et dans ce cas là, comment le modéliser ?
- la manière d'appréhender et estimer les appariements : il s'agira de fournir des paires de points appariés assorties d'un poids, en fonction de la transformation courante. On pourra à cette étape utiliser directement les points du modèle ou en interpoler de nouveaux, et moduler l'importance de chaque appariement avec différents types de pondérations.
- la manière de trouver et traiter les points aberrants : il s'agira de traiter les faux positifs, les appariements qui n'auraient pas dû être, et les faux négatifs, les appariements manquants. En fait, seul le rejet des faux positifs sera intéressant à implémenter, et nous détaillerons sa mise en oeuvre pratique et les conséquences théoriques.
- la technique de minimisation du critère adoptée pour la mise à jour de la transformation à chaque étape de l'algorithme (comme par exemple les Moindre carrés, les filtre de Kalman).

Ces différents points formeront autant de modules plus ou moins indépendants au sein de l'algorithme.

3.1.3 Organisation du chapitre

Le but de ce chapitre est de regrouper différents travaux se rapportant à l'ICP en utilisant des notations et un formalisme commun. Il permettra de préciser et clarifier si nécessaire les hypothèses, raisonnements et applications possibles, indiquant comment les modifier pour respecter les propriétés théoriques de l'algorithme de base. Il met en évidence la structure très modulaire de l'algorithme, et montre, pour chaque module, les variantes les plus intéressantes, en indiquant, après une éventuelle adaptation, leurs conséquences sur les autres parties de l'algorithme. Cette démarche permet en particulier d'indiquer quand et comment généraliser certains travaux, comme le calcul de l'incertitude, ou de mettre en évidence les liens jusqu'ici ignorés entre différents travaux, en particulier pour le calcul des pondérations.

Le lecteur pressé pourra se contenter de lire la section 3.2 qui présente la forme la plus simple de l'algorithme, et permet d'introduire les notations et concepts de bases utilisés par la suite. Il faudra cependant lire aussi le début de la section 3.6 pour comprendre la forme classique de l'algorithme, en général appelé ICP robuste (il s'agit alors de la robustesse vis-à-vis des points aberrants, et non celle qui nous intéressera souvent, i.e. la robustesse vis-à-vis de la transformation initiale).

Les sections 3.3 à 3.7 présenteront, par module, certaines variantes déjà imaginées autour l'ICP. Les deux premières sections (initialisation et interpolation) sont un peu annexes, et peuvent être ignorées. Les sections cherchent ensuite à généraliser l'algorithme à tout type de primitive. La partie 3.5, consacrée à la notion de distance, servira de charnière. On y introduira la distance de Mahalanobis, systématiquement utilisée ensuite car elle permet de prendre en compte les bruits pour des types de primitives variées. La partie 3.6 présentera en détail la gestion des points aberrants, et la partie 3.7 introduira différents types de pondérations.

Les parties 3.8 et 3.9 seront consacrées à la minimisation du critère, donc au calcul effectif de la transformation et au calcul théorique de sa covariance, qui mesure l'incertitude de l'algorithme.

3.2 ICP classique

Le premier algorithme de type ICP a été présentés par [Besl and McKay, 1992] et [Zhang, 1992; 1994], sous deux formes légèrement différentes, mais équivalentes. Notons que [Chen and Medioni, 1991; 1992] présentait déjà un algorithme de minimisation alternée proche de l'ICP.

Cet algorithme s'applique à deux nuages de points, et ne repose pas sur l'hypothèse "points homologues". C'est cependant en prenant cette hypothèse en compte que l'on peut en donner une interprétation satisfaisante.

Nous commencerons par une écriture du critère fidèle à celle que l'on trouve dans les deux premiers papiers. Nous adopterons ensuite une notation plus générale et adaptée à la suite. Nous expliquerons alors le principe de l'algorithme, puis ses principales propriétés. Nous détaillerons enfin la mise en oeuvre pratique des différentes étapes.

Notations Le modèle (resp. scène) est noté M (resp. S) et constitué de n_M (resp. n_S) points ou primitives. Nous noterons les points du modèle m_j et les points de la scène s_i, qui peuvent être assortis de normales notées $n_{m_j}(n_{s_i})$. Les transformations rigides seront notées $T = (R,t)$ et appliquées à la scène (pour la ramener dans la même position que le modèle) de la façon suivante : $T \star s_i = R.s_i + t$ (et donc $T \star t_{s_i} = R.t_{s_i}$ pour les normales). La véritable transformation qui sépare les deux ensembles est notée T_0. La distance utilisée dans le critère sera notée D. Il s'agira, au début, de la distance usuelle au carré $D(x,y) = \|x - y\|^2$.

3.2.1 Critère

Approche pragmatique

Comme nous l'avons dit, le critère essaye de quantifier la distance entre la scène et le modèle. La première mesure de distance proposée s'appuie sur une définition pragmatique de la distance entre deux ensembles de points : il s'agit de la moyenne des distances entre un point d'un ensemble et la totalité de l'autre ensemble :

$$C(T) = \frac{1}{n_S} \sum_i D(T \star s_i, M) + \frac{1}{n_M} \sum_i D(m_j, T \star S)$$

Le critère est ici symétrique (il fait jouer le même rôle à la scène et au modèle). Cependant, pour faciliter l'implémentation, nous casserons cette symétrie en ne gardant que le premier terme :

$$C(T) = \frac{1}{n_S} \sum_i D(T \star s_i, M)$$

Ainsi, on n'obtiendra pas le même résultat en inversant la scène et le modèle. Il faudra donc veiller à utiliser le cas qui donne le résultat le plus pertinent. Nous verrons dans la suite qu'il s'agit en général du cas où on utilise, en guise de scène, l'ensemble qui contient le moins de points (qui est donc sous-échantillonné par rapport à l'autre). Nous verrons aussi (à la section 3.7.2) qu'il est possible d'adapter certaines variantes pour obtenir à nouveau un critère symétrique.

Reste à définir la distance entre un point et un ensemble de points. On utilise pour cela dans l'ICP classique le carré de la distance au plus proche voisin (en anglais "Closest Point", d'où le nom de l'algorithme) :

$$D(T \star s_i, M) = \min_j D(T \star s_i, m_j) = \min_j \|T \star s_i - m_j\|^2$$

L'algorithme est donc basé sur la notion implicite de points appariés : un point de la scène est apparié à un point du modèle lorsque c'est son plus proche voisin, et c'est bien une distance quadratique moyenne entre points appariés que l'on doit minimiser.

Le critère peut finalement s'écrire :

$$C(T) = \frac{1}{n_S} \sum_i \min_j D(T \star s_i, m_j) \tag{3.1}$$

Approche supposant les points véritablement homologues

Cette première écriture semble parfaitement adaptée au problème : elle ne dépend que de T, car l'estimation des appariements y est implicite. Elle empêche de considérer des appariements multiples, et tous les appariements sont traités de façon équivalente (on ne peux donner un poids plus important à un appariement donné, lorsqu'on le considère comme plus significatif).

Nous aurons cependant besoin par la suite d'un cadre plus large, autorisant les appariements multiples et pondérés de diverses manières. Pour cela, nous allons utiliser une seconde écriture du critère, qui fait *apparaître la configuration d'appariement comme une variable explicite, indépendante de la transformation, qu'il faudra aussi estimer.*

C'est ici que nous prendrons en compte l'hypothèse des points homologues. La notion d'appariement est alors simple à définir : un point s_i de la scène et un point m_j du modèle sont dit appariés (notation : $s_i \sim m_j$) s'ils sont issus de deux mesures du *même point physique*. Nous représenterons ces appariements par une matrice à éléments binaires :

$$A_{ij} = A(s_i,m_j) = \begin{cases} 1 & \text{si} \quad s_i \sim m_j \\ 0 & \text{sinon} \end{cases} \tag{3.2}$$

On remarque que $\sum_j A_{ij}$ (resp. $\sum_i A_{ij}$) est le nombre de points appariés avec s_i (resp. m_j). Le nombre d'appariements total n'est autre que $\sum_{ij} A_{ij}$. En l'absence de points aberrants, un point de la scène (resp. modèle) doit être apparié à un et un seul point du modèle (resp. scène), et on a $\sum_j A_{ij} = \sum_i A_{ij} = 1$, donc $\sum_{ij} A_{ij} = n_S = n_M$. Ces hypothèses sont évidement trop fortes en présence de points aberrants, et nous verrons plus loin comment les assouplir (voir 3.6). Nous imposerons en attendant que chaque point de la scène soit apparié à un unique point du modèle, hypothèse adaptée au cas où la scène est sous-échantillonnés par rapport au modèle :

$$\sum_j A_{ij} = 1 \tag{3.3}$$

$$\sum_{ij} A_{ij} = n_S$$

Il est important de noter que, bien que l'on ait supposé l'existence des points appariés (mesure d'un même point physique), nous n'avons aucune information a priori sur ces appariements. Aussi, nous allons devoir estimer à la fois la transformation rigide T et les appariements au travers de A. Pour cela, on utilise le critère suivant :

$$C(T,A) = \frac{1}{\sum_{ij} A_{ij}} \sum_{ij} A_{ij}.D(T \star s_i,m_j) \tag{3.4}$$

Remarquons au passage que le terme $\sum_{ij} A_{ij}$ est positif constant. Le comportement du critère étant le même si on le multiplie par une constante positive ou si on lui ajoute une constante quelconque, on raisonnera souvent sur le critère équivalent $C(T,A) = \sum_{ij} A_{ij}.D(T \star s_i,m_j)$, sans faire véritablement de distinction entre les deux.

Pour faire le lien avec le critère $C(T)$ (équation 3.1), il faut examiner ce qu'il se passe lorsqu'on recherche les appariements A_T qui minimisent $C(T,A)$ sous la contrainte $\sum_j A_{ij} = 1$ à transformation T constante. Pour i fixé, cette contrainte signifie qu'il n'existe qu'un seul point du modèle m_J apparié au point de la scène s_i. On a alors $\sum_j A_{ij}.D(T \star s_i,m_j) = D(T \star s_i,m_J)$. Il s'agit donc de trouver la valeur de J qui minimise $D(T \star s_i,m_J)$, c'est à dire de trouver le point m_J du modèle le plus proche de $T \star s_i$. On retrouve donc l'ancien critère :

$$C(T) = \min_A C(T,A) = C(T,A_T) \qquad \text{avec} \quad A_T = \arg\min_A C(T,A) \tag{3.5}$$

Propriétés du critère

Nous avons vu que le critère $C(T)$ était homogène à $\sum_i \min_j d^2(T \star s_i,m_j)$. On peux donc faire les remarques suivantes :

- Il est continu.
- Il est infiniment dérivable presque partout. Les points posant problème sont ceux où les appariements (via la variable A_T) changent.
- Il est positif.

A titre d'exemple, les graphiques de la figure 3.1 ont été obtenus en effectuant un recalage entre une acquisition 2D 1/2 d'une mâchoire (environ 60.000 points, échantillonnés avec un pas de 0.2 mm en X et Y) et une série de pointés sur les dents (30 points, avec une précision d'environ 0.2 mm), qui nous a donné une transformation rigide. On a ensuite tracé la racine carré du critère en faisant varier la translation suivant l'axe Z autour de la transformation (haut gauche). Comme on le voit, le critère possède à première vue une large cuvette avec seul minimum (bien que d'autres minima locaux peuvent apparaître a l'extérieur de cette cuvette, donc loin du bon résultat), vers lequel l'algorithme a convergé. Mais un agrandissement de la zone centrale (haut droite) montre cependant que plusieurs minima locaux apparaissent, donnant au critère une structure caractérisée par plusieurs cuvettes. Nous n'avons d'ailleurs pas ici trouvé le meilleur. Pour comprendre ce phénomène, il faut remarquer que pour une configuration d'appariement fixée, le critère est une simple parabole (bas).

Le critère final est donc quadratique par morceaux, i.e. formé de paraboles correspondant chacune à une estimation des appariements donnée, et se raccordant aux endroits où cette estimation change brusquement.

Estimation de la variance

Cette section s'intéresse à l'estimation de la variance du bruit sur les points. Cette estimation est importante pour diverses raisons. La première est la curiosité. La seconde est

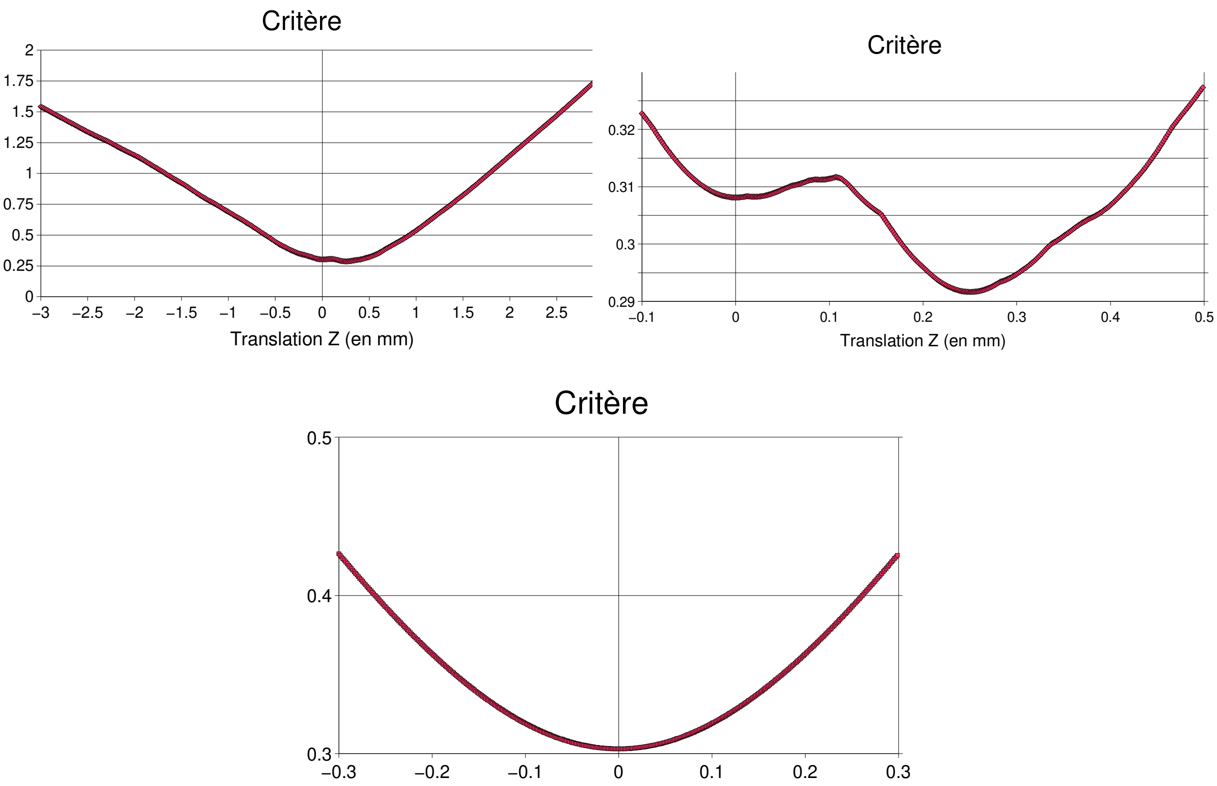


FIG. 3.1 – *Critère en fonction d'une translation en z : pour l'ICP (en haut), et pour une configuration d'appariement fixée (en bas)*

qu'elle est impliquée dans la prédiction théorique de l'incertitude de l'algorithme, présentée dans la partie 3.9. La dernière est qu'elle intervient dans la distance de Mahalanobis que nous utiliserons par la suite.

Pour comprendre comment elle est possible, il faut décomposer le vecteur d'erreur entre deux points appariés $e_{ij}(T) = T \star s_i - m_j$ en deux termes :

$$e_{ij}(T) = T \star s_i - m_j = (T \star s_i - T_0 \star s_i) + (T_0 \star s_i - m_j)$$

Le premier terme $T \star s_i - T_0 \star s_i$ est dû à l'erreur sur l'estimation de la transformation. Il est par construction nul lorsque la transformation est parfaitement estimée. Le deuxième terme $T_0 \star s_i - m_j$ est dû au bruit sur les données. En-effet, on devrait avoir $T_0 \star s_i = m_j$ et donc un terme nul, car nous avons supposé les points homologues. Cependant, nos points ne sont pas parfaitement mesurés, et il faut tenir compte de l'erreur de mesure e_{s_i} (resp. e_{m_j}) sur le point s_i (resp. m_j). On a donc finalement : $T_0 \star s_i - m_j = T_0 \star e_{s_i} - e_{m_j}$.

Nous supposerons dans la suite que les appariements ont été correctement estimés. On peut alors ré-écrire le critère sous la forme :

$$C(T,A) = \frac{1}{\sum_{ij} A_{ij}} \sum_{ij} A_{ij}.D(T\star s_i,m_j) = \frac{1}{\sum_{ij} A_{ij}} \sum_{ij} A_{ij}.\left\|(T \star s_i - T_0 \star s_i) + \left(T_0 \star e_{s_i} - e_{m_j}\right)\right\|^2$$

Lorsque les erreurs de mesures sont centrées, le terme $\sum_{ij} A_{ij}.\left(T_0 \star e_{s_i} - e_{m_j}\right)$ est d'espérance nulle, et on peut donc le négliger. On développe et on simplifie alors le critère :

$$C(T,A) = \frac{1}{\sum_{ij} A_{ij}} \sum_{ij} A_{ij}.\left\|T \star s_i - T_0 \star s_i\right\|^2 + \frac{1}{\sum_{ij} A_{ij}} \sum_{ij} A_{ij}.\left\|T_0 \star e_{s_i} - e_{m_j}\right\|^2$$

On peut vérifier qu'il atteint son minimum pour la bonne transformation. La valeur de ce minimum s'écrit :

$$\min C(T,A) \simeq \frac{1}{\sum_{ij} A_{ij}} \sum_{ij} A_{ij}.\left\|T_0 \star e_{s_i} - e_{m_j}\right\|^2$$

Dans le cas d'un bruit blanc homogène centré, les erreurs $T_0 \star e_{s_i} - e_{m_j}$ sont la réalisation d'un même vecteur aléatoire $\mathbf{e}$ d'espérance nulle. La formule ci-dessus est une estimation statistique de la variance $\mathfrak{D}.\sigma_e^2$ de ce bruit. On calculera facilement cette variance dans le cas d'un bruit isotrope : si les e_{s_i} (resp. e_{m_j}) sont les réalisations un bruit blanc homogène additif isotrope centré de variance $\sigma_{e_s}^2 = \sigma_s^2.Id$ (resp. $\sigma_{e_m}^2 = \sigma_m^2.Id$), alors $\mathbf{e}$ est un bruit blanc additif gaussien, centré de variance $\sigma_e^2 = \sigma_{e_s}^2 + \sigma_{e_m}^2$, et la valeur du critère est une estimation statistique de la variance totale :

$$\min C(T) \simeq \mathfrak{D}.\sigma_{\mathbf{e}}^2 \tag{3.6}$$

On peut vérifier sur notre exemple : la racine carré du critère est de l'ordre de 0.3 mm, ce qui nous donne un écart-type de l'ordre 0.17 mm, un peu sous estimé peut-être. Nous verrons de toute façon que cette estimation est biaisée (section 6.4.2).

3.2.2 Algorithme

Principe

Comme expliqué dans la section 3.1.2, le recalage sera trouvé en minimisant le critère $C(T)$. Le présence de la fonction min, qui exprime les relations d'appariements, le rend cependant très complexe à minimiser directement. Aussi, nous utiliserons un processus itératif, en améliorant alternativement les estimations de A (en minimisant $C(T,A)$ à T constant, ce qui permet au passage de calculer $C(T)$) et de T (en minimisant $C(T,A)$ à A constant) et à partir d'une estimation initiale de la transformation T^0 .

En cherchant à minimiser le critère à chaque étape, on peut être sûr d'une chose : il décroît toujours. Comme il est positif, la convergence de l'algorithme est donc assurée pour un nombre d'itérations infini. En pratique, on estime que l'algorithme a convergé lorsque la variation sur l'estimation de la transformation est inférieure à un seuil donné ϵ.

Le schéma général de l'algorithme est donc le suivant :

Initialisation : calcul de T^0 (voir section 3.3)

Boucle : à l'étape I :

- Mise à jour des appariements (voir section 3.2.3) : Calcul de A_{T^I} par minimisation de $C(T,A)$ à $T = T^I$ constant et calcul de $C(T^I)$.
- Mise à jour de la transformation (voir section 3.2.4 et 3.8) : Calcul de T^{I+1} par minimisation de $C(T,A)$ à $A = A_{T^I}$ constant.

Jusqu'à convergence : $d(T^{I+1},T^I) < \epsilon$

Paramètres auxiliaires Comme nous le voyons, l'algorithme le plus simple ne fait intervenir aucun paramètre. Nous serons cependant amené à introduire des paramètres auxiliaires, comme la distance maximale (voir 3.6.2). Ces paramètres pourront varier entre deux itérations, et il faudra alors veiller à ce qu'ils ne fassent pas remonter le critère (auquel cas la convergence ne serait plus assurée).

Propriétés de l'algorithme

Nous avons vu que l'algorithme permettait de converger vers un minimum du critère $C(T,A)$ qui correspond à un minimum du critère $C(T)$. Ce critère n'est cependant pas convexe (voir section 3.2.1), et le résultat du recalage dépendra de la cuvette où nous mènera la transformation initiale. Une transformation initiale de trop mauvaise qualité nous mènera vers un recalage manifestement faux. Les transformations initiales de bonne qualité ne mèneront pas toutes exactement au même recalage, et cette dépendance introduit donc une première source d'imprécision pour un modèle et une scène donnée. Une seconde source d'imprécision est le bruit sur les points de la scène et du modèle.

Nous avons donc introduit la distinction suivante (voir chapitre 6 pour plus de détails), permettant de mieux comprendre et appréhender l'incertitude et la robustesse de l'algorithme :

Robustesse (par rapport à la transformation initiale) On parlera de robustesse pour désigner la capacité de l'algorithme à ne pas se diriger vers des résultats complètement faux. Il faut pour cela démarrer dans la cuvette centrée autour de la bonne transformation. La robustesse est donc essentiellement conditionnée par la qualité de la transformation initiale, mais aussi, de façon marginale, par la qualité de la scène et du modèle. Notons au passage que les résultats complètement faux sont en général très éloignés de la bonne transfo (puisqu'ils sont à l'extérieur de la cuvette principale).

Bruit Interne ou répétabilité Si nous supposons maintenant que notre algorithme est parfaitement robuste (si nous avons une bonne méthode d'initialisation, ou si nous sommes capable de rejeter les résultats complètement faux), tous nos recalages entre un modèle et une scène donnée convergeront vers des transformations proches de la transformation cherchée. Ces différents résultats ne seront cependant pas identiques, et on désignera par bruit interne (ou imprécision interne) de l'algorithme leur variabilité.

Il ne faut cependant pas interpréter ce bruit comme la conséquence de l'initialisation, mais comme la conséquence de l'erreur sur l'estimation de la configuration des appariements, et il est donc intrinsèque à l'algorithme (d'où son nom). Ainsi, il apparaît aussi lorsqu'on recale deux scènes légèrement différentes à partir de la même transfo initiale.

Bruit externe et précision On désignera par bruit externe la conséquence du bruit sur les données sur le minimum du critère pour une estimation des appariements donnée. Ce bruit apparaît lorsqu'on effectue plusieurs recalage avec des jeux de données différents, mais acquises dans les mêmes conditions. Il est malheureusement difficile de l'isoler expérimentalement, car le bruit interne est toujours présent.

Voici un exemple artificiel ; sur la figure 3.2, nous avons ajouté au critère de l'expérience

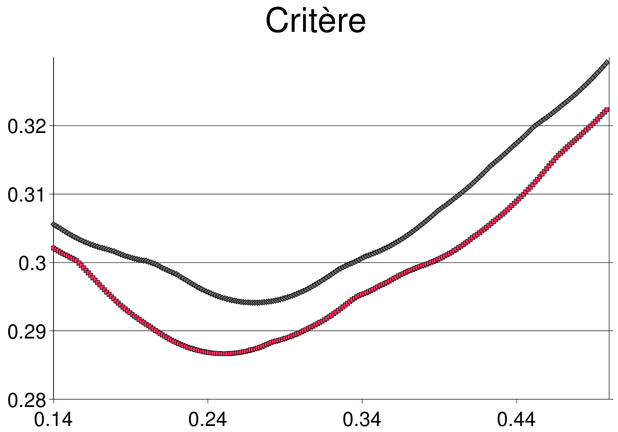


FIG. 3.2 – *Critère de l'ICP en fonction d'une translation en z : pour un nuage de point donné et le même nuage légèrement bruité.*

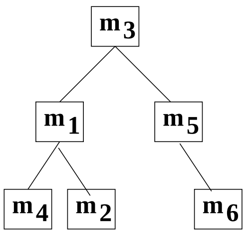
précédente le critère obtenu pour la même scène légèrement bruitée. L'estimation des appariements au minimum est inchangée, mais la position du minimum a été, comme on le voit, décalé vers la droite.

Incertitude globale On désignera par incertitude ou bruit global l'imprécision finale de l'algorithme, qui tient compte de toutes les sources d'erreurs possibles.

3.2.3 Mise à jour des appariements

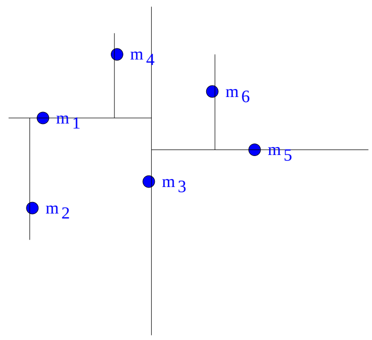
Nous nous intéressons ici à la mise en oeuvre de la phase de mise à jour des appariements, i.e. à l'estimation de A pour une transformation T connue. Ici, il s'agit de trouver, pour chaque point (primitive) de la scène dans la position courante ($T \star s_i$) le point du modèle le plus proche (m_j). La difficulté est essentiellement d'ordre algorithmique, et un très bon survol des techniques existantes peut être trouvé dans [Zhang, 1993]. Nous présentons le kD-Tree en détail.

kD-Tree On utilisera ici explicitement les coordonnées d'un point m_j. Elles seront notées : $(m_j^1, m_j^2, .., m_j^{\mathfrak{D}})$. L'idée de base du kD-Tree est de construire un arbre binaire dont chaque noeud représente un point du modèle. Cet arbre devra être à peu près équilibré, et adapté à la structure d'espace euclidien :



Ici, on dira que le noeud du point m_3 est de niveau 0. Les noeuds m_1 et m_5 sont de niveau 1, les noeuds restant de niveau 2. On définit le niveau d'un sous arbre comme le niveau du noeud qui en est le sommet. Ainsi l'arbre entier est de niveau 0, l'arbre contenant m_1, m_2 et m_4 est de niveau 1, ...

Construction L'idée est que, pour un noeud de niveau k et de point m_J, le sous arbre de gauche contienne l'ensemble des points A_g autre que m_J vérifiant $m_j^k \leq m_J^k$ (i.e. situes à gauche de m_J suivant la coordonnée k) et dans la sous arbre droit, l'ensemble des points A_d vérifiant $m_j^k > m_J^k$. Cette idée est en-fait analogue à celle utilisée dans certains algorithmes de tri rapide (de complexité en $n \log n$), sauf que la coordonnée utilisée pour le tri varie avec le niveau du tableau. Les points sont ainsi mieux répartis dans les différentes branches. Il faudra veiller à ce que le nombre de points dans chaque sous arbre soit relativement équilibré pour une véritable efficacité. Un tel arbre divise donc le plan de la façon suivante :



La construction de l'arbre se fait par une fonction récursive. Celle ci prend une sous partie A de M, cherche le point m_J partageant à peu près équitablement les points de A, crée le noeud représentant m_J, génère les sous-ensembles A_g et A_d, et s'appelle elle même avec ces sous-ensembles A_g et A_d pour générer les sous arbres attaches au noeud m_J.

Fonction Construit_arbre($A \subset M$,k)

a) rechercher un point m_J tel que

$$\mathrm{Card}(A_g = \{m_j \in A - \{m_J\} / m_j^k \leq m_J^k\}) \simeq \mathrm{Card}(A_d = \{m_j \in A - \{m_J\} / m_j^k > m_J^k\})$$

b) retourner l'arbre défini par le noeud supérieur constitué du point m_J, du sous arbre gauche construit avec Construit_arbre(A_g,$k+1$) et du sous arbre droit construit avec Construit_arbre(A_d,$k+1$).

Utilisation Une fois l'arbre construit, on voudra l'utiliser pour rechercher le plus proche voisin d'un point quelconque p. Le kD-Tree est en fait adapté à un problème légèrement différent, celui de la recherche des points dont la distance (euclidienne) avec p est inférieure à un seuil d_{max}.

En effet, lorsqu'on se situe au niveau d'un noeud de niveau k et de point m_J vérifiant $m_J^k < p^k - d_{max}$, on est sûr que pour tout point vérifiant $m_j^k \leq m_J^k$ et donc $m_j^k < p^k - d_{max}$, on aura $d(m_j,p) = \sqrt{\left\|(m_j^1 - p^1)^2 + (m_j^2 - p^2)^2 + .. + (m_j^{\mathfrak{D}} - p^{\mathfrak{D}})^2\right\|} \geq \left|m_j^k - p^k\right| > d_{max}$. On sait alors que le point m_J et les points du sous-arbre gauche ne conviendront pas, et on n'ira donc explorer ce sous arbre que si la condition $b_g = (m_J^k \geq p^k - d_{max})$ est vérifiée. Il en va de même avec le sous arbre droit et la condition $b_d = (m_J^k \leq p^k + d_{max})$. Enfin, il ne faudra tester le point m_J que si les deux conditions sont vérifiées.

La recherche des points vérifiant $d(m_j,p) \leq d_{max}$ se fait donc de manière récursive, par la fonction suivante :

Fonction Recherche_PointsProches(arbre de sommet m_J de niveau k, p, d_{max}, $var\ ListePoints$)

booléen $b_g = (p^k - d_{max} \leq m_J^k)$
booléen $b_d = (m_J^k \leq p^k + d_{max})$
si $((b_g) et (b_d))$ Rajoute m_J à $ListePoints$ si $d(m_J,p) < d_{max}$
si (b_g) Recherche_Points(sous arbre gauche de niveau $k+1$, p, d_{max})
si (b_d) Recherche_Points(sous arbre droit de niveau $k+1$, p, d_{max})

On peut facilement adapter cette fonction pour la recherche du seul point le plus proche, en passant d_{max} et $IndexPoint$ par référence. Ainsi, à chaque fois qu'un point est trouvé, on peut diminuer la distance maximale, en utilisant la distance entre le dernier point trouvé et m :

Fonction Recherche_PlusProchePoint(arbre de sommet m_J de niveau k, m, $var\ d_{max}$, $var\ IndexPoint$)

booléen $b_g = (p^k - d_{max} \leq m_J^k)$
booléen $b_d = (m_J^k \leq p^k + d_{max})$
si $((b_g) et (b_d) et (d(m_J,p) < d_{max}))$ $IndexPoint = J$ et $d_{max} = d(m_J,p)$
si (b_g) Recherche_PlusProchePoint(sous arbre gauche de niveau $k+1$, m, d_{max})
si (b_d) Recherche_PlusProchePoint(sous arbre droit de niveau $k+1$, m, d_{max})

Dans le cas où plusieurs points sont simultanément les plus proches, cette fonction ne permettra d'en connaître qu'un.

Efficacité L'algorithme ainsi présenté présente un temps moyen en $n \log n$ pour la construction et grossièrement en $d_{max}^2 \log n$ pour la recherche. Il est donc, pour un modèle donné particulièrement sensible à la distance maximale de recherche. Nous verrons comment cette distance maximale s'impose naturellement dans la gestion des points aberrants (partie 3.6).

Recherche incrémentale On peut se demander comment accélérer la recherche lorsque la distance maximale est grande voire infinie (lorsqu'on veut absolument trouver au moins un point). On a pour cela implémentée une solution qui consiste tout simplement à faire une première recherche avec une distance max assez faible, et relancer une nouvelle recherche en incrémentant la distance max tant qu'aucun point n'a été trouvé.

3.2.4 Mise à jour de la transformation

Nous supposons maintenant les appariements connus grâce à A, et nous voulons calculer la transfo qui minimise le critère C. Plusieurs techniques ont été développées, en fonction du type de primitives, certaines directes (SVD, quaternions unitaires, quaternions duaux), d'autres itératives (descente de gradient, filtrage de Kalman). Nous présenterons ici la SVD, et reviendrons plus loin sur les autres techniques (section 3.8).

La méthode SVD, introduite par [Arun *et al.*, 1987; Umeyama, 1991], est spécialement adaptée au cas où les primitives traitées sont effectivement des points. Rappelons la forme du critère $C(T)$:

$$C(R,t) = \sum_{ij} A_{ij}.\left\|Rs_i + t - m_j\right\|^2$$

Recherche de la translation La translation optimale est caractérisée par une dérivée nulle du critère : $\frac{\partial C}{\partial t} = 2.\sum_{ij} A_{ij}.(Rs_i + t - m_j)^t = 0$, donc $(\sum_{ij} A_{ij}).t = (\sum_{ij} A_{ij}.m_j) - R.(\sum_{ij} A_{ij}.s_i)$, et finalement :

$$\overline{m} = \frac{\sum_{ij} A_{ij}.m_j}{\sum_{ij} A_{ij}} \qquad \overline{s} = \frac{\sum_{ij} A_{ij}.s_i}{\sum_{ij} A_{ij}}$$
$$t = \overline{m} - R.\overline{s}$$

On peut remarquer que $\overline{m}$ (resp $\overline{s}$) sont les barycentres des points du modèle (resp. la scène) pondérés par A. Une fois calculés, nous pouvons passer les points de la scène et du modèle en coordonnées barycentriques : $s'_i = s_i - \overline{s}$ et $m'_j = m_j - \overline{m}$. Le critère se simplifie alors : $C'(R) = \sum_{ij} A_{ij}.\left\|Rs'_i - m'_j\right\|^2$, et permet le calcul de la rotation. Le calcul effectif de la translation ne se fera en-fait qu'après celui de la rotation.

Recherche de la rotation Le critère simplifié se ré-écrit $C(R) = \sum_{ij} A_{ij}.\left\|Rs'_i - m'_j\right\|^2 = \sum_{ij} A_{ij}.(\|Rs'_i\|^2 + \left\|m'_j\right\|^2 - 2(Rs'_i|m'_j)) = \sum_{ij} A_{ij}.(\|s'_i\|^2 + \left\|m'_j\right\|^2) - 2.Tr(\sum_{ij} R.A_{ij}.s'_i.m'^t_j)$. Comme $\|s'_i\|^2$ et $\left\|m'_j\right\|^2$ sont constants, il s'agit donc de maximiser

$G(R) = Tr(R.K'^t)$ où $K' = \sum_{ij} A_{ij}.m'_j.s'^t_i$ est la matrice de corrélation. Le maximum est alors atteint pour $R = U.S.V^t$, où $U.D.V^t = K'$ est une décomposition en valeur singulière de K' et $S = diag(1,...,1,det(U).det(V))$. Voir par exemple [Arun *et al.*, 1987; Umeyama, 1991] pour les détails.

Implémentation L'immense avantage de cette stratégie d'optimisation basée sur la SVD est qu'elle ne nécessite en fait pas d'effectuer le calcul des barycentres avant celui de la matrice de corrélation (et donc d'effectuer deux passes sur l'ensemble des appariements). En effet, si l'on calcule la matrice de corrélation en repère non barycentrique $K = \sum_{ij} A_{ij}.m_j.s_i^t$, on peut retrouver la matrice en repère barycentrique en écrivant : $K' = \sum_{ij} A_{ij}.(m_j - \overline{m}).(s_i - \overline{s})^t = K - \sum_{ij} A_{ij}.(m_j.\overline{s}^t + \overline{m}.s_i^t - \overline{m}.\overline{s}^t) = K - ((\sum_{ij} A_{ij}.m_j).\overline{s}^t + \overline{m}.(\sum_{ij} A_{ij}.s_i^t) - (\sum_{ij} A_{ij})\overline{m}.\overline{s}^t) = K - (\sum_{ij} A_{ij})(\overline{m}.\overline{s}^t)$.

L'optimisation se fait donc de la manière suivante : lors de la recherche des appariements, on accumule A_{ij}, $A_{ij}.m_j$, $A_{ij}.s_i$ et $A_{ij}.m_j.s_i^t$. Ensuite, on calcule dans l'ordre : les barycentres $\overline{m}$ et $\overline{s}$, la matrice de corrélation en repère barycentrique K', la rotation et finalement la translation.

Remarque Cette méthode a été présentée pour A_{ij} binaire. On vérifie cependant facilement qu'elle reste valable pour A_{ij} positif.

3.3 Initialisation

Tous les algorithmes de type ICP exigent une transformation initiale comme point de départ, et le résultat dépend de cette transformation initiale. En fait, elle conditionne le minimum local qui sera trouvé, qui est le minimum de la cuvette dans laquelle se trouve l'initialisation. Comme nous l'avons déjà dit, les minima manifestement faux sont éloignés du recalage recherché, et on les atteindra avec une mauvaise initialisation. Avec une bonne initialisation, on tombera dans l'un des nombreux minima situés à proximité de la bonne solution. Nous le répétons ici, cette variabilité des bonnes solutions ne doit pas être interprétée comme une conséquence de l'initialisation, mais plutôt comme le bruit interne de l'algorithme.

Nous présentons ici deux méthodes. La première, celle que nous utilisons, est basée sur un protocole spécifique, mais permet une bonne initialisation. La seconde est plus générale, mais très grossière, et en pratique inutilisable.

3.3.1 Utilisation d'amers géométriques

Cette méthode est en fait très proche de l'ICP. Il s'agit d'utiliser la partie "optimisation de la transformation" avec des points appariés connus, i.e. des amers géométriques.

Dans notre cas, nous utilisons les centres des dents, qui doivent être placés à la main sur l'image scanner ou sur le Picza, et interpolés à partir d'une première série de pointés effectués à la surface de ces dents pour l'imagerie per-opératoire. La précision obtenue est de l'ordre de $2mm$.

3.3.2 Barycentre et tenseur d'inertie

Cette méthode permet d'obtenir une initialisation très grossière, mais n'exige pas de protocole spécifique comme la précédente. Elle est inspirée de la physique des solides. Il s'agit de mettre en correspondance le barycentre et les axes d'inertie de la scène et du modèle.

On calcule donc le barycentre des deux ensembles de points $\overline{m} = \frac{1}{n_M}(\sum_j m_j)$ et $\overline{s} = \frac{1}{n_S}(\sum_i s_i)$, et leurs tenseur d'inertie $T_M = \sum_j(\overline{m}m_j \otimes \overline{m}m_j)$ et $T_S = \sum_i(\overline{s}s_i \otimes \overline{s}s_i)$. La rotation est alors trouvée en diagonalisant ces tenseurs d'inerties symétriques, et en mettant en correspondance les vecteurs propres (i.e. les axes d'inertie principaux) par ordre de valeurs propres (i.e. d'inertie) décroissantes. La translation est comme précédemment $t = \overline{m} - R\overline{s}$.

Cette méthode produit rarement des résultats satisfaisants, car elle est globale et donc sensible aux différences de structure entre les deux ensembles de points et aux artefacts. Ainsi, pour le recalage de la surface extraite d'un scanner sur une acquisition 2D 1/2 de la mâchoire, le résultat dépendra fortement du nombre de coupes utilisées lors de l'extraction, i.e. de la hauteur minimale et maximale de la portion de l'os extraite, et de la présence du palais sur l'acquisition 2D 1/2.

On peut robustifier cette méthode, tout du moins pour la translation, en utilisant les médianes à la place des moyennes.

3.4 Utilisation de points interpolés

Cette partie montre comment on peut "bricoler" les paires de points appariés en remplaçant les points du modèle par d'autres points interpolés. De telles opérations sont en général réalisées pour prendre en compte l'aspect surfacique des données lorsqu'on ne travaille pas avec des points homologues. Il faut cependant préciser que ces bricolages ne sont pas anodins d'un point de vue théorique car ils modifient le critère de façon différente à chaque itération et invalident ainsi la preuve de convergence de l'algorithme.

3.4.1 Projection simple

La façon la plus simple de modifier les paires de points appariés est de considérer chaque paire séparément. L'idée présentée ici peut être trouvée dans [Guéziec and Ayache, 1991; 1994]. Nous allons travailler dans le cas où *le modèle est une surface échantillonnée*, avec les normales associées à chaque point. Soit donc un appariement constitué du point m_j du modèle et du point s_i de la scène. Ces points ne seront en général pas homologues, biaisant ainsi l'algorithme. Nous allons donc essayer d'interpoler la position du point du modèle qui serait vraiment homologue à notre point de la scène. Pour cela, on peut utiliser la normale n_{m_j} au point m_j. Un tel point orienté permet de déduire une approximation plane de la surface sous-jacente. Il s'agit du plan $\Pi_j : \{x/((x - m_j).n_{m_j}) = 0\}$. Nous allons essayer de trouver sur cette approximation de la surface le point qui est le plus probablement le point homologue à notre point de la scène. Il s'agit bien-entendu du point le plus proche de $T \star s_i$, que l'on trouve facilement en effectuant la projection orthogonale de $T \star s_i$ sur le plan Π_j. Ce point, noté p_i est donné par la formule :

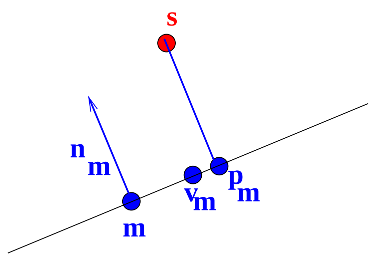
$$p_{m_j} = T \star s_i - ((T \star s_i - m_j).n_{m_j}).n_{m_j}$$

On remplacera dans la paire de point appariés le point du modèle m_j par un point virtuel défini par

$$v_{m_j} = C_{pr}.m_j + (1 - C_{pr}).p_{m_j}$$

où C_{pr} est un coefficient permettant de moduler le rôle du point réel (seul pris en compte lorsque $C_{pr} = 1$) et du projeté (seul pris en compte lorsque $C_{pr} = 0$).

Voici un exemple avec $C_{pr} \simeq 0.25$:



Les cas où C_{pr} est faible sont d'ailleurs pathologiques, car on peut alors favoriser des points très éloignés, mais bien alignés, au détriment de points plus proches. Cette remarque se confirme expérimentalement, et des mauvais recalages apparaissent lorsque $C_{pr} < 0.15$.

Dans tous les cas, les points sont modifiés différemment à chaque itération. On ne minimise donc plus un critère fixe, la convergence de l'algorithme n'est donc plus assurée et il peut même se révéler instable. On constate expérimentalement que le nombre d'itérations et donc le temps de calcul peut-être considérablement augmenté (de 50 à 500 pour un recalage scanner/Picza avec $C_{pr} = 0.5$). Nous présenterons une alternative à la section 5.3.2.

3.4.2 Projection double

Il existe aussi une méthode basée sur la projection double, introduite dans [Chen and Medioni, 1992] : lorsqu'on dispose des points munis de leurs normales sur les deux nuages, on calcule le projeté p_m de s sur le plan (m,n_m), puis le projeté p_s de p_m sur le plan (s,n_s), et on minimise la distance entre les deux projetés p_m et p_s. Cette méthode ne présente cependant pas de véritable amélioration vis à vis de la précédente, et nous ne la présenterons pas en détail.

3.5 Variantes sur la distance

Nous allons détailler ici les différentes variantes de l'ICP basées sur d'autres définitions de la distance entre deux points. Nous utiliserons en pratique la distance de Mahalanobis comme dans [Pennec, 1996], qui permet d'une part de prendre en compte le bruit sur les points (ce qui est particulièrement intéressant lorsque celui-ci est in-homogène ou anisotrope), et d'autre part de généraliser facilement l'algorithme à des types de primitives plus complexes.

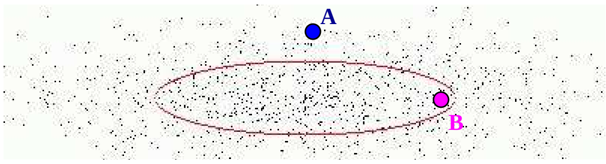
3.5.1 Distance de Mahalanobis simple

Lorsqu'on dispose d'une matrice de covariance sur les points de l'un des ensembles, on peut utiliser la distance de Mahalanobis simple pour la prendre en compte. Nous supposerons que le bruit est concentré sur les points du modèle : les points de la scène sont parfaitement connus, et chaque point du modèle m_j est altéré par un bruit blanc gaussien centré caractérisé par une matrice de covariance $\Sigma_{m_j m_j}$.

La loi de probabilité sur un vecteur aléatoire gaussien $\mathbf{x}$ de moyenne $\overline{x}$ et de covariance Σ_{xx} dans un espace de dimension $\mathfrak{D}$ est définie par une densité de la forme :

$$p(x) = \frac{\exp(-\frac{\overrightarrow{\overline{x}x}.\Sigma_{xx}^{-1}.\overrightarrow{\overline{x}x}^t}{2})}{\sqrt{(2\pi)^{\mathfrak{D}}.\,|\Sigma_{xx}|}}$$

On voit alors dans le terme $\overrightarrow{\overline{x}x}.\Sigma_{xx}^{-1}.\overrightarrow{\overline{x}x}^t$ une sorte de distance quadratique prenant en compte la covariance, qui permet de quantifier la distance du vecteur x au vecteur aléatoire gaussien $\mathbf{x}$. Ainsi, sur la figure suivante, on a représenté la covariance par une ellipse :



Le point A est plus proche de $\overline{x}$ que le point B, au sens habituel du terme; il est pourtant plus "éloigné" (au sens plus improbable) de notre point aléatoire $\mathbf{x}$, et sa distance de Mahalanobis sera plus importante.

La distance de Mahalanobis entre un point de la scène s_i transformé par T et un point du modèle m_j de covariance $\Sigma_{m_j m_j}$ se définit donc comme suit :

$$\mu^2(T \star s_i, m_j) = (T \star s_i - m_j)^t . \Sigma_{m_j m_j}^{-1} . (T \star s_i - m_j) \tag{3.7}$$

Dans le cas d'un bruit isotrope ($\Sigma_{m_j m_j} = \sigma_{m_j}^2 . Id$), cette distance est en fait la distance quadratique classique normalisée par la variance :

$$\mu^2(T \star s_i, m_j) = \frac{1}{\sigma_{m_j}^2} \|T \star s_i - m_j\|^2 \tag{3.8}$$

Il est alors particulièrement facile d'intégrer cette distance dans notre algorithme d'optimisation utilisé pour la calcul de la nouvelle distance. Le critère est en effet devenu $C(R,t) = \sum_{ij} (A_{ij}/\sigma_{m_j}^2) . \|Rs_i + t - m_j\|^2$, et il suffit tout simplement de remplacer A_{ij} par $A_{ij}/\sigma_{m_j}^2$ dans les raisonnements de la section 3.2.4.

La distance de Mahalanobis présente les intérêts suivants :

- Elle permet de prendre en compte les covariances, et donc de mieux traiter le cas non-homogène (tous les points n'ont pas la même covariance) et le cas non-isotrope (le bruit de mesure sur un point n'est pas isotrope).
- Dans un espace vectoriel et pour un vecteur aléatoire gaussien, elle suit une loi du χ^2 à $\mathfrak{D}$ degrés de libertés, dont le densité et l'espérance sont données par :

$$p_{\chi^2_{\mathfrak{D}}}(u) = \frac{(\frac{u}{2})^{\frac{\mathfrak{D}}{2}-1} . \exp(-\frac{u}{2})}{2.\Gamma(\frac{\mathfrak{D}}{2})}$$

$$E\left[\chi^2_{\mathfrak{D}}\right] = \mathfrak{D}$$

- Elle permet d'écrire simplement la densité de probabilité :

$$p(x) = \frac{\exp(-\frac{\mu^2(\overline{x},x)}{2})}{Cte}$$

- Elle est sans unité et permet donc de mélanger plusieurs types de primitives.
- Elle se généralise facilement aux espaces non-vectoriels (voir D.2.3.3).

3.5.2 Utilisation des invariants

L'ICP peut aussi être utilisé lorsque nos points sont affublés d'informations invariantes vis a vis de la transformation recherchée [Feldmar, 1995; Feldmar and Ayache, 1996]. Ces informations seront par exemple les informations différentielles d'ordre 2 sur la surface ou la courbe sous-jacente en chaque point. En effet, les techniques de segmentation d'images volumiques permettent souvent de calculer ces informations avec précision.

De par leur invariance, cette information supplémentaire n'interviendra pas directement dans la mise à jour de la transformation. Elle permettra par contre d'affiner et robustifier la recherche des points appariés et leur pondération, en prenant en compte par exemple, la forme locale de la surface sous-jacente. Ceci sera particulièrement intéressant dans le cas non-rigide, où l'on a besoin d'appariements très robustes pour optimiser une fonction ayant beaucoup de degrés de liberté.

3.5.3 Généralisation au recalage iconique

On peut généraliser l'ICP au recalage iconique [Feldmar, 1995; Feldmar *et al.*, 1997b] en intégrant l'intensité dans les coordonnées des points : (x,y,z,i). Il s'agit d'adapter l'algorithme en 4 dimensions. La transformation est alors constituée de la transformation 3D, et d'une transformation sur l'échelle d'intensité entre les images permettant de prendre en compte les différences de calibration dans le cas de l'IRM par exemple, voire le lien entre les intensités dans le cas multi-modal.

3.6 Gestion des point aberrants

Rappelons que, dans notre application, le modèle compte une centaine ou une cinquantaine de milliers de points (la surface extraite du scanner ou la Picza), et la scène une cinquantaine de milliers de points (dans le cas d'une acquisition 2D 1/2 de la mâchoire) ou quelques dizaines de points (dans le cas des séries de pointés). Tous ces points de la scène ne correspondent pas nécessairement à un point du modèle (dans le premier cas, on évalue le pourcentage de points de la scène appariés à 60%, mais il est très variable en fonction des images), et cette partie présente différentes techniques permettant de rejeter les faux positifs, c'est à dire les points de la scène qui ne devraient pas être appariés. Toutes ces méthodes sont basés sur le même principe, que nous décrirons dans la première section. Ce principe fait intervenir un seuil sur la distance, et les sections suivantes présentent différentes façon de le calculer. Ceci est en outre parfaitement en accord avec les algorithmes de recherche des points les plus proches, sur lesquels sont basés l'implémentation, dont l'efficacité dépend principalement de la distance maximum qu'on lui impose (voir 3.2.3).

Nous ne traiterons pas le cas des faux-négatifs, c'est à dire des appariements non trouvés. En effet, les méthodes que l'on pourrait imaginer exigeraient d'augmenter considérablement la distance de recherche, et seraient donc bien trop coûteuses en terme de temps de calcul.

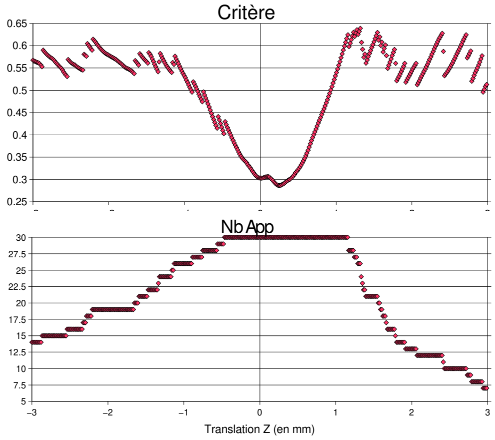


FIG. 3.3 – *Critère de l'ICP en fonction d'une translation en z : cas où l'on rejette les points trop éloignés : les variations des appariements (en bas) entraînent des discontinuités dans le critère (en haut).*

3.6.1 Schéma général

Une première forme de gestion des points aberrants peut être introduite lors de la recherche des appariements, comme dans [Zhang, 1994]. Il s'agit de rejeter les appariements manifestement faux, et améliorer ainsi la robustesse de l'algorithme. L'idée est que - pour une transformation donnée - la distance entre points (primitives) véritablement appariés doit être relativement homogène, et doit donc être inférieure à une certaine distance D_{max}. L'estimation de A est donc légèrement modifiée :

$$A'_{ij} = \begin{cases} 0 \quad \text{si} \quad D(T \star s_i, m_j) \geq D_{max} \\ \qquad A_{ij} \quad \text{sinon} \end{cases}$$

Ce changement peut cependant poser problème. En effet, le nombre d'appariements n'étant plus fixe, il peut apparaître des discontinuités dans le critère, comme illustré dans la figure 3.3. Le danger est alors de considérer les discontinuités comme des minima locaux.

On peut corriger ce problème en intégrant dans notre critère des pseudo-appariements : on représentera le cas où s_i est un point aberrant par un pseudo-appariement noté abusivement (pour ne pas surcharger les notations) $s_i \sim m_{out}$. Cet appariement sera représenté dans l'estimation de A par $A_{i\,out}$ et sera affublé d'une distance constante égale à :

$$D(T \star s_i, m_{out}) = D_{max}$$

Sauf mention contraire, l'index j variera dans l'ensemble $\{1,..,M,out\}$. Les critères deviennent donc :

$$C(T) = \sum_i \min_j D(T \star s_i, m_j)$$

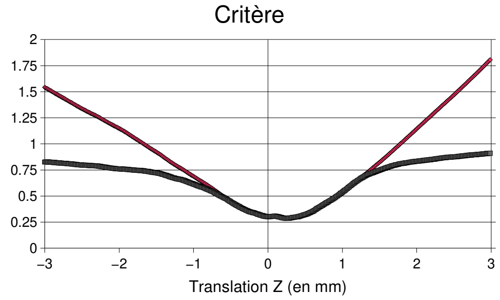


FIG. 3.4 – *Critère de l'ICP en fonction d'une translation en z : cas où l'on corrige le rejet des points trop éloignés : le critère redevient continu.*

$$C(T,A) = \sum_{ij} A_{ij} D(T \star s_i, m_j)$$

Ainsi corrigés, les critères ne perdent aucune de leurs propriétés théoriques. De plus, la minimisation de ce critère à T constant se fait sous la contrainte $\sum_j A_{ij} = 1$, et donne le point le plus proche si celui-ci est à une distance inférieure à D_{max} et un pseudo-appariement de type point aberrant dans le cas contraire, ce qui correspond bien à notre idée. Enfin, la distance $D(T \star s_i, m_{out})$ est constante, si bien que les points aberrants n'interviennent en fait pas dans la minimisation du critère à A constant. Les différents algorithmes de mise à jour de T peuvent donc être utilisés en ignorant simplement les pseudo appariements avec point aberrant.

On peut voir sur la figure 3.4 le critère sans rejet des points aberrants (courbe supérieure) et le critère avec rejet des points aberrants (avec $D_{max} = 1mm$).

Une dernière remarque importante est la suivante : D_{max} peut être remis à jour entre deux itérations. Il doit alors être décroissant - au sens large - pour ne pas faire remonter le critère.

3.6.2 Calcul la distance de coupure D_{max}

L'équation $\min C(T) \simeq \mathfrak{D}.\sigma_{\mathbf{e}}^2$, obtenue dans la section 3.2.1 permet de se faire une idée de la distance moyenne attendue lorsque le recalage est bon. On note cette distance D_{th}. Il faudra bien sûr utiliser une valeur bien plus grande en début de recalage. Les deux méthodes que nous allons voir ici pour calculer D_{max} sont d'ailleurs basées sur une mise à jour entre chaque itération, si bien qu'il faut fournir une valeur initiale pour la première itération, notée D_{init}. On prendra donc une valeur assez forte, par exemple $D_{init} = 10.D_{th}$.

De plus, il faudra veiller à ce que D_{max} soit décroissant. Dans le cas contraire, il pourrait faire remonter le critère.

Analyse statistique des distances

Nous présentons ici le calcul de la distance maximum utilisé dans [Zhang, 1994].

Ce calcul est basé sur une étude statistique des distances entre points appariés. On définit ainsi la moyenne et la variance de ces distances (on utilise la distance simple $d = \|\|$ et non la distance quadratique $D = d^2 = \|\|^2$) :

$$\mu = \frac{1}{\sum_{ij} A_{ij}} \sum_{ij} A_{ij} d(T \star s_i, m_j)$$

$$\sigma = \sqrt{\frac{1}{\sum_{ij} A_{ij}} \sum_{ij} A_{ij} (d(T \star s_i, m_j) - \mu)^2}$$

La mise à jour de D_{max} entre deux itérations est effectuée de façon très pragmatique, à l'aide de μ : faible lorsque le recalage semble bon ($\mu < D_{th}$), très forte lorsque il est très mauvais ($\mu > 6D_{th}$) :

$$D_{max} = \begin{cases} \mu + 3\sigma & \text{si } \mu < D_{th} \\ \mu + 2\sigma & \text{si } \mu < 3D_{th} \\ \mu + \sigma & \text{si } \mu < 6D_{th} \\ D_{init} & \text{sinon} \end{cases}$$

On le voit, cette méthode manque de justification, et nous lui préférerons la méthode qui suit, mieux justifiée, et plus convainquante d'un point de vue expérimental.

Distance de Mahalanobis et Test du χ^2

Lorsqu'on utilise la distance de Mahalanobis, le calcul de $D_{max} = \mu^2_{max}$ fait penser au test du χ^2. En effet, comme nous l'avons déjà précisé, la distance de Mahalanobis pour un vecteur gaussien en dimension $\mathfrak{D}$ suit une loi du χ^2 à $\mathfrak{D}$ degré de liberté. Autrement dit, si un point de la scène s_i est vraiment apparié à un point du modèle m_j, la distance de Mahalanobis $\mu^2(T \star s_i, m_j)$ est la réalisation d'une variable réelle aléatoire dont la loi suit un χ^2 à $\mathfrak{D}$ degré de liberté. Or la distribution d'une telle loi est bien connue, et résumée dans le tableau suivant :

Dimension	α	50%	90%	95%	99%
3 (points 3D)	ϵ	2.37	6.25	7.81	11.34
5 (pts orientes 3D)	ϵ		9.24	11.07	15.09
6 (transfos rigides)	ϵ	5.35	10.65	12.59	16.81

Ce tableau se lit de la façon suivante : pour une dimension donnée, α est la probabilité que la distance de Mahalanobis soit inférieure à ϵ. Par exemple, lorsqu'on travaillera sur les points en 3D, on est sûr à 90% que deux points appariés ont une distance de Mahalanobis inférieure à 9.24. On rejettera donc par exemple les points dont la distance de Mahalanobis est supérieure à cette limite, en fixant $\mu^2_{max} = 9.24$.

3.7 Utilisation d'appariements pondérés multiples

Cette partie s'intéresse à différentes manières de pondérer les appariements obtenus. Nous avons jusqu'ici utilisé un poids unitaire pour le point le plus proche, et un poids nul pour les autres. Ceci est le principal responsable des problèmes de répétabilité et robustesse de l'ICP. En effet, le critère n'est lisse que par morceaux, il n'est lisse que dans les zones où l'estimation des appariements représentée par A est constante. Mais chaque changement dans l'estimation de A traduit un changement brusque du point du modèle apparié à un point de la scène, et donc un changement brusque du comportement du critère, qui fait apparaître de multiples minima locaux.

Les techniques qui suivent visent à éliminer ce problème en introduisant au final un poids non-entier - et plus régulier vis à vis de la transformation - aux différents points du modèle appariables à un point de la scène, au lieu de ne garder que le plus proche. Elles introduiront de gros changements dans la première étape de l'algorithme, mais elles n'introduiront aucun changement dans la mise à jour de la transformation, car nos techniques de calculs de la transformation fonctionnent avec des poids positifs (voir remarque à la fin de 3.2.4).

Les techniques présentées ici ont toutes été introduites par Rangarajan et al., à partir de critères modifiés. Nous reviendrons dans le chapitre suivant sur l'une d'elle, l'EM, en justifiant les changements apportés au critère.

3.7.1 EM

L'algorithme EM tel que présenté ici a été introduit dans le cadre du recalage de nuages de points par Rangarajan [Chui and Rangarajan, 2000]. La première idée est d'utiliser des poids d'appariements non binaires : A_{ij} peut maintenant prendre n'importe quelle valeur de l'intervalle [0,1]. Le poids total pour un point de la scène doit toujours être égal à 1 : $\sum_j A_{ij} = 1$. Le critère à minimiser est le critère de l'ICP auquel on ajoute l'entropie des poids A_{ij} :

$$C(T,A) = \sum_{ij} A_{ij} D(T \star s_i, m_j) - \alpha \sum_{ij} A_{ij} \log A_{ij}$$

On minimise alors ce critère par la méthode des multiplicateurs de Lagrange, pour prendre en compte la contrainte $\sum_j A_{ij} = 1$. On obtient alors pour l'étape d'estimation de A :

$$A_{ij} = \frac{\exp\left(-D(T \star s_i, m_j)/\alpha\right)}{\sum_k \exp\left(-D(T \star s_i, m_k)/\alpha\right)}$$

On obtient donc des poids gaussiens normalisés. Le problème avec cette façon de présenter l'EM est qu'elle ne donne aucun sens au paramètre α, et donc aucune manière de le fixer. Nous reviendrons sur ce point au chapitre suivant.

Une autre idée introduite par Rangarajan est celle de recuit-simulé : il s'agit de donner une grande valeur à α en début de recalage, et diminuer celle-ci ensuite. Ceci permet d'augmenter la robustesse en début de recalage.

3.7.2 EM Symétrisé

L'idée, introduite sous le nom de SoftAssign [Rangarajan *et al.*, 1997a], est de symétriser l'EM en faisant jouer exactement le même rôle à la scène et au modèle, pour que le résultat du recalage soit identique lorsqu'on inverse scène et modèle. Pour cela, il faut tout d'abord gérer les points aberrants de la même façon pour les deux ensembles. On introduit donc un pseudo-appariement de la forme $s_{out} \sim m_j$ pour signifier que le point m_j est un point aberrant. On introduit de même $A_{out\,j}$. Dans les notations qui suivent, sauf mention contraire, i décrit l'ensemble $\{1,..,n_S,out\}$ et j l'ensemble $\{1,..,n_M,out\}$. Le couple (i,j) ne sera jamais être égal à (out,out), ce qui n'aurait aucun sens.

La configuration moyenne d'appariements obéit maintenant à la double contrainte :

$$\begin{cases} \forall i \neq out & \sum_j A_{ij} = 1 \\ \forall j \neq out & \sum_i A_{ij} = 1 \end{cases} \tag{3.9}$$

Le critère est le même que précédemment. Rangarajan explique, dans [Rangarajan *et al.*, 1997b], comment optimiser ce critère par rapport à A sous la double contrainte. Tout d'abord, en utilisant la méthode des multiplicateurs de Lagrange, on trouve :

$$A_{ij} = \frac{\exp\left(-D(T \star s_i, m_j)/\alpha\right)}{\nu_i . \rho_j}$$

Les constantes ν_i et ρ_j doivent permettre de vérifier la double contrainte 3.9. On ne peut hélas les calculer directement, mais seulement de façon itérative, grâce au théorème de Sinkhorn. On pose $A^0_{ij} = \exp\left(-D(T \star s_i, m_j)/\alpha\right)$, soit $\nu^0_i = 1$ et $\rho^0_j = 1$. A chaque itération I, on calcule $A^{I+1/2}_{ij}$ en normalisant les colonnes de A^I_{ij}, puis A^{I+1}_{ij} en normalisant les lignes de $A^{I+1/2}_{ij}$:

$$A^{I+1/2}_{ij} = \frac{A^I_{ij}}{\sum_j A^I_{ij}} \qquad \text{soit} \quad \nu^{I+1}_i = \sum_j \frac{\exp\left(-D(T \star s_i, m_j)/\alpha\right)}{\rho^I_j}$$

$$A^{I+1}_{ij} = \frac{A^{I+1/2}_{ij}}{\sum_i A^{I+1/2}_{ij}} \qquad \text{soit} \quad \rho^{I+1}_j = \sum_i \frac{\exp\left(-D(T \star s_i, m_j)/\alpha\right)}{\nu^{I+1}_i}$$

Le théorème de Sinkhorn montre que le résultat converge vers une matrice vérifiant la contrainte.

Discussion Cette méthode est coûteuse à implémenter. On est en effet obligé de stocker l'ensemble des poids des appariements pendant chaque itération, et appliquer le théorème de Sinkhorn, ce qui nécessite de nombreux calculs.

De plus, les hypothèses doivent aussi être symétriques : la scène doit être sous-échantillonné par rapport au modèle (hypothèse que nous avons toujours faite), et le modèle doit donc être aussi sous-échantillonné par rapport à la scène, et on doit finalement avoir deux ensembles équivalents. Nous avons vérifié expérimentalement que sur des ensembles trop dissemblables (comme les nôtres), l'algorithme se comportait très mal. On retrouve les mêmes problèmes avec d'autres algorithmes symétriques (notamment la variante de l'ICP basée sur l'information mutuelle [Rangarajan *et al.*, 1999]).

3.7.3 Entropie

Cette autre version symétrique, proposée dans la première partie de [Rangarajan *et al.*, 1999] est basée toujours sur le même critère. Cependant la contrainte de normalisation fait cette fois intervenir la totalité des appariements :

$$\sum_{ij} A_{ij} = 1$$

Les poids sont alors donnés par la formule suivante :

$$A_{ij} = \frac{\exp(-D(T \star s_i, m_j)/\alpha)}{\sum_{kl} \exp(-D(T \star s_l, m_k)/\alpha)}$$

3.7.4 Discussion

Nous avons comparé expérimentalement la précision (par la méthode de l'étalon zinc, voir section 6.3.2) et la robustesse (par la méthode des cartes de convergence, voir section 6.2) de ces différentes techniques sur des recalages entre nuages de Pointés per-opératoires en mode contact (voir section 1.5.3) contenant quelques dizaines de points et des acquisitions Picza à différentes résolutions (voir section 1.5.2).

Les résultats présentés dans la figure 3.5 montrent deux choses :

- En ce qui concerne la précision, l'EM semble le meilleur. La raison profonde est que sa répétabilité (voir 6.3.1) est quasi-nulle, contrairement à celle de l'ICP qui est une source notable d'imprécision.
- Les algorithmes basés sur des poids non-unitaires ont des comportements très proches. Ils ont permis une nette amélioration de la robustesse. L'entropie est dans

tous les cas plus mauvais que l'EM. Ceci peut s'expliquer par le fait que tous les points de la scène ne se voient plus attribuer le même poids total $\sum_j A_{ij}$. Ainsi, les points qui ont beaucoup de voisins sont favorisés vis à vis des points qui en ont peu, déséquilibrant ainsi l'importance de chaque point de la scène.

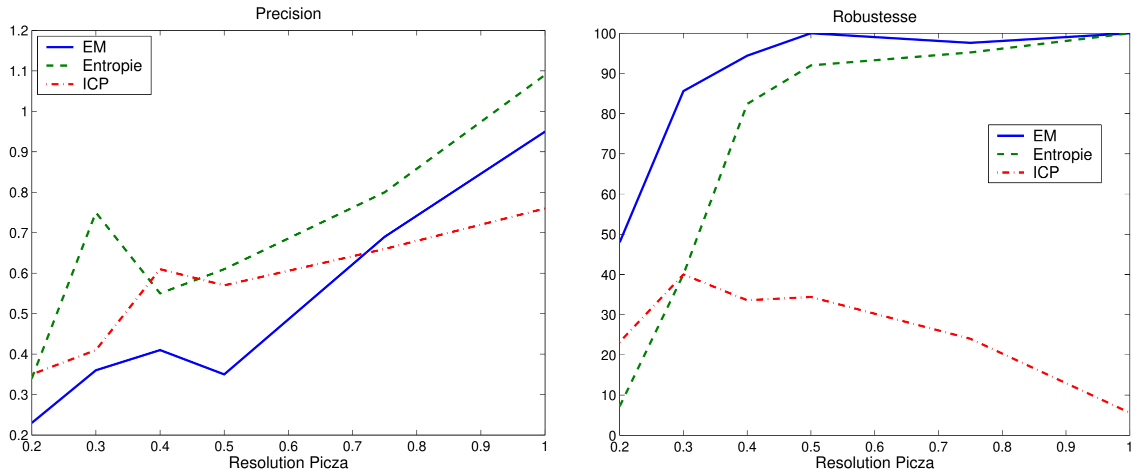


FIG. 3.5 – *Précision (écart-type en mm) et robustesse (pourcentage de convergences correctes) des algorithmes ICP, EM et Entropie.*

Notre conclusion est que l'EM est l'algorithme le plus prometteur pour les données sur lesquelles nous travaillons.

3.8 Variantes sur les méthodes d'optimisations

Il existe de nombreuses méthodes permettant, une fois le critère écrit et les pondérations A_{ij} calculées de calculer la nouvelle transformée. Certaines sont directes, plus spécifiques mais plus rapides, d'autres sont itératives, plus générales et plus intéressantes d'un point de vue théorique, comme la descente de gradient ou le filtrage de Kalman.

Notons qu'il existe aussi des méthodes qui permettent de réduire le nombre d'itération et donc d'accélérer la convergence [Besl and McKay, 1992]. Nous ne nous y sommes pas intéressé car le gain en temps de calcul est de l'ordre de 2, ce dont nous n'avons pas eu besoin.

3.8.1 Méthodes directes

Il existe d'autres méthodes que la SVD (section 3.2.4) permettant le calcul direct de la transformation : elles sont basées sur les matrices orthogonales [Horn *et al.*, 1988], les quaternions unitaires [Faugeras and Hebert, 1986; Horn, 1987] ou les quaternions duaux [Walker *et al.*, 1991]. Nous ne détaillerons pas ces méthodes qui s'appliquent uniquement aux points et donnent des résultats comparables, en terme de rapidité et de précision, à la méthode basée sur la SVD [Eggert *et al.*, 1997; Schonenenberger, 1998].

3.8.2 Méthode itérative : descente de gradient de type Newton

Gradient du critère

Dans cette méthode, nous aurons besoin du gradient du critère $C(T)$ que nous allons ici calculer. Pour alléger les (nombreuses) équations qui suivent, nous utiliserons les notations abrégées :

$$s_{Ti} = T \star s_i \quad \text{et} \quad A_T = A(T) \quad \text{et donc} \quad A_{Tij} = A_{ij}(T)$$

Nous ferons ici abstraction des changements brusque de la configuration d'appariement, en considérant que A_T reste constant pour de petite variations autour de la transformation. Nous effectuerons le calcul pour le critère écrit de la manière suivante :

$$C(T) = \frac{1}{2} \sum_{ij} A_{Tij}.D(s_{Ti},m_j)$$

On a donc :

$$\frac{dC(T)}{dT} = \frac{1}{2} \sum_{ij} A_{Tij}.\frac{dD(s_{Ti},m_j)}{dT} = \frac{1}{2} \sum_{ij} A_{Tij}.\left(\frac{dD(s_{Ti},m_j)}{ds_{Ti}}\right).\left(\frac{ds_{Ti}}{dT}\right)$$

Le cas où $j = out$ est simple. En effet $D(s_i,m_{out})$ est constant et donc de gradient nul. Reste à calculer le gradient de la distance de Mahalanobis pour de vrais appariements. On notera $e_{ij} = T \star s_i - m_j$ le vecteur d'erreur. La distance de Mahalanobis se ré-écrit alors : $\mu^2(T \star s_i,m_j) = e_{ij}^t.\Sigma_{e_{ij}e_{ij}}^{-1}.e_{ij}$ avec

$$\Sigma_{e_{ij}e_{ij}} = R.\Sigma_{s_i s_i}.R^t + \Sigma_{m_j m_j}$$

si les bruits sur les points de la scène et du modèle ne sont pas corrélés (ce qui est en général le cas). Le gradient de cette distance par rapport à un point est :

$$\frac{d\mu^2(s_{Ti},m_{out})}{ds_{Ti}} = 2.e_{ij}^t.\Sigma_{e_{ij}e_{ij}}^{-1}$$

et finalement, le gradient du critère s'écrit :

$$\frac{dC(T)}{dT} = \sum_{i\,j \neq out} A_{Tij}.e_{ij}^t.\Sigma_{e_{ij}e_{ij}}^{-1}.\frac{de_{ij}}{dT}$$

Principe

On notera le gradient $\Phi(T) = (\frac{dC(T)}{dT})^t$. Il doit être nul au minimum du critère. L'idée de la descente de gradient de type Newton est d'approcher ce point nul en effectuant une succession d'approximations linéaires. L'approximation linéaire en T^I de $\Phi(T)$ est donnée par $\Phi(T) = \Phi(T^I) + H(T^I).(T - T^I)$, où $H = \frac{d\Phi}{dT}$. Le point nul de cet approximation linéaire est $T^{I+1} = T^I - H(T^I)^{-1}.\Phi(T^I)$.

Le Hessien $H(T)$ sera approximé en négligeant la dérivé de A_{Tij} (nulle presque partout pour l'ICP classique) :

$$H = \frac{d(\frac{dC(T)}{dT})^t}{dT} = \frac{1}{2}\sum_{ij} A_{Tij}.H_T D(s_{Ti},m_j)$$

on calcule le hessien de la distance en négligeant la dérivé seconde par rapport à T :

$$H_T D(s_{Ti},m_j) = \left(\frac{ds_{Ti}}{dT}\right)^t (H_{s_{Ti}} D(s_{Ti},m_j)) \left(\frac{ds_{Ti}}{dT}\right)$$

dans le cas de la distance de Mahalanobis, le hessien de la distance par rapport à un point est :

$$H_{s_{Ti}} \mu^2(s_{Ti},m_j) = 2.\Sigma^{-1}_{e_{ij}e_{ij}}$$

soit finalement :

$$H \simeq \sum_{ij} A_{ij}.\left(\frac{de_{ij}}{dT}\right)^t.\Sigma^{-1}_{e_{ij}e_{ij}}.\left(\frac{de_{ij}}{dT}\right)$$

Discussion

On peut se demander combien d'itérations de la descente de gradient on va réaliser à chaque étape d'estimation de la transformation. En-effet, nous avons jusqu'ici présenté cette dernière comme une minimisation du critère, mais une simple diminution de celui-ci assure la convergence (certes moins rapide) de l'ICP. On n'est donc pas forcément obligé de faire tourner la descente de gradient jusqu'à sa convergence. Faire une seule itération a l'immense avantage de se résumer à l'accumulation du gradient et du hessien du critère pour la transformation courante lors du calcul des appariements et ne nécessite donc pas leur stockage. Mais lorsque les appariements ne varient plus (typiquement en fin de recalage), les stocker est certainement plus efficace que de les recalculer à chaque itération. Tout compte fait, nous avons privilégié la première approche, plus simple.

La méthode de la descente de gradient présente l'intérêt de se généraliser à n'importe quel type de primitive. Nous l'utiliserons dès que les méthodes directes ne seront plus applicables[9].

9. Nous l'avons aussi testé sur les points, pour la comparer aux méthodes directes. Ceci à permis de

3.8.3 Méthode itérative : filtrage de Kalman

Le filtrage de Kalman, dans sa version étendue, semble aussi être parfaitement adapté à notre problème et d'autant plus intéressant qu'il fournit directement une covariance sur la transformation obtenue, qui permet d'estimer l'incertitude du recalage [Ayache, 1991]. Mais si on réfléchit à l'algorithme et ses fondements statistiques, on se rend compte que la seule différence avec notre descente de gradient et notre méthode de prédiction de l'incertitude est la prise en compte d'une covariance sur la transformation initiale, bien inutile dans notre cas car elle est très grande devant la covariance finale du recalage, et peut même générer des instabilités numériques.

3.9 Prédiction théorique de l'incertitude

Comme nous l'avons dit dans la section 3.2.1, l'incertitude sur la transformation finale est la conséquence de deux types de bruit, le bruit interne et externe. Cette partie présente les travaux de [Pennec, 1996] qui permettent la prédiction théorique du bruit externe de l'ICP (i.e. la conséquence du bruit sur les données) pour le recalage d'amers géométriques. Précisons que [Kanatani, 1996] a trouvé parallèlement la même formule pour la limite de Cramer-Rao (i.e. la meilleure précision que l'on puisse espérer pour un tel recalage, quel que soit l'algorithme utilisé, cf eq. C.3). En tenant compte de ces deux séries de travaux, on comprend que l'ICP est l'algorithme optimal[10]. Enfin, notons que [Fitzpatrick *et al.*, 1999] a présenté une autre méthode, plus heuristique et moins générale.

3.9.1 Principe

L'idée fondatrice du calcul qui suit est d'écrire le résultat du recalage comme une fonction φ des données (les points du modèle et de la scène) et d'examiner comment se propage le bruit des données par cette fonction, i.e. de calculer comment les erreurs de mesures sur les points perturbent, en moyenne, le résultat du recalage.

Il s'agit d'un calcul au premier ordre, au sens où il utilise une approximation linéaire de la fonction φ et suppose les bruits gaussiens. Ces approximations seront bonnes lorsque le bruit ne sera pas trop important.

---

vérifier que les résultats étaient identiques lorsqu'on attendait la convergence de la descente de gradient et très proches lorsqu'on la limitait la descente de gradient à une seule itération (le recalage est cependant alors à peu près deux fois plus long).

10. Ce résultat n'est pas surprenant : dans les problèmes de régression linéaire classique, on retrouve la même égalité entre limite de Cramer-Rao (i.e. meilleure précision possible) des estimateurs et précision théorique de l'estimateur aux moindres carrés. Cette égalité prouve alors que l'estimateur aux moindres carrés est le meilleur. Ceci est illustré dans le cas du calcul de la moyenne dans les annexes (voir C.1.4 et C.1.3).

Il s'agit de plus d'un calcul qui suppose que *le recalage aurait donné un résultat exact en l'absence de bruit sur les données*. Il faut pour cela que les points soient effectivement homologues, et que les appariements soient connus ou correctement estimés.

Les deux éléments de base pour ce calcul sont :

- La fonction φ : la méthode de recalage minimise un critère, et son résultat peut ainsi être exprimée comme une fonction implicite des données, dont les dérivées (qui servent dans les calculs qui suivent) sont calculables à partir des dérivés du critère par le théorème des fonctions implicites. Ainsi, ce qui suit peut s'appliquer quelle que soit la stratégie d'optimisation adoptée (même si certains calculs font fortement penser à la descente de gradient), dès lors qu'un minimum du critère a été effectivement atteint.
- Le modèle de bruit sur les données : le calcul qui suit s'appuie sur la covariance de ce bruit, qui doit donc être connue ou bien estimée. Cette covariance regroupe la covariance du bruit sur chaque point et les covariances croisées des différents points. Nous supposerons en fin de calcul que ces dernières sont nulles (i.e. que les bruits sur les différents points sont indépendants, ce qui est en général le cas avec des amers géométriques), pour obtenir une formule simple. Mais on pourrait prendre en compte un modèle de bruit faisant intervenir une corrélation entre les points, pourvue qu'elle soit connue au travers des covariances croisées.

Nous allons ici redonner les détails mathématiques des deux étapes permettant de mener à bien le calcul. La première explique comment propager l'erreur par la fonction φ, et la seconde montre comment définir cette fonction pour un algorithme de minimisation de critère. Ce raisonnement est en-fait très général, et a été appliqué à de très nombreux problèmes s'appuyant sur une minimisation de critère.

Propagation de l'incertitude pour une fonction C^1 Nous allons maintenant examiner le cas d'un objet probabiliste défini comme le résultat de l'application d'une fonction C^1 à un autre objet probabiliste.

Examinons d'abord le cas affine bijectif : y est défini à partir de x par : $y = Bx + t$ où B est une matrice inversible et t un vecteur. La densité de y est donnée par $p_y(y) = |B|^{(-1)} .p_x(B^{(-1)}.(y-t))$. Ceci permet de calculer l'espérance et la covariance de y :

$$\overline{y} = E\left[y\right] = \int y.p_x(B^{(-1)}.(y-t))\frac{dy}{|B|} = \int (Bx+t).p_x(x).dx = B\overline{x} + t$$

$$\Sigma_{yy} = E\left[(y-\overline{y})(y-\overline{y})^t\right] = \int (Bx - B\overline{x})(Bx - B\overline{x})^t p_x(x).dx = B\Sigma_{xx}B^t$$

Dans le cas non linéaire, nous utilisons un DL d'ordre 1 de la fonction autour de la moyenne de la primitive de départ : si $\varphi(x) = \varphi(\overline{x}) + J_\varphi(\overline{x}).(x-\overline{x}) + O(\|x-\overline{x}\|^2)$, on a :

$$\overline{y} = E\left[\varphi(x)\right] = \varphi(\overline{x}) + O\left(E\left[\|x-\overline{x}\|^2\right]\right)$$
$$\Sigma_{yy} = E\left[(y-\overline{y})(y-\overline{y})^t\right] = E\left[(\varphi(x)-\varphi(\overline{x}))(\varphi(x)-\varphi(\overline{x}))^t\right] + O\left(E\left[\|x-\overline{x}\|^3\right]\right)$$
$$= J_\varphi(\overline{x}).E\left[(x-\overline{x})(x-\overline{x})^t\right].J_\varphi(\overline{x})^t + O\left(E\left[\|x-\overline{x}\|^3\right]\right)$$
$$\Sigma_{yy} = J_\varphi(\overline{x}).\Sigma_{xx}.J_\varphi(\overline{x})^t + O\left(E\left[\|x-\overline{x}\|^3\right]\right)$$

Lorsque les bruits ne sont pas trop important (i.e. lorsque $\|x-\overline{x}\|$ reste faible), on peut donc utiliser l'approximation suivante :

$$\overline{y} = \varphi(\overline{x})$$

$$\Sigma_{yy} = J_\varphi(\overline{x}).\Sigma_{xx}.J_\varphi(\overline{x})^t$$

Remarquons que l'on mesure ici la covariance de l'estimée y par rapport à l'estimée moyenne $\overline{y}$. Mais l'estimée moyenne ne correspond pas toujours à la véritable valeur de y, auquel cas on dit que φ est un estimateur biaisé de y. La covariance par rapport à $\overline{y}$ sous-estime alors la covariance par rapport à la véritable valeur de y. C'est pourquoi nous avons supposé que le recalage donnait le bon résultat en l'absence de bruit, i.e. que $\overline{y} = \varphi(\overline{x})$ donnait la bonne valeur de y et donc que φ n'est pas biaisé.

La transformation comme fonction des données On rassemblera les données dans le grand vecteur $\chi = (m_1^t,...,m_{n_M}^t,s_1^t,...,s_{n_S}^t)$, dont on supposera la covariance connue : $\Sigma_{\chi\chi} = (\Sigma_{m_1m_1},...,\Sigma_{m_{n_M}m_{n_M}},\Sigma_{s_1s_1},...,\Sigma_{s_{n_S}s_{n_S}})$. La transformation fournie par l'algorithme est celle qui minimise le critère, et donc obéit à $\Phi(T,\chi) = \frac{\partial C(T,\chi)}{\partial T} = 0$ (et $H = \frac{\partial \Phi(T,\chi)}{\partial T} = \frac{\partial^2 C(T,\chi)}{\partial T^2}$ définie positive). La fonction φ est donc une fonction implicite des données χ, définie par : $\Phi(\varphi(\chi),\chi) = 0$. Le théorème des fonctions implicites permet alors de calculer le Jacobien de φ, dont nous avons besoin, à partir des dérivées de Φ, que nous connaissons :

$$J_\varphi(\chi) = \frac{\partial\varphi}{\partial\chi} = -(\frac{\partial\Phi}{\partial T})^{(-1)}(\frac{\partial\Phi}{\partial\chi}) = -H^{(-1)}(\frac{\partial\Phi}{\partial\chi})$$

Résultat final En combinant les résultats des paragraphes précédents, on peut calculer la covariance sur la transformation donnée par l'algorithme de recalage :

$$\Sigma_{TT} = H^{(-1)}.(\frac{\partial\Phi}{\partial\chi}).\Sigma_{\chi\chi}.(\frac{\partial\Phi}{\partial\chi})^t.H^{(-1)} \tag{3.10}$$

avec :

$$\Phi(T,\chi) = \frac{\partial C(T,\chi)}{\partial T}$$
$$H = \frac{\partial\Phi(T,\chi)}{\partial T}$$

3.9.2 Application à l'ICP

Il ne nous reste plus ici qu'à calculer les différents éléments de la formule précédente, pour le critère de l'ICP avec distance de Mahalanobis. Le calcul de Φ et H a été effectué pour la descente de gradient (voir 3.8.2) :

$$\Phi = \frac{1}{2}\sum_{ij} A_{ij}\left(\frac{\partial s_{Ti}}{\partial T}\right)^t (\Phi_{s_{Ti}}D(s_{Ti},m_j)) = \sum_{ij} A_{ij}\left(\frac{\partial e_{ij}}{\partial T}\right)^t.\Sigma_{e_{ij}e_{ij}}^{-1}.e_{ij}$$

$$H = \frac{1}{2}\sum_{ij} A_{ij}.\left(\frac{\partial s_{Ti}}{\partial T}\right)^t (H_{s_{Ti}}D(s_{Ti},m_j))\left(\frac{\partial s_{Ti}}{\partial T}\right) = \sum_{ij} A_{ij}.\left(\frac{\partial e_{ij}}{\partial T}\right)^t.\Sigma_{e_{ij}e_{ij}}^{-1}.\left(\frac{\partial e_{ij}}{\partial T}\right) \tag{3.11}$$

Il reste donc à calculer $\frac{\partial\Phi}{\partial\chi}$: on néglige les dérivées secondes, et on a

$$\frac{\partial\Phi}{\partial\chi} = \frac{1}{2}\sum_{ij} A_{ij}.\left(\frac{\partial s_{Ti}}{\partial T}\right)^t\left(\frac{d.\Phi_{s_{Ti}}D(s_{Ti},m_j)}{d(s_{Ti},m_j)}\right)\left(\frac{\partial(s_{Ti},m_j)}{\partial\chi}\right)$$

et donc :

$$\begin{aligned}\Sigma_{TT} = H^{(-1)}.&\left(\tfrac{1}{2}\textstyle\sum_{i_1j_1} A_{i_1j_1}.\left(\frac{\partial s_{Ti_1}}{\partial T}\right)^t\left(\frac{d.\Phi_{s_{Ti}}D(s_{Ti_1},m_{j_1})}{d(s_{Ti_1},m_{j_1})}\right)\left(\frac{\partial(s_{Ti_1},m_{j_1})}{\partial\chi}\right)\right).\\ &\Sigma_{\chi\chi}.\left(\tfrac{1}{2}\textstyle\sum_{i_2j_2} A_{i_2j_2}.\left(\frac{\partial(s_{Ti_2},m_{j_2})}{\partial\chi}\right)^t\left(\frac{d.\Phi_{s_{Ti}}D(s_{Ti_2},m_{j_2})}{d(s_{Ti_2},m_{j_2})}\right)\left(\frac{\partial s_{Ti_2}}{\partial T}\right)\right).H^{(-1)}\end{aligned}$$

On récrit alors le terme $\left(\frac{\partial(s_{Ti_1},m_{j_1})}{\partial\chi}\right)\Sigma_{\chi\chi}\left(\frac{\partial(s_{Ti_2},m_{j_2})}{\partial\chi}\right)^t = R.\Sigma_{s_{i_1}s_{i_2}}.R^t + \Sigma_{m_{j_1}m_{j_2}}$, pour obtenir :

$$\Sigma_{TT} = H^{(-1)}.G.H^{(-1)}$$

avec

$$\begin{aligned}G = \tfrac{1}{4}\textstyle\sum_{i_1j_1i_2j_2} A_{i_1j_1}A_{i_2j_2}.&\left(\frac{\partial s_{Ti_1}}{\partial T}\right)^t\left(\frac{d.\Phi_{s_{Ti_1}}D(s_{Ti_1},m_{j_1})}{d(s_{Ti_1},m_{j_1})}\right).\\ &.(R.\Sigma_{s_{i_1}s_{i_2}}.R^t + \Sigma_{m_{j_1}m_{j_2}}).\left(\frac{d.\Phi_{s_{Ti_2}}D(s_{Ti_2},m_{j_2})}{d(s_{Ti_2},m_{j_2})}\right)\left(\frac{\partial s_{Ti_2}}{\partial T}\right)\end{aligned}$$

On peut alors simplifier ce calcul lorsque les erreurs sur les différents appariements de poids non nuls sont indépendantes, en remarquent que la covariance $\Sigma_{s_{i_1}s_{i_2}}$ (resp. $\Sigma_{m_{j_1}m_{j_2}}$) est dans ce cas nulle si $i_1 \neq i_2$ (resp $j_1 \neq j_2$). Le terme G devient alors :

$$\begin{aligned}G = \tfrac{1}{4}\textstyle\sum_{ij_1j_2} A_{ij_1}.A_{ij_2}.&\left(\frac{\partial s_{Ti}}{\partial T}\right)^t(H_{s_{Ti}}D(s_{Ti},m_{j_1})).R.\Sigma_{s_is_i}^{-1}.R^t(H_{s_{Ti}}D(s_{Ti},m_{j_2}))\left(\frac{\partial s_{Ti}}{\partial T}\right)+\\ \tfrac{1}{4}\textstyle\sum_{i_1i_2j} A_{i_1j}.A_{i_2j}.&\left(\frac{\partial s_{Ti}}{\partial T}\right)^t\left(H_{s_{Ti_1}}D(s_{Ti_1},m_j)\right).\Sigma_{m_jm_j}^{-1}.\left(H_{s_{Ti_2}}D(s_{Ti_2},m_j)\right)\left(\frac{\partial s_{Ti_2}}{\partial T}\right)\end{aligned}$$

que l'on peut factoriser :

$$\begin{aligned} G &= \sum_i G_i.R.\Sigma_{s_i s_i}^{-1}.R^t.G_i^t + \sum_j G_j.\Sigma_{m_j m_j}^{-1}.G_j^t \\ G_i &= \frac{1}{2}\sum_j A_{ij}.\left(\frac{\partial s_{Ti}}{\partial T}\right)^t (H_{s_{Ti}} D(s_{Ti}, m_j)) \\ G_j &= \frac{1}{2}\sum_i A_{ij}.\left(\frac{\partial s_{Ti}}{\partial T}\right)^t (H_{s_{Ti}} D(s_{Ti}, m_j)) \end{aligned}$$

Dans le cas de la distance de Mahalanobis, et lorsque il n'existe qu'un A_{ij} non nul (et égal à 1) pour i ou j donné (c'est le cas pour l'ICP, si on évite d'apparier plusieurs points de la scène au même point du modèle), le terme G devient :

$$G = \frac{1}{2}\sum_j A_{ij}.\left(\frac{de_{ij}}{dT}\right)^t.\Sigma_{e_{ij}e_{ij}}^{-1}.\Sigma_{e_{ij}e_{ij}}.\Sigma_{e_{ij}e_{ij}}^{-1}.\left(\frac{de_{ij}}{dT}\right) = \frac{1}{2}\sum_j A_{ij}.\left(\frac{de_{ij}}{dT}\right)^t.\Sigma_{e_{ij}e_{ij}}^{-1}.\left(\frac{de_{ij}}{dT}\right) = H \tag{3.12}$$

et finalement :

$$\Sigma_{TT} = H^{(-1)}.H.H^{(-1)} = H^{(-1)} \tag{3.13}$$

Cette matrice de covariance pourra être étudiée directement, ou pourra permettre d'exprimer la précision sur la position dans une zone cible (voir section 6.1.6).

3.9.3 Discussion

Les tests expérimentaux montrent que cette méthode donne des résultats beaucoup trop sous-estimés (la précision prédite dans la zone cible oscille entre $0.02\,mm$ et $0.1\,mm$ alors que les résultats expérimentaux montrent qu'elle est plutôt de l'ordre de $0.2\,mm$) pour le recalage de surfaces.

3.9.4 Méthodes de type "Bootstrap"

Il existe une autre grande catégorie de méthodes pour estimer l'incertitude d'un algorithme comme le nôtre, les méthodes de type "bootstrap" [Duda *et al.*, 2001, Section 9.4]. Nous présentons ici la méthode proposée par [Matei and Meer, 1999], qui est une méthode de type paramétrique, i.e. qui s'appuie comme notre prédiction sur une modélisation du bruit sur les données χ. Mais plutôt que de prédire la conséquence de ce bruit par un calcul basé sur une approximation au premier ordre de la fonction de recalage φ, elle la mesure par simulation, en observant la variabilité des recalages obtenus pour une centaine de jeux de données χ' obtenues en bruitant χ suivant le modèle de bruit retenu.

L'avantage de ces méthodes est qu'elles donnent de résultats un peu plus pertinents, car elles utilisent directement l'algorithme de recalage, et évitent donc les approximations que nous avons dû faire pour la prédiction (approximation linéaire de la dérivé du critère, non prise en compte des variations des appariements, non prise en compte des dérivés secondes dans le calcul du Hessien, ...).

Les défauts de ces méthodes sont qu'elles ne donnent pas de formule explicite, et qu'elles sont beaucoup plus lentes car elles exigent de faire de nombreux recalages.

Par conséquent, nous avons surtout utilisé les méthodes "bootstrap" pour valider la prédiction théorique (notamment dans la section 5.4). La figure 3.6 montre un exemple où la prédiction théorique fournit des résultats très proche des méthodes "bootstrap", ce qui permet de justifier les approximations réalisées dans les calculs et choisir finalement la prédiction théorique pour l'usage quotidien.

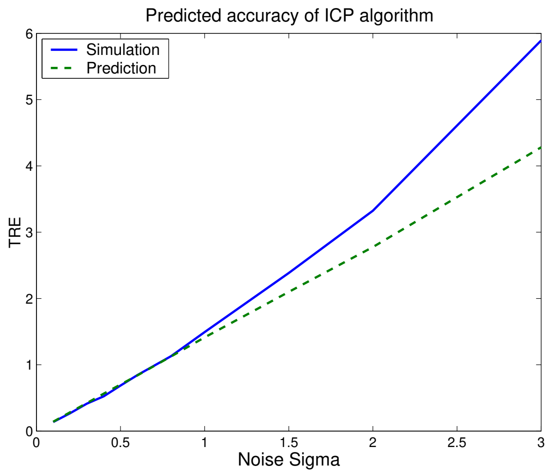


FIG. 3.6 – *Comparaison de la précision prédite théoriquement et simulée par "bootstrap" de l'ICP pour un recalage de 3 points identifiés avec différents niveaux de bruits. On constate que les chiffres sont très proches pour des variances inférieures à 2 mm.*

3.10 Conclusions

Cet état de l'art suggère de nombreuses directions pour améliorer l'ICP. Un premier axe porte sur la résolution de ses problèmes de robustesse. Deux idées se dégagent :

- l'EM associé au recuit simulé semble être l'approche la plus efficace pour la robustesse par rapport à la transformation initiale. Il faut cependant lui donner une justification théorique, et savoir comment régler ses paramètres. C'est ce que nous ferons au prochain chapitre.
- l'utilisation de primitives plus complexes permet d'améliorer la robustesse aux points aberrants : nous pourrions en l'occurrence utiliser des points orientés, puisque nous disposons d'une information sur la normale à la surface en chaque point de nos nuages. Ce sera le sujet du chapitre 5, puis, dans un cadre plus général, des chapitres 7 et 8.

Un second axe porte sur l'adaptation de l'ICP au recalage de surfaces (où les points ne sont pas véritablement appariés) et en particulier sur l'estimation théorique de l'incertitude car elle fonctionne mal sous sa forme actuelle. De plus, nous ne savons absolument pas comment prédire les autres sources de bruit (interne par exemple). Nous aborderons ce dernier point dans la section 5.4.

Chapitre 4

ICP et EM multi-échelle

Sommaire

4.1 Introduction

4.1.1 Problématique

L'état de l'art présenté au chapitre précédent montre une chose en particulier : la quasi totalité des développements qui ont été faits autour de l'ICP sont très pragmatiques, heuristiques, reposant la plupart du temps sur la minimisation d'un ou plusieurs critères bien choisis, mais pas véritablement justifiés. Nous souhaiterions donc mettre en place une façon de dériver ces critères qui s'appuie sur des hypothèses claires et un raisonnement rigoureux permettant de mieux comprendre pourquoi les algorithmes fonctionnent, et comment les adapter en fonction des données.

Plusieurs travaux suggèrent ou présentent une modélisation probabiliste du recalage :

- Les travaux de Pennec [Pennec, 1996] justifient l'utilisation de la distance de Mahalanobis dans l'ICP pour des primitives variées (points, trièdres, ...).
- Les travaux de Rangarajan et al. (voir section 3.7) présentent des variantes de l'ICP basées sur des critères inspirés des statistiques.
- Wells [Wells, 1997] propose une modélisation statistique de l'ICP et suggère d'appliquer l'EM à l'ICP. Il ne l'implémente et ne le teste malheureusement pas.
- Les travaux de [Luo and Hancock, 1999] appliquent l'EM au recalage rigide, mais de façon peu claire et en se focalisant sur le problème de l'appariement des points.

Nous nous proposons donc ici de regrouper et compléter ces travaux, pour justifier l'ICP/EM, puis de leur donner une nouvelle interprétation multi-échelle.

4.1.2 Organisation du chapitre

La première étape pour dériver l'ICP/EM est de ré-écrire notre problème de recalage sous forme d'un problème de maximisation de vraisemblance. Nous le ferons à la section 4.2, où nous verrons en particulier que l'ICP est en-fait l'approche "Maximum de vraisemblance pur". Nous verrons ensuite comment l'adapter à la façon de l'EM (section 4.3).

D'un point de vue pratique (section 4.4), le principal intérêt de l'ICP/EM sera de lisser plus ou moins le critère de l'ICP, en fonction d'un paramètre d'échelle, la variance. Nous verrons qu'à petite échelle, le critère sera le même que l'ICP, avec ses nombreux minima locaux et les problèmes de robustesse par rapport à la transformation initiale que cela entraîne. Ces minima disparaîtront au fur et à mesure que nous augmenterons l'échelle, jusqu'à se fondre en un seul minimum global, qui correspondra à l'alignement des centres et moments d'inerties des deux nuages de points. Le recalage sera alors parfaitement robuste, mais très imprécis. Nous obtiendrons finalement un recalage à la fois robuste et précis en appliquant une méthode de raffinement progressif de l'échelle ("coarse-to-fine" en anglais), i.e. en partant d'une grande échelle et en la diminuant au cours des itérations. Puis nous poursuivrons l'analogie avec les approches multi-résolutions connues en sous-échantillonnant nos données en fonction de l'échelle, pour réduire le temps de calcul. Ce sous-échantillonnage sera effectué avec un algorithme de décimation très simple qui permettra en outre d'homogénéiser l'échantillonnage des nuages de points, et donnera finalement des résultats bien plus robustes et rapides que l'ICP.

Dans la section 4.5, nous calculerons les différentes dérivés du critère de l'ICP/EM, qui permettront d'adapter les méthodes d'optimisation par descente de gradient et les méthodes de prédiction théorique de l'incertitude.

Dans le section 4.6, nous pour parlerons de la parallélisation des algorithmes (ICP et ICP/EM), qui permettra, en utilisant plusieurs ordinateurs, d'accélérer encore le recalage.

Précisons que la partie principale de ce chapitre (section 4.2 à 4.4) est une version légèrement remaniée de l'article de conférence publié à ECCV'02 [Granger and Pennec, 2002b].

4.2 Statistical formulation of the ICP algorithm

Let s_i be the $n_{\mathcal{S}}$ points of the scene $\mathcal{S}$, m_j the $n_{\mathcal{M}}$ points of the model $\mathcal{M}$ and T a rigid transformation from the scene to the model. Assuming that $T \star s_i$ *is homologous to* m_j (i.e. a measure of m_j) with a known noise model, its density probability function can be defined by:

$$p(s_i|m_j,T) = p(T \star s_i|m_j) \tag{4.1}$$

In the case of an additive and centered Gaussian noise of covariance Σ, this probability can be defined using the Mahalanobis distance $\mu^2(x,y) = (x-y)^t.\Sigma^{(-1)}.(x-y)$:

$$p(s_i|m_j,T) = k^{-1}.\exp(-\mu^2(T \star s_i,m_j)/2)$$

To represent the matches estimation, we use a binary matrix A: $A_{ij} = 1$ if s_i matches m_j and 0 otherwise. Since each scene point s_i is assumed to correspond exactly to one model point with index say $j^\star$, we have $A_{ij} = \delta_{jj^\star}$ and $\sum_j A_{ij} = 1$ for all scene indices i. In order to represent random matches, we use a random matching matrix $\mathbf{A}$. Each possible matching matrix A has a probability $p(A) = P(\mathbf{A} = A)$ and verifies the previous constraints: $\overline{A_{ij}} = E[\mathbf{A}_{ij}] = P[\mathbf{A}_{ij} = 1] \in [0,1]$ and $\sum_j \overline{A_{ij}} = 1$. Finally, since scene points will assumed to be independent (see below), and using $\alpha^1 = \alpha$ and $\alpha^0 = 1$, we can write:

$$p(A) = \prod_{ij/A_{ij}=1} p(\mathbf{A}_{ij} = 1) = \prod_{ij} (\overline{A_{ij}})^{A_{ij}}$$

The first example of such a random matrix is the a-priori probability of the matches:

this is the law giving the probability that a given scene point is a measure of a given model point knowing nothing else: $p(\mathbf{A}_{ij} = 1) = \overline{\pi_{ij}}$. A relevant choice is usually the uniform law :

$$\overline{\pi_{ij}} = \frac{1}{n_{\mathcal{M}}}$$

Now, the joint probability of s_i and $\mathbf{A}_{ij}$ is simply: $p(s_i,A_{ij} = 1|\mathcal{M},T) = \overline{\pi_{ij}}.p(s_i|m_j,T)$. We can write this last equality for a whole row of A:

$$p(s_i,A_i|\mathcal{M},T) = \prod_j (\overline{\pi_{ij}}.p(s_i|m_j,T))^{A_{ij}}$$

Assuming that *all scene points are conditionally independent*, the joint probability of the scene and the matches is the product of each of them:

$$p(\mathcal{S},A|\mathcal{M},T) = \prod_i p(s_i,A_i|\mathcal{M},T) = \prod_{ij} (\overline{\pi_{ij}}.p(s_i|m_j,T))^{A_{ij}} \tag{4.2}$$

Finally, we are looking for the transformation that maximises the likelihood of the observed scene, or equivalently minimises its negative log:

$$C_{ICP}(T,A) = -\log p(\mathcal{S},A|\mathcal{M},T) = \sum_{ij} A_{ij}.(-\log p(s_i|m_j,T) - \log \overline{\pi_{ij}}) \tag{4.3}$$

For an homogeneous Gaussian noise and uniform priors, this simplifies into:

$$C_{ICP}(T,A) = \tfrac{1}{2} \sum_{ij} A_{ij}.\mu^2(T \star s_i,m_j) + Cte \tag{4.4}$$

One recognises here the standard ICP criterion using the Mahalanobis distance (see eq. 3.4 and 3.7). This proves that ICP is no more than a maximum likelihood approach of the registration problem. Moreover, [Kanatani, 1996] showed that this is the best (minimal variance) estimator. Note that the criterion is invariant w.r.t a global scaling of the noise covariance. This property will not hold any more for the following EM formulation.

4.3 Derivation of the ICP/EM algorithm

4.3.1 ML estimation of the transformation ignoring the matches

In the previous section, the transformation and the matches were both estimated by the maximisation of the likelihood of the scene. But for registration purposes, the matches are not parameters of interest. Two different ideas arise: on the one hand, we could maximise the scene likelihood knowing only the transformation. We present this method in this section. On the other hand, there is no reason to consider only the most likely matches, especially when there are ambiguities. We could consider all the possible matches, compute their respective probability, and use the expectation of the previous criterion (eq. 4.3) w.r.t. these probabilities (a kind of Bayesian estimation). This is presented in section 4.3.2.

Let us go back to the probability of each scene point. As we don't want to deal with the matches estimation, we have to consider the likelihood of a scene point knowing only the transformation:

$$p(s_i|\mathcal{M},T) = \textstyle\sum_{\{A_i\}} p(s_i,A_i|m_j,T) = \sum_j \overline{\pi_{ij}}.p(s_i|m_j,T)$$

This amounts to seeing the scene points as measurements of a mixture of probabilities around the model points: there is no more homology between a scene point and one of the model points. This interpretation is specially adapted for our case since the model is a surface and not a collection of landmarks. Assuming once again the independence of the scene points measurements, the likelihood of the scene and its negative log are:

$$p(\mathcal{S}|\mathcal{M},T) = \textstyle\prod_i \sum_j \overline{\pi_{ij}}.p(s_i|m_j,T) \tag{4.5}$$

$$C_{EM}(T) = \textstyle\sum_i (-\log \sum_j \overline{\pi_{ij}}.p(s_i|m_j,T)) \tag{4.6}$$

Unfortunately, this criterion has no closed form solution and we do not know any robust and efficient method to minimise it directly.

4.3.2 "Bayesian" estimation of the transformation

We now turn to the second method, which implies the computation of the a-posteriori probability of each possible matches matrix A knowing the scene, the model and the transformation. This can be achieved using Bayes rule, and eq. 4.5 and 4.2. To simplify the notations, we denote by $\mathbf{A}_T$ the random matching matrix defined by:

$$p(\mathbf{A}_T = A) = p(A|\mathcal{S},\mathcal{M},T) = \tfrac{p(\mathcal{S},A|\mathcal{M},T)}{p(\mathcal{S}|\mathcal{M},T)} = \textstyle\prod_{ij} \left(\frac{\overline{\pi_{ij}}.p(s_i|m_j,T)}{\sum_k \overline{\pi_{ik}}.p(s_i|m_k,T)} \right)^{A_{ij}} \tag{4.7}$$

Since we have $p(A_T = A) = \prod_{ij} (\overline{(A_T)_{ij}})^{A_{ij}}$, we obtain by identification:

$$\overline{(A_T)_{ij}} = \frac{\overline{\pi_{ij}}.p(s_i|m_j,T)}{\sum_k \overline{\pi_{ik}}.p(s_i|m_k,T)} \tag{4.8}$$

Thus, the idea is to optimise the expectation of our first criterion (eq. 4.3) w.r.t. this law:

$$C_{\mathbf{A}_T}(T) = \mathbf{E}_{\mathbf{A}_T}\left[C_{ICP}(T,A)\right] = \sum_{ij} \overline{(A_T)_{ij}}.(-\log p(s_i|m_j,T) - \log \overline{\pi_{ij}}) \tag{4.9}$$

At this point, the algorithm is obvious: starting from a first estimation of the transformation, we alternatively compute the probability of matches $\overline{(A_T)_{ij}}$ and optimise this new criterion. Unfortunately, this new algorithm does not correspond to the minimisation of a well posed criterion, as $\overline{(A_T)_{ij}}$ depends on T and vice versa. Thus, the convergence is not ensured.

4.3.3 Equivalence of the two approaches: the EM algorithm

In fact, it turns out that the algorithm justified in section 4.3.2 is no more than the optimisation of the criterion presented in section 4.3.1, through an Expectation-Maximisation (EM) approach [Couvreur, 1996; Neal and Hinton, 1998]. Thus, the algorithm is ensured to converge.

To relate the criterion $C_{EM}(T) = -log(p(\mathcal{S}|\mathcal{M},T))$ with the optimisation method, let us introduce the matching matrix in the criterion using Bayes rule and take its expectation for any law on A: $C_{EM}(T) = E_{\mathbf{A}}\left[C_{EM}(T)\right] = -E_{\mathbf{A}}\left[\log p(\mathcal{S},A|\mathcal{M},T)\right] + E_{\mathbf{A}}\left[\log p(A|\mathcal{S},\mathcal{M},T)\right]$. Since this criterion is still independent of $\mathbf{A}$, we create a secondary criterion that depends explicitly on them by adding the Kullback-Leibler distance between $\mathbf{A}$ and the a-posteriori law on the matches: $KL(\mathbf{A}||\mathbf{A}_T) = E_{\mathbf{A}}\left[\log p(A) - \log p(A|\mathcal{S},\mathcal{M},T)\right]$. This distance is positive, and null only for $\mathbf{A} = \mathbf{A}_T$. The criterion rewrites :

$$\begin{aligned} C_{EM}(T,\mathbf{A}) &= C_{EM}(T) + KL(\mathbf{A}||\mathbf{A}_T) && (4.10)\\ &= E_{\mathbf{A}}\left[\log p(\mathcal{S},A|\mathcal{M},T)\right] + E_{\mathbf{A}}\left[\log p(A)\right] && (4.11)\end{aligned}$$

The EM algorithm is the alternative minimisation of this criterion.

E-Step: estimation of the matches Here, T is fixed and we minimise the criterion w.r.t. $\mathbf{A}$. Using eq. (4.10), we are left with $\mathbf{A} = \mathbf{A}_T$. Since the distribution of the $(\mathbf{A}_T)_{ij}$ is binary and normalized, it can be uniquely characterized by its mean value $\overline{(A_T)_{ij}}$. As scene points are independant, they can be treated seperatly. In practice, for each scene point, we search all close model points, and compute their weights using eq. (4.8).

M-Step: estimation of the transformation Now, the matches probability are fixed and equal to $\mathbf{A}_T$, and we minimise the criterion w.r.t. T. Using eq. (4.11) and dropping the constant term $E_{\mathbf{A}}\left[\log p(A)\right]$, we are left with the minimisation of the expectation of $\log(p(\mathcal{S},A|\mathcal{M},T)))$, which is exactly the criterion (4.9).

Optimality and convergence: After the E-Step, we have exactly $KL(\mathbf{A}||\mathbf{A}_T) = 0$ and then $C_{EM}(T,\mathbf{A}) = C_{EM}(T)$. This proves that our "Bayesian" estimation is a computational trick for minimising $C_{EM}(T)$ and thus the algorithm is ensured to converge toward a local minimum of $C_{EM}(T)$.

4.4 Multi-Scale gaussian ICP/EM with decimation

We focus in this section on the simple case where the noise is homogeneous, Gaussian and isotropic with uniform priors on the matches. In this case, we are left with only one free parameter: the variance of the Gaussian noise. We first analyze the influence of the variance on the criterion shape and show that it behaves as a smoothing factor. This leads to a multiscale scheme designed to improve robustness without penalizing the accuracy: due to the high computational cost at coarse scales, we combine an annealing scheme with a simple but very efficient decimation technique as an approximation of the criterion. This technique is very similar to the multi-scale approaches used in standard image (i.e. intensity) analysis domains, and lead to substantial improvements in terms of robustness and computation times.

4.4.1 ICP/EM for a homogeneous Gaussian noise and uniform priors

For an isotropic Gaussian noise, the Mahalanobis distance reduces to a rescaled distance: $\mu^2(T\star s_i,m_j) = \frac{1}{\sigma^2}\|T\star s_i,m_j\|^2$. For an homogeneous noise (all points have the same variance) and uniform priors, the $\overline{\pi_{ij}}$ and the normalisation constants of the gaussian laws are all equal, and the two EM steps turn out to be particularly simple[11]:

$$\overline{(A_T)_{ij}} = \frac{\exp(-\|T\star s_i - m_j\|^2/2\sigma^2)}{\sum_k \exp(-\|T\star s_i - m_k\|^2/2\sigma^2)} \tag{4.12}$$

$$C_{\mathbf{A}_T}(T) = n_{\mathcal{S}}.\mathfrak{D}.\log(\sigma) + \frac{1}{2\sigma^2}\sum_{ij}\overline{(A_T)_{ij}}.\|T\star s_i - m_j\|^2 \tag{4.13}$$

Practically, the only difference with the ICP for optimizing the transformation is the presence of non-binary weights. In the rigid case, this can be solved by a straightforward adaptation of the SVD or the unit quaternion methods [Eggert *et al.*, 1997].

One can even simplify eq. 4.13 by considering that each scene point s_i contribute to the criterion through $\sum_j \overline{(A_T)_{ij}}\|T\star s_i - m_j\|^2$. As the weights are normalized, this is equal to $\|T\star s_i - \overline{m_i}\|^2 + \sum_j \overline{(A_T)_{ij}}\|\overline{m_i} - m_j\|^2$, where $\overline{m_i}$ is the barycenter of the m_j weighted by the $\overline{(A_T)_{ij}}$. As the second term is constant during the M-Step, we are left with the

11. On the contrary, when noise is not homogeneous, one has to take the priors and normalisation constants into account when computing the weights. Thus, the most probable point is not always the closest one.

optimization of $\sum_i \|T \star s_i, \overline{m_i}\|^2$, which is a standard non-weighted least-square.

One important practical problem is the rejection of outliers. The theoretical developments of the previous section could easily be adapted to add a probability to match a scene point to the background, as done in [Wells, 1997, pp.78]. In our case, this turns out to be almost equivalent to thresholding the Mahalanobis distance. Thus, we only need to look for matches up to a maximum distance (e.g. $3.\sigma$). Model points farther away are considered as outliers and are given a null weight.

To determine the variance, we may use a ML approach, which amounts to minimizing the EM criterion with respect to σ given a transformation and a set of (probabilistic) matches. This gives a simple RMS estimation:

$$\sigma^2 = \frac{1}{n_{\mathcal{S}}.\mathfrak{D}} \sum_{ij} \overline{(A_T)_{ij}}.\|T \star s_i, m_j\|^2. \tag{4.14}$$

One could think to estimate the variance this way at each EM iteration as a third step of the optimization process, but we experimentally observed that the decrease was too sharp and was sticking the algorithm in local minima.

4.4.2 Criterion shape and annealing scheme

To understand the influence of the variance, we analyzed in appendix B the asymptotic values of the criterion and showed that, for high values of the variance, it simply aligns inertia centers and moments of the two point sets, while it reaches the standard ICP criterion for small values of σ. Intermediate values of the variance are presented in Fig. 4.1: one clearly see that the brittle shape of the ICP criterion (with numerous local minima) is smoothed into an almost quadratic shape for a higher variance with a decreasing number of local minima for intermediate values. One can also see that the correct (i.e. the global) minimum is unfortunately shifted when the variance is increased. This suggests that we should start from a large variance, which should guaranty the robustness, and track this minimum as the variance decrease to end up with the real noise variance that will ensure the most accurate results.

Since using the ML estimation of the variance imposes too fast a decay, we chose to impose a slight but regular decrease by dividing the variance parameter after each iteration with a fixed value called annealing coefficient (typically 1.1). Notice that doing so can sometimes lead to an increase of the criterion value between two step. An alternative approach would be to wait for the convergence before decreasing the variance parameter, but we experimented that it was slower and less efficient. The annealing process is stopped either when the variance reaches a predefined value, computed on a good registration of typical data-sets, or when it reaches the value estimated using eq. 4.14 at the current iteration. The initial value of the variance obviously depends on the quality of the ini-

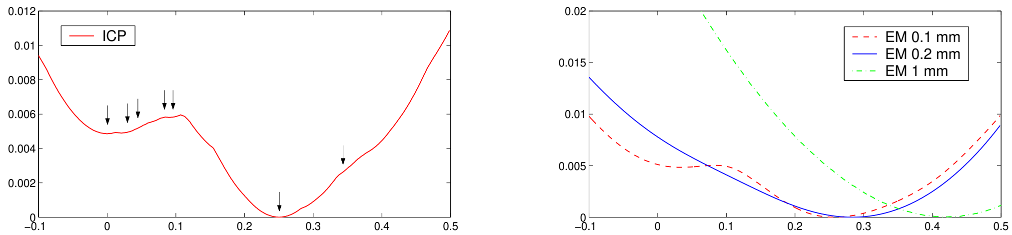


FIG. 4.1 – *Criterion vs Z-translation for ICP (on the left) and for EM (on the right). There are two relevant minima for ICP (0, where the algorithm stopped, and 0.25, where criterion is minimum) and irrelevant ones (e.g. 0.1 and 0.3). The EM criterion has a much smoother shape. For an under-estimated variance (0.1 mm) the irrelevant minima have disappeared. For the approximative noise variance (0.2 mm), even the first minima has disappeared. For a larger (overestimated) variance (1 mm), the criterion is almost quadratic, but the global minimum has been shifted.*

tialization (i.e. the initial estimation of the transformation). We have no theoretical way of evaluating this value yet. In our experiments, we used a value varying from 16 to 100 times the typical real noise variance (i.e. between 4 and 10 times the standard-deviation).

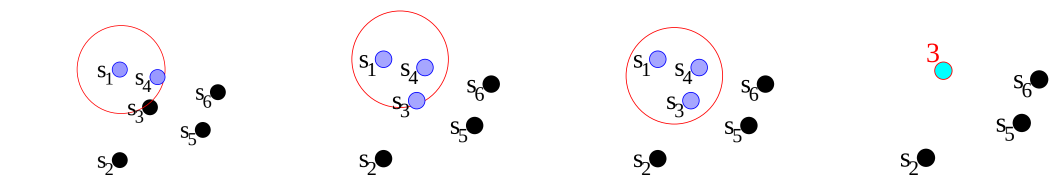
4.4.3 Decimation

In this paper, we are interested in large number of scene points (typically 20000 to 50000). With the ICP algorithm, most of the computation time is spent in the closest point search. This is usually implemented using a kD-tree or some similar method and the search is limited to the threshold distance defined for outliers rejection. The same technique is of course applied when using the ICP/EM algorithm, except that all points closer than a maximal Mahalanobis distance are taken into account.

With the annealing scheme, the problem is that the maximum Euclidean search distance is then proportional to the standard deviation: with a large variance, virtually all model points have to be retrieved and the kD-tree becomes completely inefficient. This has an "exponential" influence on the computation time that practically forbids the use of the annealing scheme. The main idea was then to see the variance as a down-sampling parameter: at a coarse scale, we do not need to be accurate and we can approximate roughly our data-set. When the variance goes down, we have to refine our approximation to increase the accuracy.

To speed-up the algorithm at a given scale, the basic idea is to observe that if n scene points are close enough (w.r.t. the variance), they will share almost the same closest points within the model with almost the same weights. Thus, approximating these n points by considering n time one of them will optimize a good approximation of our criterion (n is called the *decimation weight*). In fact, we strongly believe that we have to use their barycenter to obtain a second order approximation of the criterion (but we lack a detailed proof). This leads to a simple decimation technique that merges scene points that are close to each other. More precisely, we consider that the distance of a point to its approximation has to be inferior to $\alpha.\sigma$.

In fact, any decimation technique that replace a subset of close points by their barycenter will be compatible with ICP/EM. One could imagine lots of different ways of finding these subsets of points. We present in Fig. 4.2 a simple and very fast technique: the sphere decimation. It consists in adjusting a sphere of radius $\alpha.\sigma$ centered on the barycenter of the points it contains. The modifications of the ICP/EM algorithm including the decimation and annealing schemes are presented in Table 4.1.



(a) Sphere init. (b) Iteration 1 (c) Iteration 2 (d) Fusion

FIG. 4.2 – ***The greedy sphere decimation algorithm.*** *At the beginning, we define a* remaining list *containing all the points of the original cloud and we create an empty* decimated cloud. *We iteratively add points in the decimated cloud as follows. Starting from a remaining point, we initialize a sphere of radius $\alpha.\sigma$ centered on this point (a). Then, we iteratively find all remaining points in this sphere, compute their barycenter $\overline{s}$, and move the center of the sphere to this barycenter (b,c). After convergence, we create a new point in the "decimated cloud" at the center of the sphere with a weight equal to the number of remaining points within the sphere, and we remove these remaining points (d). The whole process ends up when there are no more remaining points.*

Initialization : Compute a first estimation of T, and set σ^2 to its initial value.
Repeat

> **Decimation** : Decimate the scene with a sphere radius $\alpha.\sigma$.
> **E-Step** : For each s_i in the decimated scene with decimation weight w_{s_i}:
> Search all m_j such that $\|T \star s_i - m_j\|^2 < \sigma^2.\mu^2_{max}$ using a kD-tree
> Compute the weights $\overline{(A_T)_{ij}}$ using Eq. 4.12
> **M-Step** : Re-estimate T by minimising $\sum_{ij} w_{s_i} \overline{(A_T)_{ij}}.\|T \star s_i - m_j\|^2$.
> **Annealing** : Divide σ^2 by the annealing coefficient. If σ^2 is below the final noise variance, set it to the final noise variance.

Until convergence

TAB. 4.1 – *Pseudo-code of the Multiscale ICP/EM registration algorithm.*

4.4.4 Analogy with multi-scale intensity-based techniques

As the variance can be considered as a scale parameter, it is interesting to determine how our annealing and decimation scheme relates to multiscale techniques already developed for intensity-based registration. We analyze in this section the coarse-to-fine aspects, realized by the coupling of the EM algorithm and the annealing scheme, and the down-sampling aspects, realized by the decimation, and linked to the multi-grid approach with intensity images. Our discussion will be quite pragmatic, though we think it could be integrated into a rigorous multi-scale framework (see for instance [Lindeberg, 1994]).

Coarse-to-fine approaches

The main idea underlying multiscale intensity- (i.e. image-) based registration approaches is to get rid of small details that create local minima and trap the algorithm when the initialization is too far away from the solution. This is especially the case for non-rigid registration [Thévenaz *et al.*, 1998]. This is usually implemented using a pyramid of blurred images with an increasing Gaussian kernel [Jolion, 1994].

Using a Gaussian mixture model for representing the probability of measuring points follows essentially the same idea: we are representing the pdf of measuring one point of the model as a blurred version of the original (Dirac) model measurements. As the variance grows to larger values, we will loose the precise local shape of the surface to focus on a more global shape.

The main difference with images is that our measurements are not on a regular grid: precomputing the pdf on a regular grid would forbid the use of an efficient ICP-like algorithm. This is why the blurring is introduces directly within the algorithm via the EM weights. Moreover, the normalization has to be done for each scene point independently (as distances are different) and cannot be incorporated into the blurring kernel. This could be a difficult point in the rigorous modeling of our algorithm as a multiscale approach.

In the multiscale theory, one could show that the causality hypothesis implies that the local minima at a fine scale are gradually merging with the scale, and that there is no bifurcation (creation of new minima). Although we experimented a similar behavior with ICP/EM (see Fig. 4.1), we lack a rigorous multiscale setting to prove it. Such a result would be especially interesting since we know that there is a unique minimum at the largest scale (if the inertia moments of the point clouds are sufficiently anisotropic): could we find a path along scales leading from this unique minimum to the global minimum at the finest scale? If there exists moreover a technique to choose the right path, we could obtain a perfectly robust algorithm.

Decimation

Multi-scale techniques are often coupled with a down-sampling of the image at each-scale, designed to improve the computational efficiency. This down-sampling is usually integrated into a multi-grid approach where the resolution of images is for instance divided by two at each coarser scale. This is implemented through a resampling algorithm that takes into account the Gaussian blurring, and is designed to minimize the impact on the registration criterion while reducing drastically the volume of data.

Our decimation technique was designed with the same concern, but there are some major differences that could have a theoretical impact. Firstly, the scale does not need to be divided by two at each scale to obtain an efficient algorithm: we can have a smoother scale evolution that gives better results (see section 4.4.2). The counterpart is that we cannot construct a pyramid using a recursive technique and we have to down-sample the set of points at each scale from the original set of points. In our case, this is not a problem as this step is computationally cheap. Secondly, our decimation is applied before the implicit Gaussian blurring, and not concurrently as for images. This means that we compute an ad-hoc approximation of the point location pdf, and not the best possible one. However, the possible gain of such an optimisation would probably be discarded by the extra computation time.

4.5 Compléments sur l'ICP/EM

Nous allons ici faire quelques compléments sur l'EM. Nous allons en particulier calculer les différentes dérivées de son critère, qui peuvent être utile pour implémenter une descente de gradient (voir 3.8.2) ou adapter la prédiction de l'incertitude (voir 3.9).

4.5.1 Gradient du critère & descente de gradient

Le critère de l'EM est donc :

$$C_{EM}(T) = \textstyle\sum_i (-\log \sum_j \overline{\pi_{ij}}.p(s_i|m_j,T)) = \sum_i (-\log \sum_j \overline{\pi_{ij}}.\exp(-D(s_{Ti},m_j)/2)) \quad (4.15)$$

nous allons dériver le critère sur cette dernière forme, pour simplifier les calculs. Pour le gradient du critère, nous avons :

$$\frac{dC_{EM}(T)}{dT} = \frac{1}{2}\sum_i \frac{\sum_j \overline{\pi_{ij}}.\exp(-D(s_{Ti},m_j)/2).\frac{dD(s_{Ti},m_j)}{dT}}{\sum_k \overline{\pi_{ik}}.\exp(-D(s_{Ti},m_j)/2)} = \frac{1}{2}\sum_{ij} A_{Tij}\frac{dD(s_{Ti},m_j)}{dT}$$

$$\frac{dC_{EM}(T)}{dT} = \frac{1}{2}\sum_{ij} A_{Tij}.\left(\frac{dD(s_{Ti},m_j)}{ds_{Ti}}\right).\left(\frac{ds_{Ti}}{dT}\right)$$

on retrouve ainsi une dérivé analogue à l'ICP, et c'est la raison profonde pour laquelle les méthodes d'optimisations directes qui ne reposent que sur des dérivés simples du critère (Barycentre+SVD) ont pu être directement adaptées. Ceci ne sera pas vrai pour les dérivés secondes, car le terme de pondération n'est pas constant par morceau comme dans l'ICP, et va donc intervenir dans les dérivations suivantes.

En ce qui concerne la descente de gradient de type Newton (voir 3.8.2), celle-ci fait aussi appel au Hessien du critère. Elle l'utilise comme un "facteur de vitesse" permettant de pondérer le gradient qui indique la direction dans laquelle aller pour minimiser le critère. Si on utilise directement le hessien de l'ICP, on utilisera une vitesse inexacte, et la conséquence sera d'augmenter le nombre d'itération de l'algorithme, sans pour autant la remettre en question.

Cas gaussien On peut écrire le gradient du critère de l'EM dans le cas gaussien (voir 3.8.2 pour les dérivés des distances) :

$$\frac{dC_{EM}(T)}{dT} = \sum_{ij} A_{Tij}.e_{ij}.\Sigma^{-1}_{e_{ij}e_{ij}}.(\frac{\partial e_{ij}}{\partial T})$$

4.5.2 Hessien du critère & vraie descente de gradient de type Newton

Nous allons ici calculer le Hessien du critère de façon exacte. Il va nous falloir pour cela la dérivée des poids d'appariements :

$$A_{Tij} = \frac{\overline{\pi_{ij}}.\exp(-D(s_{Ti},m_j)/2)}{\sum_k \overline{\pi_{ik}}.\exp(-D(s_{Ti},m_j)/2)}$$

$$\frac{dA_{Tij}}{dT} = -\frac{1}{2}\frac{\left\{\begin{array}{c} \left(\frac{dD(s_{Ti},m_j)}{dT}.\overline{\pi_{ij}}.\exp-\frac{D(s_{Ti},m_j)}{2}\right).\left(\sum_k \overline{\pi_{ik}}.\exp-\frac{D(s_{Ti},m_k)}{2}\right) \\ -\left(\overline{\pi_{ij}}.\exp-\frac{D(s_{Ti},m_j)}{2}\right).\left(\sum_k \frac{dD(s_{Ti},m_k)}{dT}.\overline{\pi_{ik}}.\exp-\frac{D(s_{Ti},m_k)}{2}\right)\end{array}\right\}}{\left(\sum_k \overline{\pi_{ik}}.\exp-\frac{D(s_{Ti},m_k)}{2}\right)^2}$$

$$\frac{dA_{Tij}}{dT} = -\frac{1}{2}A_{Tij}\left(\frac{dD(s_{Ti},m_j)}{dT} - \sum_k A_{Tik}.\frac{dD(s_{Ti},m_k)}{dT}\right)$$

Pour le hessien du critère, nous obtenons après simplification :

$$H = \frac{1}{2}\left\{\begin{array}{c} \sum_i \left(\sum_j A_{Tij}H_T D(s_{Ti},m_j)\right) \\ -\sum_j A_{Tij}\frac{dD(s_{Ti},m_j)}{dT}^t.\frac{dD(s_{Ti},m_j)}{dT} \\ +\left(\sum_j A_{ij}.\frac{dD(s_{Ti},m_j)}{dT}\right)^t\left(\sum_k A_{ik}.\frac{dD(s_{Ti},m_j)}{dT}\right)\end{array}\right\}$$

Discussion Nous avons vérifié expérimentalement sur la base de test (voir 6.1.5) les différences de comportement entre une descente de gradient utilisant le hessien de l'ICP et une utilisant ce hessien corrigé. Nous n'avons pas noté de différences notables entre les transformations obtenues et le nombre d'itérations a diminué d'environ 30% en utilisant la nouvelle version du Hessien.

4.5.3 Calcul d'incertitude

Pour adapter le calcul d'incertitude (voir section 3.9), nous devons aussi calculer la dérivé de Φ par rapport aux données, pour calculer le terme G. Le calcul est analogue au précédent :

$$\frac{dA_{Tij}}{d\chi} = -\frac{1}{2}A_{Tij}\left(\frac{dD(s_{Ti},m_j)}{d\chi} - \sum_k A_{ik}.\frac{dD(s_{Ti},m_j)}{d\chi}\right)$$

$$\frac{d\Phi}{d\chi} = \frac{1}{2}\sum_i \left\{ \begin{array}{c} \sum_j A_{Tij} \frac{d\frac{dD(s_{Ti},m_j)}{dT}^t}{d\chi} \\ -\sum_j A_{Tij} \frac{dD(s_{Ti},m_j)}{dT}^t . \frac{dD(s_{Ti},m_j)}{d\chi} \\ +\left(\sum_j A_{ij}.\frac{dD(s_{Ti},m_j)}{dT}\right)^t \left(\sum_k A_{ik}.\frac{dD(s_{Ti},m_k)}{d\chi}\right) \end{array} \right\}$$

Dans le cas où les points sont indépendants, et $\Sigma_m = 0$ on obtient finalement :

$$G = \sum_i G_i.R.\Sigma_{s_is_i}^{-1}.R^t.G_i^t + \sum_j G_j.\Sigma_{m_jm_j}^{-1}.G_j^t$$

$$G_i = \frac{1}{2}\left(\frac{\partial s_{Ti}}{\partial T}\right)^t . \left\{ \begin{array}{c} \sum_j A_{Tij}\left(H_{s_{Ti}} D(s_{Ti},m_j)\right) \\ -\sum_j A_{Tij} \frac{dD(s_{Ti},m_j)}{ds_{Ti}}^t . \frac{dD(s_{Ti},m_j)}{ds_{Ti}} \\ +\left(\sum_j A_{ij}.\frac{dD(s_{Ti},m_j)}{ds_{Ti}}\right)^t \left(\sum_k A_{ik}.\frac{dD(s_{Ti},m_k)}{ds_{Ti}}\right) \end{array} \right\}$$

$$G_j = \frac{1}{2}\left(\frac{\partial s_{Ti}}{\partial T}\right)^t . \left\{ \begin{array}{c} \sum_j A_{Tij}\left(H_{s_{Ti}} D(s_{Ti},m_j)\right) \\ -\sum_j A_{Tij} \frac{dD(s_{Ti},m_j)}{ds_{Ti}}^t . \frac{dD(s_{Ti},m_j)}{ds_{Ti}} \\ +\left(\sum_j A_{ij}.\frac{dD(s_{Ti},m_j)}{ds_{Ti}}\right)^t \left(\sum_k A_{kj}.\frac{dD(s_{Tk},m_j)}{ds_{Tk}}\right) \end{array} \right\}$$

Nous reviendrons sur cette formule de prédiction de l'incertitude adaptée à l'EM dans le cadre de la prédiction théorique de la précision du recalage sur les surfaces, section 5.4.

4.6 Parallélisation

4.6.1 Introduction

Lorsque l'on dispose de plusieurs processeurs, il peut être intéressant de répartir les calculs entre eux, afin de profiter de toute la puissance disponible et accélérer ainsi le recalage. Ceci peut arriver dans deux cas de figures différents :

– Les machines multi-processeurs : les machines équipés de plusieurs processeurs, autrefois extrêmement chères et spécialisées, sont devenues courantes, à l'image des PC bi-pros disponibles chez Epidaure. Il faut savoir que sur ce type de PC, un programme normal ne sera exécuté que sur un seul processeur. L'intérêt est de pouvoir faire tourner plusieurs programmes en même temps. Cependant, à l'aide d'une programmation spécifique (dite "par processus allégés", "multi-thread" en anglais), on pourra adapter un programme pour qu'il se divise en plusieurs processus qui seront répartis sur les différents processeurs. Ces différents processus pourront communiquer rapidement entre eux et partager la même mémoire mais il faudra alors faire attention qu'il n'écrivent pas au même endroit en même temps, ce qui peut rendre la parallélisation assez délicate.

– Les réseaux de machines : lorsque plusieurs machines sont connectées entre elles via un réseau, on peut les faire communiquer pour qu'elles se répartissent le travail. Le partage de mémoire sera en revanche impossible, et les communications assez lentes (de l'ordre de 0.1 ms).

Nous ne nous consacrerons ici qu'au deuxième cas, plus général. En effet, on pourra toujours lancer deux instances du programme sur un bi-pro et les faire communiquer comme si elles étaient sur des machines différentes. Les programmes seront alors répartis sur les deux processeurs par le système d'exploitation, réalisant ainsi la parallélisation. Ils ne pourront en revanche pas partager la mémoire (on aura donc deux copies des mêmes données) et les communications seront plus lentes. Ceci ne sera heureusement pas gênant dans notre cas, car le recalage utilise peu de mémoire (quelques méga-octets pour des nuages de 100.000 points), et ne nécessite pas de communications intensives et rapides, comme nous le verrons par la suite.

Pour paralléliser un algorithme, il faut distinguer (voir figure 4.3 86) :

– les parties parallèles : il s'agit des parties où plusieurs calculs indépendants doivent être menés. Dans le cas de l'ICP et ses variantes, il s'agit des traitements pour chaque point de la scène (la recherche des voisins, le calcul des distances, des poids, ...), qui sont indépendants pour deux points de la scène différents. C'est bien évidement ces calculs que nous devrons répartir entre les différentes machines.
– les parties séquentielles : il s'agit des calculs qui ne peuvent pas être rendus indépendants, et ne peuvent donc être parallélisés. Dans notre cas, ce sera le calcul de la nouvelle estimée de la transformation. D'une part, ce calcul dépend des résultats des calculs pour chaque point de la scène, et on doit donc attendre que tous les points de la scène aient été traités avant de pouvoir l'effectuer. D'autre part, il ne peut lui-même pas se décomposer en calculs indépendants, et ne peut donc être effectué que sur une seule machine.

Le gain de temps entre la version classique et la version parallèle de l'algorithme est conditionné par les temps relatifs des calculs parallèles et séquentiels. En notant t_s le

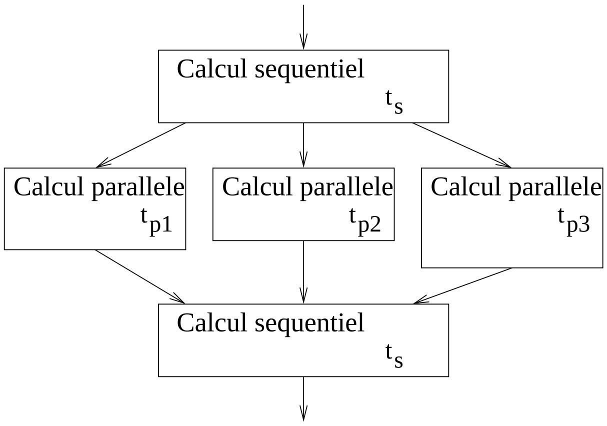


FIG. 4.3 – *Division des calculs en phases parallèles et séquentielles.*

temps total des calculs séquentiels (identique dans les deux versions), t_p le temps des calculs parallèles dans la version classique et t_{p_i} les temps des calculs parallèles effectués sur la machine i dans la version parallélisée, on peut exprimer le temps de la version classique $T_{classique} = t_s + t_p$ et le temps de la version parallèle $T_{parallele} = t_s + \max(t_{p_i})$. Le temps optimum sera atteint quand d'une part la parallélisation en elle-même n'introduit pas de temps supplémentaire (dû par exemple à la communication entre les machines ou à l'algorithme de répartition des calculs entre les machines), i.e. quand $\sum t_{p_i} = t_p$, et quand, d'autre part les calculs sont équitablement répartis entre les machines, i.e. quand les t_{p_i} sont tous égaux. On a alors $t_{p_i} = t_p/n$ où n est le nombre de machines. Le rapport optimum entre les temps de calcul des deux versions est donc : $\frac{T_{parallele}}{T_{classique}} = \frac{\max(t_{p_i})+t_s}{t_p+t_s} = \frac{t_p/n+t_s}{t_p+t_s}$ que l'on peut écrire plus simplement :

$$\frac{T_{parallele}}{T_{classique}} = \frac{1-F}{n} + F \quad avec \quad F = \frac{t_s}{t_p + t_s}$$

Cette loi est appelée loi d'Amdahl et donne donc le facteur d'accélération maximum possible, en fonction du pourcentage de calcul séquentiel.

Dans le cas de l'ICP, ce pourcentage est très faible (le temps de calcul de la nouvelle transformation est négligeable), si bien que le temps de calcul est théoriquement divisé par un facteur proche du nombre de machines mises en jeux, à condition de bien répartir les calculs et ne pas perdre de temps avec les instructions de la parallélisation elles-mêmes.

4.6.2 Parallélisation statique

Nous allons dans un premier temps montrer comment répartir les calculs entre les différentes machines sans chercher à rendre cette répartition efficace. Pour cela nous allons diviser la scène en plusieurs sous-nuages qui seront traités sur chaque machine. Nous verrons comment regrouper ensuite les résultats des traitements. Nous utiliserons pour cela une bibliothèque appelée MPI, que nous allons maintenant présenter.

MPI

Nous devons faire communiquer les machines entre elles. Nous utiliserons pour cela le protocole MPI, qui présente l'avantage d'être standardisé et disponible sur la plupart des machines et systèmes d'exploitation possible (et en particulier sur les PC Linux utilisés chez Epidaure et sur les PC Windows utilisés chez AREALL). Les différentes versions de MPI se composent de :

- une bibliothèque de fonctions dédiées à la communication entre machines que l'on peut directement utiliser en C et C++.
- un moteur (un exécutable) permettant de lancer simultanément sur plusieurs machines plusieurs instances d'un même programme (ou même plusieurs programmes différents) ainsi que les modules qui leur permettront effectivement de communiquer grâce aux fonctions de la bibliothèque.

Lorsqu'on lance les différentes instances à l'aide du moteur, celui-ci va leur attribuer des numéros entre 0 et $n-1$ (où n est le nombre d'instances). On pourra en particulier faire jouer un rôle particulier à l'une de ces instances (la 0 en général), que l'on appellera alors le maître, les autres étant les esclaves.

Nous aurons besoin des fonctions MPI suivantes :

- MPIInit(), MPICommrank(int& id_proc), MPICommsize(int& nb_proc) : ces fonctions permettent d'initialiser le processus de communication en début de programme et de connaître le numéro d'identification de l'instance ainsi que le nombre total d'instances.
- MPIReduce(variables, int id_maitre) : cette fonction additionne les variables des esclaves aux variables du maître, identifié par son numéro d'instance (souvent 0).
- MPIBroadcast(variables, int id_maitre) : cette fonction copie les variables du maître dans celles des esclaves.

Ces fonctions sont bloquantes : il faut attendre que tous les processeurs aient appelé la même fonction avant qu'elle ne libère la main et laisse ainsi chaque instance poursuivre ses calculs. Il sera donc préférable de ne les utiliser qu'en début et fin de phase parallèle pour synchroniser les différents processus à ces moments là, et de ne pas les appeler pendant la phase parallèle elle-même pour ne pas bloquer les calculs.

Parallélisation formelle de l'ICP/EM

Rappelons maintenant l'algorithme de l'ICP/EM :

Initialisation : Calculer une première estimée de T, et mettre le paramètre d'échelle σ^2 à sa valeur initiale.

Répéter

// Décimation : Décimer la scène avec un rayon de sphère $\alpha.\sigma$.

Étape E : Pour chaque point s_i ide la scène décimée avec le poids de décimation n_{s_i}:
Chercher les points du modèle m_j tels que $\|T \star s_i - m_j\|^2 < \sigma^2.\mu_{max}^2$ en utilisant un kD-Tree.
Calculer les poids $\overline{(A_T)_{ij}}$ suivant l'eq. 4.12

Étape M : Ré-estimer la transformation T en minimisant $\sum_{ij} n_{s_i} \overline{(A_T)_{ij}}.\|T \star s_i - m_j\|^2$.

Recuit simulé : Diviser σ^2 par le coefficient de recuit. Si σ^2 est inférieur à l'échelle finale, le mettre à l'échelle finale.

Jusqu'à convergence

Nous devons maintenant nous demander quelle est la plus grande partie que nous pouvons paralléliser. On ne peut évidemment pas effectuer plusieurs itérations simultanément car une itération dépend de la précédente.

A l'intérieur d'une itération, les calculs pour chaque point de la scène sont indépendants. On peut donc répartir les points entre les différents processeurs et ne récupérer que leur résultat sur un maître. Mais jusqu'où peut-on synthétiser le résultat sur un processeur donné pour limiter au maximum les communications et les déséquilibres entre maître et esclaves ? Les grandeurs qui nous intéressent sont les suivantes :

- Le critère $C(T) = \sum_i log \sum_j p(s_i/T \star m_j)$ et ses variantes (e.g. la distance quadratique moyenne entre points appariés).
- La nouvelle transformation, dont le calcul par la méthode de la SVD (voir 3.2.4) est basé sur les barycentres des nuages pondérés comme $\overline{s} = \frac{\sum_i(\sum_j \overline{A_{ij}} s_i)}{\sum_i(\sum_j \overline{A_{ij}})}$ pour la scène et sur la matrice de corrélation en repère barycentrique $K = \frac{\sum_i(\sum_j \overline{A_{ij}}.m_j.s_i^t)}{\sum_i(\sum_j \overline{A_{ij}})} - \overline{m}.\overline{s}^t$.
- Les statistiques comme la distance quadratique moyenne $\frac{\sum_i(\sum_j \overline{A_{ij}} d^2(s_i,m_j))}{\sum_i(\sum_j \overline{A_{ij}})}$.

On peut constater que toutes ces grandeurs peuvent être définies comme des accumulations de grandeurs pour chaque point de la scène, i.e. mises sous la forme $F = \sum_i f(s_i)$: on peut donc se contenter de transmettre les valeurs par point de la scène $f(s_i)$ voire les accumuler sur chaque machine et ne transmettre que les valeurs pour des groupes de points $\sum_{\{i_1,\ldots,i_p\}} f(s_i)$.

Finalement nous pouvons sur un processeur effectuer les calculs pour un groupe de points donné, accumuler les valeurs $f(s_i)$, et attendre que tous les points aient été traités pour effectuer les communications (l'addition finale sur le maître des sommes partielles de chaque instance, à l'aide de la fonction MPIReduce) et les calculs séquentiels sur le maître (le calcul de la nouvelle transformation, les affichages, le passage à l'itération suivante). Il faudra ajouter à cela des communications au début de la nouvelle itération (la communication de la nouvelle transformation aux esclaves, à l'aide de la fonction MPIBroadcast). Ainsi le temps de communication et le temps séquentiel est extrêmement réduit.

En ce qui concerne la répartition des points, nous nous contenterons ici d'une répartition équitable en nombre entre les différentes instances du programme. L'instance de numéro id devra donc traiter les points $int((id * nb_{points})/nb_{process})$ à $int(((id+1) * nb_{points})/nb_{process}) - 1$ où int désigne la partie entière. C'est tellement simple qu'aucune communication ne sera nécessaire pour faire connaître cette répartition.

Lors de l'implémentation effective de l'algorithme, il faudra bien vérifier que la version parallèle produit bien exactement les mêmes résultats que la version classique. Nous l'avons bien évidement systématiquement vérifié, et n'aborderons plus ce point par la suite.

4.6.3 Parallélisation dynamique

La parallélisation statique permet d'atteindre des vitesses satisfaisantes (supérieures à la moitié de la vitesse maximale théorique). Les temps pour chaque point de la scène peuvent varier énormément (en fonction du nombre de points du modèle à proximité), et la répartition équitable en nombre de points à traiter n'est donc pas synonyme de répartition équitable en temps. Mais ces déséquilibres sont atténués par le grand nombre de points si bien que les temps de calculs de chaque instance restent voisins. Cependant ces résultats se dégradent lorsqu'apparaissent des déséquilibres entre la puissance des différentes machines, soit parce qu'elles sont différentes soit parce que certaines sont déjà occupées. La machine la plus lente fait alors attendre toutes les autres pour la phase séquentielle. Ainsi nous avons comparé les résultats entre une grappe de n bi-pros à 935 MHz et une grappe de n bi-pros à 935 MHz et 1 bi-pro à 500 MHz. Nous avons tracé le temps de calcul en fonction de la puissance totale disponible (figure 4.4 gauche). L'expérience confirme qu'un seul processeur moins rapide retarde significativement les autres. Le rendement (rapport entre temps effectif et temps théorique) varie entre 50 et 80 %.

Pour remédier à cet inconvénient, on utilisera une technique usuelle de répartition dynamique des calculs appelée vol de tache : un processeur qui a fini ses calculs propose au plus chargé de lui reprendre une partie de ses calculs pour le soulager. Ceci suppose que les processeurs puissent communiquer entre eux pour se tenir informés de l'avancement de leur tache. On peut envisager plusieurs stratégies :

- Une stratégie maître/esclave : un processeur joue ici un rôle particulier, celui de centraliser les informations. Chaque esclave ne peut communiquer qu'avec le maître, ce qui simplifie beaucoup le nombre de messages en circulation.

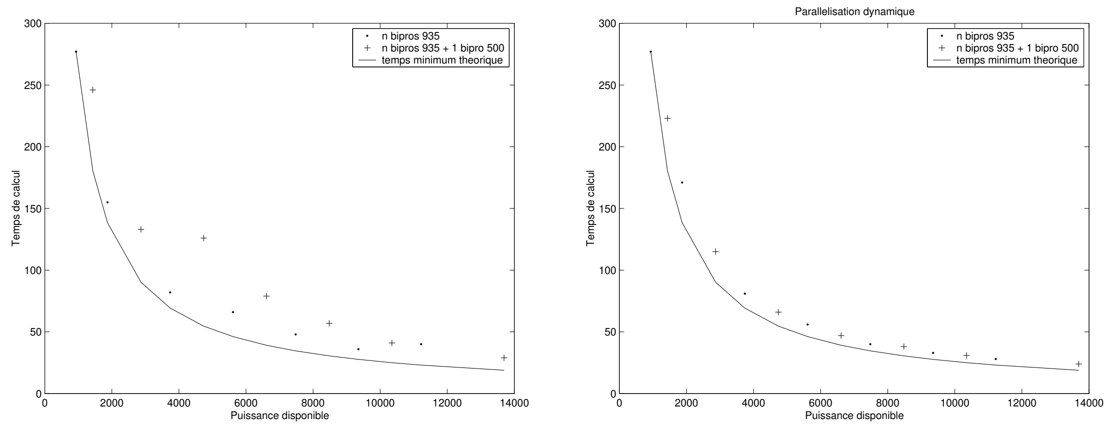


FIG. 4.4 – *Temps de calcul de l'ICP en fonction de la puissance disponible : temps théorique sur une seule machine, temps mesuré sur une grappe homogène et sur une grappe hétérogène. Les rendements se situent entre 50 et 80 % pour la parallélisation statique (à gauche) et sont supérieurs à 90 % pour la parallélisation dynamique (à droite).*

– Une stratégie complètement connecté : aucun processeur ne joue de rôle particulier, et tous peuvent communiquer avec tous. Ceci permet d'accélérer les communications (on ne passe par aucun intermédiaire), mais il peut y avoir plusieurs messages en circulation au même moment, ce qui peut devenir complexe à gérer.
– Une stratégie cyclique : chaque processeur ne peut communiquer qu'avec le suivant (et le dernier avec le premier), formant un cycle. Ainsi, tous jouent le même rôle et il devient très simple de contrôler le nombre de messages en circulation. En revanche, la communication entre deux processeurs éloignés sur le cycle passera par plusieurs intermédiaires et sera donc plus lente (de l'ordre de 1 ms pour 10 processeurs).

Nous choisirons une stratégie cyclique, préférant la simplicité à la rapidité. Nous simplifierons encore en faisant circuler en permanence un unique message sur le cycle. Pour cela, nous utiliserons les fonctions de communication non-bloquantes entre deux machines de MPI :

– ISend(destinataire, message, id_message) : demande au moteur MPI d'envoyer un message à un destinataire donné. L'envoi n'est pas forcément effectué immédiatement, mais la main est rendue au programme, permettant ainsi de poursuivre les calculs. Un numéro d'identification est donc donné au message, permettant de tester ultérieurement s'il a bien été envoyé.
– IRecv(expéditeur, message, id_message) : demande au moteur MPI de se préparer à recevoir un message d'un expéditeur donné. La main est rendue immédiatement au programme, permettant ainsi de poursuivre les calculs. Un numéro d'identification est donc donné au message, permettant de tester ultérieurement s'il a été reçu.
– Test(id_message) : permet de tester si le message a été effectivement envoyé (suite à ISend) ou reçu (suite à IRecv). Dans tous les cas la main est rendue immédiatement.

– Wait(id_message) : attend que le message ait été effectivement envoyé (suite à ISend) ou reçu (suite à IRecv). Cette instruction est donc bloquante.

Ainsi, nous ne perdrons pas de temps avec la communication. L'inconvénient est que nous ne serons pas immédiatement prévenu qu'un message est arrivé, il faudra attendre pour cela l'appel effectif de la fonction Test. En pratique, nous appellerons la fonction Test après le calcul de chaque point de la scène, car la fréquence de traitement est inférieure à celle des communications ($0.1ms$ par point).

Le message que nous ferons circuler contiendra les informations suivantes :

– Le nombre de points déjà traités par chaque processeur : ceci permettra d'avoir une idée de la vitesse de chacun pour pouvoir optimiser la répartition.
– Le nombre de points restant à traiter : ceci permettra de savoir quels processeurs sont disponibles (ceux qui n'ont plus de points à traiter) et lesquels sont encore très chargés.
– Une instruction éventuelle, permettant d'indiquer les opérations à faire pour réaliser effectivement le vol de tâche : il y a deux instructions possibles : la proposition de vol (l'instance i propose de soulager l'instance j), l'acceptation de vol (l'instance i confie les points n_1 à n_2 à l'instance j).

L'algorithme de répartition dynamique doit tenir compte de l'aspect périmé des messages : le temps que le message transite entre l'instance i et j, l'instance i aura traité de nouveaux points. Les informations la concernant ne seront pas à jour. L'instance j, si elle est libre, ne peut donc pas se lancer aussitôt dans le traitement de points d'une autre instance, car celle-ci les a peut-être déjà traités entre temps. Elle doit donc se contenter de proposer ses services et attendre qu'on lui réponde.

La question est alors de savoir à qui elle offre ses services. Elle peut le décider elle-même en jugeant la vitesse et la charge de chaque processeur, informations hélas périmées. On pourrait alors penser laisser ce choix aux autres processeurs, au moment où ils reçoivent le message : le premier processeur accepte le vol de tache, puis le deuxième lui reprend s'il est plus chargé (il juge alors sur des informations récentes) et ainsi de suite. Mais il faudrait alors signaler au premier processeur que ce n'est finalement pas lui qui a été délesté, ce qui complique la communication.

Nous avons finalement opté pour la solution suivante : l'instance libre choisit l'instance qu'elle va délester et envoie le message adéquat, et l'instance à délester choisit lorsqu'elle le reçoit le nombre de points optimal à donner, et envoie le message adéquat. Finalement, le vol de tache se fait en un seul cycle de communication.

Le choix du nombre de points à voler se fait de la façon suivante : si v_i est le nombre de points déjà traité et c_i le nombre de points restant pour l'instance i ($c_j = 0$ pour l'instance libre), la nouvelle répartition des points c_i' doit être proportionnel au nombre de points déjà traité :

$$c'_i = \frac{v_i}{\sum v_i} \sum c_i$$

Si le temps de traitement des points était uniforme, cette répartition serait optimale (le temps de calcul restant, proportionnel à c_i/v_i, est identique pour toutes les instances). Enfin, pour le choix de la meilleure instance à délester, on calcule c'_i pour chaque instance encore chargée, et on choisit celle pour laquelle le gain de temps $(c_i - c'_i)/v_i$ est le plus important.

L'algorithme final au niveau de chaque instance est finalement le suivant :

Si le message a été reçu par l'instance j

Mise à jour : On met à jour les informations relative à l'instance j (nombre de points déjà traités et restant à traiter)

Confirmation : Si une instruction du type acceptation de vol concernant l'instance j est lancée (l'instance i confie les points n_1 à n_2 à l'instance j)
On lance le traitement des points n_1 à n_2 sur l'instance j, et on met à jour le message (nouveau nombre de points à traiter pour j, effacement de l'instruction de vol)

Acceptation : Si une instruction du type proposition de vol concernant l'instance j est lancée (l'instance i propose de soulager l'instance j)
On calcule le nombre de points optimum à confier à l'instance i : $c'_i = c_j \frac{v_i}{v_i+v_j}$, on supprime autant de points de la liste des points à traiter par l'instance j, et on met à jour le message (nouveau nombre de points à traiter pour j, nouvelle instruction : l'instance j confie les points n_1 à n_2 à i)

Proposition : Si aucune instruction n'est lancée et que l'instance j a traité tous ses points
On calcule le nombre de points optimum à confier aux différentes instances $c'_i = c_i \frac{v_i}{v_i+v_j}$, on en déduit l'instance i qui a le plus besoin d'aide (celle qui maximise $(c_i - c'_i)/v_i$), et on met l'instruction : l'instance j propose de soulager l'instance i.

Renvoi : On envoie le message au processeur suivant

Cet algorithme permet d'entretenir la communication cyclique (le message est dans tous les cas renvoyé au processeur suivant immédiatement après la mise à jour) et de gérer le vol des tâches tel que nous l'avons présenté.

Les temps de calcul mesurés avec cette nouvelle version de l'algorithme (figure 4.4 droite) montre que cette fois, la machine la plus lente n'a pas ralenti le processus, et le rendement est dans tous les cas de l'ordre de 90%, donc largement satisfaisant.

4.6.4 Conclusion

La parallélisation de l'algorithme ICP/EM est simple. Il n'y a en-effet aucun besoin de partage de la mémoire ou de communication intensive entre machines. La seule difficulté est de bien équilibrer les charges des différents processeurs, ce qu'un algorithme de répartition dynamique réalise très bien. Les essais menés sur une grappe de 15 processeurs ont montré que le temps de calcul atteint était pratiquement divisé par le nombre de processeurs mis en jeux.

4.7 Conclusion

Nous avons dans ce chapitre exprimé la vraisemblance d'un recalage à partir d'un modèle de bruit sur les données exprimé par la probabilité $p(T \star s_i|m_j)$ d'obtenir un point s_i de la scène connaissant le point m_j du modèle qui lui correspond. Nous avons alors dérivé deux algorithmes permettant de trouver la transformation qui maximise cette vraisemblance :

- En utilisant seulement la configuration d'appariements la plus probable : on a retrouvé l'ICP dans le cas gaussien, et montré comment le généraliser pour des modèles de bruit plus complexes.
- En utilisant toutes les configurations d'appariements possibles, pondérées par leur probabilité a posteriori : on a alors dérivé un nouvel algorithme de recalage basé sur les idées de l'EM, que nous avons appelé ICP/EM dans le cas gaussien.

L'ICP/EM dépendant explicitement de la variance du bruit sur les données, nous avons fait varier ce nouveau paramètre et avons constaté que l'algorithme présentait des propriétés caractéristiques des approches multi-échelle : un comportement précis et peu robuste (car proche de l'ICP) pour de faibles variances, un comportement robuste mais peu précis (car proche de l'alignement des centres et tenseurs d'inerties) pour de grandes variances. Nous l'avons logiquement incorporé dans une stratégie de raffinement progressif de l'échelle qui a permis d'améliorer notablement la robustesse du recalage. Nous l'avons enfin associé à une approche multi-résolution, grâce à une technique simple et efficace de décimation, pour éviter l'explosion des temps de calcul.

Les résultats de ces travaux sont extrêmement encourageants :

- Le schéma multi-échelle permet un recalage bien plus robuste et un peu plus précis que l'approche ICP classique.
- L'utilisation de la décimation et la parallélisation de l'algorithme permet d'atteindre des temps de recalages très inférieurs à la seconde sur nos données, et on peut donc envisager des applications temps réels.

Nous nous tiendrons donc à cette approche (ICP/EM avec décimation et raffinement progressif de l'échelle) dans la suite de cette thèse. Pour l'adapter à différents problèmes, nous profiterons de l'entière liberté laissée dans le choix de la loi de probabilité $p(T \star s_i | m_j)$. Le prochain chapitre présente plusieurs variantes simples, et le chapitre 8 des variantes plus complexes reposant sur un véritable modèle probabiliste de surfaces.

Chapitre 5

Adaptation au recalage de surfaces

Sommaire

Les algorithmes que nous avons jusqu'ici présentés (dans l'état de l'art comme dans le chapitre sur l'EM) ne sont pas spécifiquement dédiés aux surfaces. Ils font même l'hypothèse que les points des deux nuages sont identifiés et sont donc théoriquement adaptés aux nuages d'amers géométriques (comme des marqueurs, ou des points particuliers tels les points de crêtes). Il se trouve que ces algorithmes fonctionnent relativement bien avec les courbes et les surfaces. Nous avons tout de même cherché à tenir compte de l'origine surfacique de nos nuages de points pour améliorer nos résultats. Nous présentons ici plusieurs de ces variantes originales, indépendantes les unes des autres :

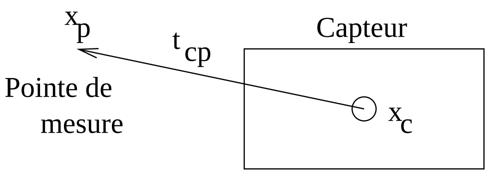
- La correction de décalage a pour but d'évaluer un biais systématique dans la direction normale à la surface. L'estimer permet de le corriger, et éviter ainsi un biais dans le résultat du recalage (section 5.1).
- L'utilisation de la distance entre normales a pour but d'améliorer l'estimation des appariements en rejetant les points dont les normales sont trop éloignées. Elle permet essentiellement d'améliorer la robustesse aux points aberrants (section 5.2).
- Les points semi-anisotropes permettent de donner plus d'importance à l'information dans la direction normale à la surface que dans la direction tangentielle. Ils permettent un gain en robustesse (section 5.3).

Enfin, nous abordons le problème de l'adaptation de la prédiction de l'incertitude théorique aux surfaces.

5.1 Correction de décalage normal

Un problème que nous avons rencontré est celui de la présence d'un décalage systématique sur la position des points dans la direction indiquée par la normale. Il peut apparaître sur deux types de nuages de points que nous utilisons : les points per-opératoires (voir 1.5.3) et les surfaces segmentées dans le scanner (voir 1.5.1). Nous détaillons ici ces deux cas.

Points per-opératoires Ils sont mesurés à l'aide d'une pointe reliée rigidement à un capteur de position/orientation :



Le système de suivi permet de mesurer la position du centre du capteur x_c et son orientation, donnée par sa rotation R_c par rapport à une position de référence. Si l'on connaît maintenant le vecteur t_{cp} entre le centre du capteur et la pointe de mesure dans cette position de référence, on peut en déduire à tout moment la position de la pointe

$x_p = x_c + R_c t_{cp}$. Le vecteur t_{cp} est en fait estimé une bonne fois pour toute pour chaque ensemble capteur/pointe, par une méthode de calibration spécifique. Mais celle-ci peut donner des résultats imprécis, générant une erreur systématique sur la position de la pointe. Nous nous contenterons ici de modéliser l'erreur dans l'axe de la pointe.

En résumé, notre système d'acquisitions de points nous donne la position biaisée de la pointe x_p et l'orientation de son axe (qui est déduite de la matrice R_c), que nous utilisons directement ensuite comme estimée de la normale à la surface n_p. Si δ est le décalage, i.e. l'erreur sur t_{cp} dans l'axe de la pointe, on peut retrouver la véritable position de la pointe x_v en appliquant la formule suivante :

$$x_v = x_p - \delta n_p$$

Surfaces segmentées dans le scanner Les problèmes de décalages peuvent aussi apparaître lorsqu'on extrait les surfaces des scanners en utilisant un “Marching-Cube” basé sur le seuillage direct de l'intensité des voxels. Cet algorithme extrait en-effet l'isosurface correspond à une intensité seuil donnée. Si cette intensité seuil est sous-estimée, la surface sera décalée vers les intensités faibles, et inversement. La direction du décalage correspond donc à la direction dans laquelle varie l'intensité, i.e. au gradient de l'image. Or nous utilisons justement ce gradient comme estimée de la normale à la surface n_p. Nous avons donc la même relation entre la coordonnée mesurée du point x_p et sa coordonnée véritable x_v que dans le cas précédent :

$$x_v = x_p - \delta n_p$$

Cependant, le décalage δ n'est plus fixe, mais dépend de la vitesse de variation de l'intensité dans l'image (i.e. la norme du gradient). Il est même inversement proportionnel à cette vitesse de variation. Heureusement, nous avons vérifié que cette vitesse de variation est relativement constante (les interfaces entre les tissus osseux et les autres tissus présentent en-effet à peu près les mêmes profils d'intensité), et nous utiliserons donc un δ fixe par la suite.

5.1.1 Conséquence du décalage normal sur le recalage

Nous allons ici montrer quelles peuvent être les conséquences du décalage normal sur le recalage de surfaces, en fonction de la distribution des points. Nous supposerons pour simplifier que seuls les points de la scène sont corrompus par un décalage normal.

Appariements correctement estimés Nous supposerons dans un premier temps que le décalage n'induit pas d'erreurs dans l'estimation des appariements, et comparerons le résultat du recalage avec les vrais points et avec les points biaisés, en s'appuyant sur la méthode de la SVD (voir 3.2.4).

Le barycentre pondéré des points de la scène est, pour les points de la scène non biaisés :

$$\overline{s} = \frac{\sum_{ij} A_{ij} s_i}{\sum_{ij} A_{ij}}$$

Il devient, lorsqu'on ajoute un décalage qui transforme le point s_i en $s_i + \delta n_{s_i}$:

$$\overline{s}' = \frac{\sum_{ij} A_{ij}(s_i + \delta n_{s_i})}{\sum_{ij} A_{ij}} = \overline{s} + \delta \frac{\sum_{ij} A_{ij} n_{s_i}}{\sum_{ij} A_{ij}}$$

Les matrices de corrélation en repère barycentrique sont :

$$K = \frac{\sum_{ij} A_{ij}.(m_j - \overline{m}).(s_i - \overline{s})^t}{\sum_{ij} A_{ij}} = \frac{\sum_{ij} A_{ij}.m_j.s_i^t}{\sum_{ij} A_{ij}} - \overline{m}.\overline{s}^t$$

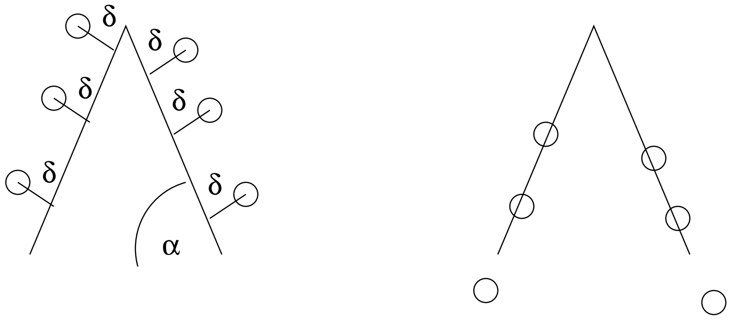
$$K' = \frac{\sum_{ij} A_{ij}.m_j.(s_i + \delta n_{s_i})^t}{\sum_{ij} A_{ij}} - \overline{m}.(\overline{s} + \delta \frac{\sum_{ij} A_{ij} n_{s_i}}{\sum_{ij} A_{ij}}) = K + \delta \left(\frac{\sum_{ij} A_{ij}.(m_j - \overline{m}).n_{s_i}^t}{\sum_{ij} A_{ij}} \right)$$

La décomposition SVD de ces matrices qui permet le calcul des rotations étant non linéaire, il est difficile de tirer une conclusion générale de ces formules. Cependant, lorsque le décalage δ est négligeable devant l'extension spatiale de la scène (ce qui est heureusement le cas, les décalages faisant quelques dixièmes de mm pour une scène de plusieurs cm), δn_{s_i} restera négligeable devant $s_i - \overline{s}$, et le terme $\delta \left(\frac{\sum_{ij} A_{ij}.(m_j - \overline{m}).n_{s_i}^t}{\sum_{ij} A_{ij}} \right)$ restera ainsi négligeable devant K. Les matrices K et K' seront sensiblement égales, ainsi que leur décompositions SVD et donc les rotations obtenues.

On peut par contre directement se faire une idée de l'influence du biais sur la translation, puisque celle-ci est égale à $t = R\overline{s} - \overline{m}$. Si la rotation est effectivement proche pour le recalage avec et sans biais, nous obtiendrons une différence pour la translation de l'ordre de $\delta \frac{\sum_{ij} A_{ij} n_{s_i}}{\sum_{ij} A_{ij}}$. Cette erreur dépend donc de la répartition des normales. Si elles ne privilégient aucune direction (par exemple lorsqu'on recale deux sphères ou deux cubes complets), la moyenne pondérée des normales $\frac{\sum_{ij} A_{ij} n_{s_i}}{\sum_{ij} A_{ij}}$ et donc l'erreur sur le recalage seront faibles. En revanche, si les normales sont mal réparties (ce qui est par exemple le cas avec la mâchoire inférieure, où les normales sont plutôt dirigées vers le haut) et pointent en moyenne vers la direction $\frac{\sum_{ij} A_{ij} n_{s_i}}{\sum_{ij} A_{ij}}$, nous retrouverons une erreur sur la translation dans cette direction (verticale dans le cas des mâchoires) qui restera inférieure à δ.

Appariements mal estimés Ce résultat pourrait paraître rassurant, dans la mesure où les décalages restent assez faibles (inférieurs à $0.2\,mm$). Mais il est hélas inexact, car l'erreur sur l'estimation de la transformation va générer des erreurs sur l'estimation des appariements qui vont générer des erreurs sur l'estimation de la transformation et ainsi de suite. Les erreurs, si elles existent (i.e. si $\frac{\sum_{ij} A_{ij} n_{s_i}}{\sum_{ij} A_{ij}}$ n'est pas négligeable) vont donc s'amplifier.

Il est impossible d'exprimer explicitement l'erreur ainsi obtenue. On peut par contre l'illustrer sur l'exemple d'un cône :



Ici le modèle est représenté par le trait continu, les points de scène par les cercles. On voit à gauche les points dans leur position correcte. On voit à droite ce que va logiquement donner le recalage sur un pareil cas à cause de du décalage δ : la translation va être décalée de $\delta/\sin\alpha$ vers le bas, et l'erreur peut ainsi devenir très importante.

5.1.2 Estimation et correction du décalage

Comme dans le paragraphe précédent, nous supposerons dans un premier temps que le problème de décalage n'est présent que sur les points de la scène. D'après ce qui précède, il devient nécessaire de prendre en compte le décalage dès lors que les normales ont une moyenne pondérée $\frac{\sum_{ij} A_{ij} n_{s_i}}{\sum_{ij} A_{ij}}$ non négligeable. Nous allons voir ici comment l'estimer et le corriger en l'intégrant directement dans le procédé statistique qui nous a servi à dériver, dans le chapitre précédent, les algorithmes ICP et ICP/EM. Ce procédé s'appuyait sur la maximisation de la vraisemblance, i.e. de la probabilité des données sachant la transformation et les appariements. Nous procéderons de même avec le paramètre de décalage δ. Nous donnerons l'algorithme général (pour un modèle de bruit quelconque), et l'illustrerons dans le cas gaussien isotrope, où nous exhiberons une solution explicite.

Supposons donc le décalage δ connu. Sa correction est alors triviale : il suffit de remplacer le point s_i par le point corrigé $s_i - \delta n_{s_i}$. Ce point corrigé est le point réel de la surface. C'est donc lui qui suit la loi de probabilité donnée par l'équation 4.1. Il suffit donc de remplacer cette loi de probabilité dans la totalité des équations du chapitre précédent (et en particulier dans les critères) pour obtenir les algorithmes ICP et ICP/EM tenant compte du décalage sur les points de la scène. La nouvelle loi est :

$$p(s_i|m_j,T,\delta) = p(T \star (s_i - \delta n_{s_i})|m_j) \tag{5.1}$$

et le critère de l'ICP (voir 4.3) devient (nous ne donnerons pas ici les détails pour l'EM, les résultats étant identiques) :

$$C_{ICP}(T,A,\delta) = \sum_{ij} A_{ij}.(-\log p(T \star (s_i - \delta n_{s_i})|m_j) - \log \overline{\pi_{ij}}) \tag{5.2}$$

On constate alors que pour les étapes d'estimations des appariements et de la transformation (les étapes E et M dans l'EM), il suffit donc de remplacer les points de la scène par les points corrigés. Ce critère permet de plus de dégager une approche pour l'estimation du décalage δ : nous allons aussi l'estimer en minimisant le critère (i.e. en maximisant la vraisemblance). Le problème est de savoir quand. Rappelons le déroulement de l'algorithme ICP classique :

Initialisation : calcul de T^0

Boucle : à l'étape I :

- Calcul de A^I : minimisation de $C(T,A)$ à $T = T^I$ constant.
- Calcul de T^{I+1}: minimisation de $C(T,A)$ à $A = A^I$ constant.

Jusqu'à convergence

Nous allons intégrer l'estimation du décalage au sein même de cet algorithme de minimisation alternée. A l'évidence, il est plus aisé d'obtenir une estimée initiale de δ que des appariements (il suffit de prendre un décalage nul). Il faut donc placer l'estimation de δ après l'estimation des appariements, et utiliser à la première itération une estimée initiale du décalage δ^0, nulle en général.

Il reste alors le choix entre placer l'estimation du décalage avant l'étape d'estimation de la transformation (il faut alors prendre en compte la nouvelle valeur du décalage pour celle-ci), ou après (il faut alors prendre la nouvelle valeur de la transformation pour l'estimation du décalage). La deuxième possibilité semble préférable car elle laisse la première partie de l'algorithme inchangée. Le choix devra cependant tenir compte de considérations d'efficacité algorithmique, en fonction du modèle de bruit choisi.

Les deux algorithmes possibles sont donc les suivants :

Initialisation : calcul de T^0 et δ^0 (nul en général)

Boucle : à l'étape I :

- Calcul de A^I : minimisation de $C(T,A,\delta)$ à $T = T^I$ et $\delta = \delta^I$ constants.
- Calcul de δ^{I+1} : minimisation de $C(T,A,\delta)$ à $A = A^I$ et $T = T^I$ constants.
- Calcul de T^{I+1}: minimisation de $C(T,A,\delta)$ à $A = A^I$ et $\delta = \delta^{I+1}$ constants.

Jusqu'à convergence

ou (algorithme préférable) :

Initialisation : calcul de T^0 et δ^0 (nul en général)

Boucle : à l'étape I :

- Calcul de A^I : minimisation de $C(T,A,\delta)$ à $T = T^I$ et $\delta = \delta^I$ constants.
- Calcul de T^{I+1}: minimisation de $C(T,A,\delta)$ à $A = A^I$ et $\delta = \delta^I$ constants.
- Calcul de δ^{I+1} : minimisation de $C(T,A,\delta)$ à $A = A^I$ et $T = T^{I+1}$ constants.

Jusqu'à convergence

5.1.3 Cas Gaussien isotrope homogène

Nous avons jusqu'ici présenté l'algorithme pour un modèle de bruit quelconque, mais cette généralité nous a, comme d'habitude, empêché de lui donner une forme explicite. Nous allons ici nous restreindre au modèle de bruit gaussien habituel, et même à un bruit gaussien isotrope homogène, qui permet justement d'obtenir une forme explicite.

Le critère avec un tel modèle est (voir 4.4) :

$$C_{ICP}(T,A,\delta) = \frac{1}{2}\sum_{ij} A_{ij}.d^2(T \star (s_i - \delta n_{s_i}),m_j) = \frac{1}{2}\sum_{ij} A_{ij}.d^2(Rs_i + t - \delta R n_{s_i},m_j) \tag{5.3}$$

La décalage doit minimiser le critère et donc annuler sa dérivé :

$$\frac{\partial C_{ICP}(T,A,\delta)}{\partial \delta} = \sum_{ij} A_{ij}(Rn_{s_i}|Rs_i + t - \delta Rn_{s_i} - m_j) = 0$$

$$\sum_{ij} A_{ij}(Rn_{s_i}|Rs_i) + \sum_{ij} A_{ij}(Rn_{s_i}|t) - \sum_{ij} A_{ij}(Rn_{s_i}|m_j) = \delta \sum_{ij} A_{ij}(Rn_{s_i}|Rn_{s_i})$$

on peut simplifier cette équation en se rappelant que $(Ra|Rb) = (a|b)$ et en particulier $(Rn_{s_i}|Rn_{s_i}) = (n_{s_i}|n_{s_i}) = 1$, et que $(Ra|b) = Tr(Rab^t)$. On obtient alors :

$$\left(\sum_{ij} A_{ij}(n_{s_i}|s_i)\right) + \left(R\left(\sum_{ij} A_{ij}n_{s_i}\right)|t\right) - Tr\left[R\left(\sum_{ij} A_{ij}n_{s_i}m_j^t\right)\right] = \delta\left(\sum_{ij} A_{ij}\right)$$

Cette formule permet, lorsque $\sum_{ij} A_{ij}(n_{s_i}|s_i)$, $\sum_{ij} A_{ij}n_{s_i}$, $\sum_{ij} A_{ij}n_{s_i}m_j^t$ et $\sum_{ij} A_{ij}$ sont connus (i.e. quand les appariements sont connus pour une transformation T^I donnée), de calculer le décalage quelle que soit la transformation. On peut alors aussi bien calculer le décalage avant la transformation (en utilisant T^I dans la formule précédente) qu'après la transformation (en utilisant le nouvelle valeur T^{I+1}). Le calcul de la transformation ne laisse pas toujours cette liberté : si le décalage est calculé avant, les variables qu'elle utilise pourront ne plus être valables (c'est par exemple le cas des gradients, utilisés à la section 3.8.2), et on doit refaire une passe sur les appariements (i.e. ré-estimer A) pour tenir compte du nouveau décalage avant de calculer la nouvelle transformation. Finalement, on peut et on doit ré-estimer le décalage après avoir mis à jour la transformation, comme recommandé.

Cette formule est enfin très intéressante d'un point de vue de la simplicité algorithmique, car elle s'appuie sur quatre accumulateurs, comme la SVD utilisée pour le calcul de la transformation (voir 3.2.4 et 4.6.2). Grâce à ces accumulateurs, il est inutile de stocker les appariements ou d'effectuer plusieurs passes sur l'ensemble des appariements, et on peut facilement paralléliser l'algorithme.

5.1.4 Application

On peut se demander s'il est vraiment raisonnable d'appliquer la correction du décalage durant l'intégralité du recalage. De manière générale, ajouter une nouvelle variable à optimiser dans le critère (i.e. augmenter la dimension de l'espace de recherche) va introduire de nouveaux minima locaux et aggraver les problèmes de robustesse. Ainsi, dans notre cas, les surfaces sont encore assez éloignées lors des premières itérations, et on risque alors d'estimer un décalage important qui n'existe pourtant pas et induire ainsi en erreur la suite de l'algorithme.

Aussi avons-nous comparé plusieurs stratégies différentes :

- On a estimé le décalage à chaque itération dès la première itération.
- On a laissé converger l'algorithme sans estimation du décalage, puis l'avons relancé avec l'estimation du décalage à chaque itération.
- On a laissé converger l'algorithme sans estimation du décalage, on a estimé le décalage, puis avons relancé l'algorithme à décalage fixé, attendu une nouvelle convergence, ré-estimé le décalage et ainsi de suite jusqu'à ce que l'estimation soit stable.

La première stratégie s'est révélée, comme attendue, instable (elle diverge dans 95% des cas). Les deux autres stratégies ne perturbent pas la robustesse du recalage, la dernière étant cinq fois plus lente mais probablement un peu plus précise.

5.1.5 Discussion

Les ingénieurs d'AREALL ont rapidement trouvé des solutions de calibration efficaces pour faire disparaître les problèmes de décalages dans nos données. Nous n'avons donc pas approfondi nos travaux sur la correction du décalage, et n'avons en particulier pas essayé de les valider. Les rares expériences ont donné des indices encourageants (les ordre de grandeur des décalages mesurés semblait corrects, la précision du recalage était légèrement améliorée), mais nous ne les avons pas jugé assez nombreuses et pertinentes pour les présenter ici, et ne pouvons donc affirmer que la méthode marche. Elle est pourtant intéressante d'un point de vue théorique (elle s'intègre parfaitement à l'ICP/EM) et on peut songer à plusieurs variantes :

- La correction du décalage sur le modèle : elle se déduit directement de la précédente, en adaptant les équations.
- La correction conjointe des décalages sur la scène et le modèle : nous avons essayé d'activer à la fois la correction des décalages sur le modèle et la scène. On a alors pu constater que l'algorithme devenait instable, probablement parce que les deux corrections sont concourantes.
- La correction d'un décalage complet (i.e. pas uniquement dans la direction de la normale) : celle-ci s'est avérée instable dans le cadre du recalage de mâchoires, mais a par-contre fourni une méthode de calibration du décalage satisfaisante lorsqu'elle était utilisée sur un cube de métrologie.

5.2 Utilisation de la distance entre normales

Lorsqu'on dispose de points orientés (i.e. munis d'un vecteur unitaire qui représente la normale à la surface), on peut prendre en compte la distance entre normales pour mieux estimer les appariements (en rejetant les points proches qui ont des normales très différentes) et ainsi rendre l'algorithme moins sensible aux artefacts.

5.2.1 Distance quadratique entre points orientés

La seule définition de distance rigoureuse entre normales est leur angle (voir eq. D.5) : $d(T \star n_{s_i}, n_{m_j}) = (\widehat{T \star n_{s_i}, n_{m_j}})$. On obtient alors une distance entre points orientés en utilisant une combinaison linéaire des distances quadratiques entre points et entre normales :

$$D(T \star (s_i, n_{s_i}), (m_j, n_{m_j})) = \alpha. \|Rs_i + t - m_j\|^2 + \beta.(\widehat{Rn_{s_i}, n_{m_j}})^2$$

Remarquons au passage que les deux distances que l'on mélange n'ont pas les mêmes unités : *mm* pour la distance entre points et *radian* pour la distance entre normales. Les coefficients α et β permettent de mélanger ces unités différentes. Une façon de les estimer est d'utiliser la distance de Mahalanobis.

5.2.2 Distance de Mahalanobis simple entre points orientés

La distance de Mahalanobis simple entre normales est, dans le cas isotrope (voir section 7.2.9) :

$$\mu^2(T \star n_{s_i}, n_{m_j}) = \frac{1}{\sigma^2_{n_{m_j}}}(\widehat{Rn_{s_i}, n_{m_j}})^2$$

La distance de Mahalanobis entre points orientés peut être facilement définie et calculée lorsque le bruit sur la position du point et le bruit sur l'orientation de la normale sont supposés indépendants. Il s'agit alors tout simplement de la somme des distances de Mahalanobis entre points et entre normales :

$$\begin{aligned} \mu^2(T \star (s_i,n_{s_i}),(m_j,n_{m_j})) &= \mu^2(T \star s_i,m_j) + \mu^2(T \star n_{s_i},n_{m_j}) \\ &= \tfrac{1}{\sigma^2_{e_{ij}}} \left\|T \star s_i - m_j\right\|^2 + \tfrac{1}{\sigma^2_{n_{m_j}}} (\widehat{Rn_{s_i},n_{m_j}})^2 \end{aligned} \tag{5.4}$$

Il n'existe cependant aucun algorithme permettant de minimiser directement un critère faisant intervenir une telle mesure. Nous avons donc introduit une version approchée de cette distance (la distance vectorielle, voir 7.3.2), qui s'intègre très bien dans l'algorithme de calcul direct de la transformation fondé sur la SVD (voir 3.2.4) :

$$D(T \star n_{s_i},n_{m_j}) = (\widehat{Rn_{s_i},n_{m_j}})^2 \simeq \left\|Rn_{s_i} - n_{m_j}\right\|^2$$

Soit, en guise de distance entre points orientés :

$$D(T \star (s_i,n_{s_i}),(m_j,n_{m_j})) = \frac{1}{\sigma^2_{e_{ij}}}.\left\|Rs_i + t - m_j\right\|^2 + \frac{1}{\sigma^2_{n_{m_j}}}.\left\|Rn_{s_i} - n_{m_j}\right\|^2$$

On voit alors facilement que l'on peux minimiser cette distance par la méthode de la SVD. Le nouveau terme ne dépendant pas de la translation, le calcul de celle-ci est inchangé. Le calcul de la rotation se fait en utilisant $K = \sum_{ij} A_{ij}.\left[(1/\sigma^2_{e_{ij}}).m_j.s_i^t + (1/\sigma^2_{n_{m_j}}).n_{m_j}.n_{s_i}^t\right]$ comme matrice de corrélation.

On peux donc résumer la mise à jour de la transfo par cette série d'équations :

$$\overline{m} = \frac{\sum_{ij}(A_{ij}/\sigma^2_{e_{ij}}).m_j}{\sum_{ij}(A_{ij}/\sigma^2_{e_{ij}})} \qquad \overline{s} = \frac{\sum_{ij}(A_{ij}/\sigma^2_{e_{ij}}).s_i}{\sum_{ij}(A_{ij}/\sigma^2_{e_{ij}})}$$

$$K = \sum_{ij} \left((A_{ij}/\sigma^2_{e_{ij}}).m_j.s_i^t + (A_{ij}/\sigma^2_{n_{m_j}}).n_{m_j}.n_{s_i}^t\right)$$

$$K' = K - (\sum_{ij}(A_{ij}/\sigma^2_{e_{ij}})).\overline{m}.\overline{s}^t$$

$$SVD\, sur\, K' \quad \rightarrow R$$

$$t = \overline{m} - R\overline{s}$$

5.2.3 Discussion

Les surfaces segmentées à partir de scanners avec des artefacts contiennent beaucoup de points aberrants. Avec ces surfaces, même l'ICP/EM multi-échelle s'est avéré souvent incapable de retrouver une transformation correcte lorsqu'on n'utilisait pas les normales.

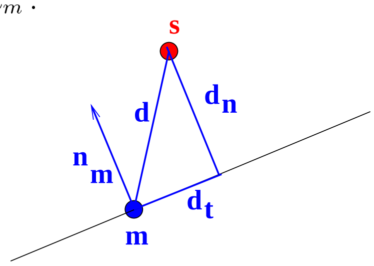
En revanche, en utilisant les normales, des taux de robustesses corrects ont pu être atteints. Sur des surfaces propres (i.e. avec peu ou pas de points aberrants), l'utilisation des normales a aussi permis une amélioration significative de la robustesse. Enfin, en terme de précision, l'utilisation des normales permet de rejeter les appariements les moins pertinents et apporte ainsi une amélioration. Les résultats expérimentaux et le problème du réglage du paramètre (l'écart-type sur l'orientation des normales) sont présentés en détail dans la section 6.4.3.

5.3 Points semi-anisotropes

Nous avions présenté lors de l'état de l'art une technique permettant de prendre en compte l'aspect surfacique des données (voir 3.4.1) mais qui présentait des problèmes d'instabilité. L'idée était d'accorder plus d'importance aux distances dans la direction normale à la surface. En-effet, les nuages de points sont issus de la même surface échantillonnée de deux manières différentes. Les distances dans le plan tangent à la surface correspondent à ces différences d'échantillonnage et ne sont donc pas nécessairement nulles. Mieux vaut donc leur accorder une importance moindre. Nous présentons ici une nouvelle méthode qui s'appuie sur un modèle de bruit semi-anisotrope (isotrope dans le plan tangent à la surface mais différent dans la direction normale) qui s'intègre directement dans notre modèle statistique, et garantit ainsi la stabilité de l'algorithme.

5.3.1 Bruit semi-anisotrope

Nous souhaitons donc accorder une importance différente à la distance d_t dans le plan tangent et à la distance d_n dans le plan normal à la surface en un point m du modèle, plan connu grâce à la normale n_m :



La façon la plus simple d'exprimer mathématiquement cette contrainte dans notre canevas statistique est de supposer que le bruit sur les points (le bruit que nous avons défini dans l'équation 4.1 et qui permet de dériver l'algorithme ICP et EM) est lui-même semi-anisotrope. On peut même utiliser deux modèles de bruit complètement différents dans la direction normale à la surface (qui correspond au bruit de mesure) et dans le plan tangent de la surface (où on observe à la fois le bruit de mesure et l'échantillonnage de la surface). On compose alors ces deux modèles de bruits pour obtenir $p(s_i|m_j,T) = p(T \star s_i|m_j)$ et adapter l'algorithme (ICP ou ICP/EM) en fonction.

5.3.2 Cas gaussien

Nous allons illustrer le paragraphe précédent, volontairement très général, en abordant le cas gaussien. Nous supposerons donc que le bruit de mesure sur le point est composé de deux bruits gaussiens centrés additifs et indépendants, le premier d'écart-type σ_t^2 dans le plan tangent à la surface (i.e. la plan perpendiculaire à n_m) et le second d'écart-type σ^2 dans la direction perpendiculaire à la surface (i.e. dans l'axe de la normale). Ces deux bruits s'additionnent pour former un bruit gaussien dont la covariance Σ_{mm} peut s'exprimer dans la base orthonormale (n_m,e_2,e_3) (où (e_2,e_3) est une base orthonormale du plan tangent) : on a alors

$$\Sigma = \begin{pmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma_t^2 & 0 \\ 0 & 0 & \sigma_t^2 \end{pmatrix} = \sigma_t^2.Id + (\sigma^2 - \sigma_t^2) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

soit dans une base quelconque :

$$\Sigma = \sigma_t^2.Id + (\sigma^2 - \sigma_t^2).n_m n_m^t$$

Nous avons supposé la loi gaussienne, et connaissons sa covariance. On peut donc l'écrire :

$$p(s_i|m_j,T) = p(T \star s_i|m_j) = \frac{\exp(-\mu^2(T \star s_i,m_j))}{\sqrt{(2\pi)^{\mathfrak{D}}.\,|\Sigma|}}$$

on constate alors que la constante de normalisation $\sqrt{(2\pi)^{\mathfrak{D}}.\,|\Sigma|} = \sqrt{(2\pi)^{\mathfrak{D}}.\sigma^2.\sigma_t^2.\sigma_t^2}$ est constante. On peut donc faire tous nos calculs à cette constante près.

Ainsi, la phase de recherche des appariements dans le cas de l'ICP devra maximiser, pour s_i donné le terme $\exp(-\mu^2(T \star s_i,m_j))$, ou de façon équivalente, minimiser :

$$\mu^2(T \star s_i,m_j) = \frac{d_t^2}{\sigma_t^2} + \frac{d_n^2}{\sigma^2} = \frac{1}{\sigma_t^2}d^2 + (\frac{1}{\sigma^2} - \frac{1}{\sigma_t^2})d_n^2$$

Dans le cas de l'EM, les poids d'appariements seront donnés par :

$$\overline{A_{ij}} = \frac{\exp(-\mu^2(T \star s_i,m_j))}{\sum_k \exp(-\mu^2(T \star s_i,m_k))}$$

deux précisions pratiques sont nécessaires :

- Le lien entre le seuil sur la distance de Mahalanobis μ_{max}^2 utilisée pour le rejet des appariements (voir 3.6.2) et le seuil sur la distance d_{max} utilisée par le kD-Tree pour la recherche des appariements (voir 3.2.3) n'est plus tout à fait évident, puisqu'il n'y a plus qu'un seul écart-type. Ce qui est important, c'est de bien prendre en compte tous les points dont la distance de Mahalanobis ne dépasse pas le seuil. Il faut donc choisir pour d_{max} la distance maximale permettant d'obtenir la distance de Mahalanobis seuil : $d_{max} = \max_{s_i,m_j/\mu^2(T\star s_i,m_j)=\mu^2_{max}}(d(T\star s_i,m_j))$. Dans notre cas, elle dépend de l'écart-type maximum :

$$d_{max} = \sqrt{\mu^2_{max}.\max(\sigma_t^2,\sigma^2)}$$

- Approche multi-échelle : On peut se demander, si on applique, dans l'approche multi-échelle, le même coefficient pour les deux variances σ_t^2 et σ^2. C'est ce que nous faisons.

En ce qui concerne l'estimation de la transformation entre les deux nuages de points, nous devons minimiser le critère suivant (voir 4.3) :

$$C_{ICP}(T,A) = \sum_{ij} A_{ij}.\mu^2(T\star s_i,m_j) = \sum_{ij} A_{ij}.\left(\tfrac{1}{\sigma_t^2}d^2(T\star s_i,m_j) + (\tfrac{1}{\sigma^2}-\tfrac{1}{\sigma_t^2})d_n^2(T\star s_i,m_j)\right) \tag{5.5}$$

Il n'existe malheureusement pas à notre connaissance de solution explicite à la minimisation de ce critère. Nous appliquerons donc une descente de gradient, en adaptant les formules qui donnent $\Phi = \frac{dC(T)}{dT}$ et $H = \frac{d\Phi(T)}{dT}$ (voir 3.8.2), avec $\Sigma^{-1}_{e_{ij}e_{ij}} = \frac{1}{\sigma_t^2}.Id + (\frac{1}{\sigma^2}-\frac{1}{\sigma_t^2}).n_{m_j}n_{m_j}^t$:

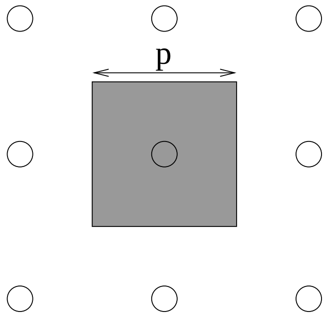
$$\frac{dC(T)}{dT} = \sum_{ij} A_{ij}.\left(\frac{1}{\sigma_t^2}.e_{ij}^t.\left(\frac{de_{ij}}{dT}\right) + \left(\frac{1}{\sigma^2}-\frac{1}{\sigma_t^2}\right).\left(e_{ij}|n_{m_j}\right).\left(\left(\frac{de_{ij}}{dT}\right).n_{m_j}\right)^t\right)$$

$$H \simeq \sum_{ij} A_{ij}.\left(\frac{1}{\sigma_t^2}.\left(\frac{de_{ij}}{dT}\right)^t.\left(\frac{de_{ij}}{dT}\right) + \left(\frac{1}{\sigma^2}-\frac{1}{\sigma_t^2}\right).\left(\left(\frac{de_{ij}}{dT}\right).n_{m_j}\right).\left(\left(\frac{de_{ij}}{dT}\right).n_{m_j}\right)^t\right)$$

5.3.3 Surfaces échantillonnées

Reste à savoir comment fixer nos deux paramètres σ^2 et σ_t^2 pour une surface échantillonnée. Comme nous l'avons déjà mentionné, les distances normales correspondent au véritable bruit de mesure, et le paramètre σ^2 doit donc correspondre à la variance σ_m^2 de celui-ci. En revanche, les erreurs dans le plan tangent peuvent se décomposer comme une erreur due à la mesure et une due au pas d'échantillonnage. Ces deux erreurs sont évidement indépendantes, et on peut donc additionner leurs variances pour fixer le paramètre σ_t^2.

Pour évaluer la variance de l'erreur due au pas d'échantillonnage, il faut calculer la variance de la distance entre un point quelconque de la surface et son plus proche voisin dans notre version échantillonnée de la surface. Les deux points sont ici supposés exacts, puisque l'on traite le bruit de mesure séparément. On peut faire ce calcul lorsque l'échantillonnage est effectué sur une grille à peu près régulière (ce qui est vrai pour les surfaces segmentées dans le scanner, présentées section 1.5.1, et pour les acquisitions 2D 1/2, présentées section 1.5.2), et la surface à peu près lisse. On fait alors une approximation plane de la surface autour d'un point échantillonné, et on regarde les points quelconques qui seront appariés à ce point. Il sont contenus dans un carré de la taille le pas d'échantillonnage :



on peut calculer la variance de la distance des points du carré au point central : elle est égale à $\frac{p^2}{6}$ où p est la pas d'échantillonnage. Finalement, on a :

$$\sigma^2 = \sigma_m^2$$

$$\sigma_t^2 = \sigma_m^2 + \frac{p^2}{6}$$

5.3.4 Discussion

Les expériences, présentées à la section 6.4.3, montrent que l'utilisation de points semi-anisotropes apportent au gain notable dans la robustesse mais parfois une légère diminution de la précision de l'algorithme. De plus, la variance tangentielle doit être plus importante que la variance normale pour apporter de bons résultats, comme le suggère le paragraphe précédent.

En conclusion, on a ainsi pu donner moins d'importance aux distances dans le plan tangent aux surfaces grâce à une méthode s'incorporant directement dans notre canevas statistique, et ainsi amélioré la robustesse du recalage.

Nous reviendrons sur cette méthode dans le chapitre consacré aux surfaces, où nous verrons qu'elle correspond à un cas particulier très simple des modèles probabilistes d'éléments de surface (section 8.3.3).

5.4 Prédiction théorique de l'incertitude pour les surfaces

Nous allons maintenant aborder le problème de la prédiction théorique de l'incertitude du recalage de surfaces. Nous nous appuierons pour cela sur la technique utilisée pour le recalage d'amers géométriques présentée dans la section 3.9, qui a été validée expérimentalement [Pennec, 1996]. Nous avions conclu cette partie en expliquant qu'elle donnait des résultats malheureusement très sous-estimés lorsqu'on cherchait à l'appliquer au recalage de surfaces. La raison de cet échec sont évidentes : toutes les hypothèses étaient vérifiées dans le cas du recalage d'amers géométriques, mais certaines ne le sont plus du tout pour les surfaces échantillonnées. Ces hypothèses problématiques sont :

- "En l'absence de bruit sur les données, le résultat du recalage aurait été exact" : ceci est vrai pour les amers, lorsque les appariements sont connus ou correctement estimés. Mais pour les surfaces, la notion même d'appariement n'a pas grand sens, car les points de la scène et du modèle ne représentent pas les mêmes points physiques du fait de l'échantillonnage. Les appariements sur lesquels s'appuient l'ICP et l'EM sont donc par nature inexacts, et entraînent une erreur de recalage même en l'absence de bruit.
- "Les bruits sur les points sont indépendants" : la véracité de cette hypothèse dépend de l'origine du nuage. Il faut bien comprendre qu'il s'agit du bruit de mesure sur les points, et il ne faut donc pas croire qu'ils sont nécessairement corrélés parce qu'ils proviennent de la même surface. Ainsi, si les points ont été acquis avec un Picza, ou avec la sonde per-opératoire en mode point à point, ils ont de bonnes chances d'être corrompus par un bruit indépendant (aux biais systématiques près). En revanche, les points qui proviennent de la segmentation des scanners sont corrélés, car l'intensité du même voxel peut intervenir dans le calcul de plusieurs points. Enfin, pour les nuages de points per-opératoires pris en mode rafales, ils sont tellement proches qu'on peut retrouver les mêmes erreurs sur deux points voisins (voir section 1.5.3).

Nous chercherons dans la section 5.4.1 une première solution (il faudra attendre la section 8.9 pour une solution plus subtile et mieux justifiée) au problème des appariements et de la corrélation.

Nous verrons dans la section 5.4.2 comment utiliser ces résultats pour guider le chirurgien lors de l'acquisition des points dans le but d'optimiser la précision du recalage.

5.4.1 Appariements semi-anisotropes

Nous allons aborder une solution simple, mais générale et efficace au problème des appariements. Simple parce qu'elle reprend les idées de la prédiction telle que nous l'avons déjà présentée (voir section 3.9) et de l'approximation plane des surfaces qui vient de mener aux modèles de bruits semi-anisotrope (section précédente). Efficace parce qu'elle apporte une amélioration sensible des résultats. Générale parce qu'elle pourra être facilement adaptée aux modèles plus subtils introduits dans la deuxième partie de cette thèse (section 8.9).

Bruit algorithme et bruit réel

On ne veut en-effet pas remettre en cause la méthode de prédiction en elle-même, mais s'attaquer à son talon d'Achille : elle sous-estime l'imprécision parce qu'elle ne prend pas en compte un éventuel biais systématique. Et pourtant, non, en l'absence de bruit sur les données, le résultat de l'ICP classique n'aurait pas été exact, parce que les points sont échantillonnés différemment. Il nous faut donc faire deux choses :

- Prendre aussi en compte le bruit dû à l'échantillonnage dans le bruit sur les données utilisé par la prédiction.
- Appliquer la prédiction à un algorithme de recalage moins biaisé.

Il faut tout d'abord identifier les termes mathématiques qui, dans l'équation 3.10 que nous rappelons ici, correspondent à ces deux points :

$$\Sigma_{TT} = H^{(-1)}.\left(\frac{\partial \Phi}{\partial \chi}\right).\Sigma_{\chi\chi}.\left(\frac{\partial \Phi}{\partial \chi}\right)^t.H^{(-1)}$$

- Le modèle de bruit "réel" sur les données utilisé par la prédiction s'exprime au travers du terme $\Sigma_{\chi\chi}$.
- Le modèle de bruit qui est pris en compte dans le choix et le paramétrage de l'algorithme de recalage en lui-même intervient ici par l'intermédiaire des dérivés du critère H et $\frac{\partial \Phi}{\partial \chi}$.

On peut ainsi prédire théoriquement la précision d'un algorithme reposant sur tels hypothèses (par exemple les plus simples : les points se correspondent physiquement et le bruit de mesure est gaussien isotrope pour l'ICP classique) pour un bruit qui respecte d'autres hypothèses (un bruit plus représentatif des surfaces par exemple).

Approximation plane de la surface

Nous illustrons ces idées en utilisant ici une approximation plane de la surface, i.e. un modèle de bruit semi-anisotrope, avec l'ICP classique (la généralisation à l'ICP/EM ou à tout autre algorithme est évidente). Il faut donc faire le distinguo entre la matrice de covariance utilisée par l'ICP $\Sigma_{e_{ij}e_{ij}}$ et la matrice de covariance réelle du bruit $\Sigma_{\chi\chi}$ (les points sont supposés indépendants de covariance $\Sigma_{r_{ij}r_{ij}}$) dans les eq. 3.11 et 3.12 :

$$H = \sum_{ij} A_{ij}.\left(\frac{\partial e_{ij}}{\partial T}\right)^t.\Sigma_{e_{ij}e_{ij}}^{-1}.\left(\frac{\partial e_{ij}}{\partial T}\right)$$

$$\frac{\partial \Phi}{\partial \chi} = \sum_{ij} A_{ij}.\left(\frac{\partial e_{ij}}{\partial T}\right)^t.\Sigma_{e_{ij}e_{ij}}^{-1}.\left(\frac{\partial e_{ij}}{\partial \chi}\right)$$

$$G = \sum_{ij} A_{ij}^2.\left(\frac{\partial e_{ij}}{\partial T}\right)^t.\Sigma_{e_{ij}e_{ij}}^{-1}.\Sigma_{r_{ij}r_{ij}}.\Sigma_{e_{ij}e_{ij}}^{-1}.\left(\frac{\partial e_{ij}}{\partial T}\right)$$

$$\Sigma_{TT} = H^{(-1)}.\left(\sum_{ij} A_{ij}^2.\left(\frac{\partial e_{ij}}{\partial T}\right)^t.\Sigma_{e_{ij}e_{ij}}^{-1}.\Sigma_{r_{ij}r_{ij}}.\Sigma_{e_{ij}e_{ij}}^{-1}.\left(\frac{\partial e_{ij}}{\partial T}\right)\right).H^{(-1)} \quad (5.6)$$

On pourra donc jouer sur ces deux covariances différentes :

– Dans la covariance de l'ICP $\Sigma_{e_{ij}e_{ij}}$ (qui devra être la même que celle effectivement utilisée dans l'algorithme de recalage lui-même), on pourra faire tendre la composante tangente σ_t^2 vers l'infini, pour obtenir un algorithme moins biaisé.
– Dans la covariance "réelle" $\Sigma_{r_{ij}r_{ij}}$, augmenter la composante tangente σ_t^2 pour prendre en compte le bruit dû à l'échantillonnage.

Cas de l'EM

L'adaptation de la formule à l'ICP/EM est immédiate (voir 4.5.3). Les termes supplémentaires sont en général négligeables, et n'apportent donc pas de changements significatifs dans l'incertitude prédite. On peut alors se demander si l'utilisation de l'EM peut présenter un quelconque intérêt ici. La réponse est oui : si la certitude prédite est la même, elle est plus proche de la réalité pour l'ICP/EM, car la répétabilité de celui-ci, non-prise en compte ici, est bien meilleure.

Problème de la corrélation entre points

Le cas où les points sont corrélés entre eux est problématique. Il faudrait, pour le résoudre élégamment, prendre en compte la corrélation dans la matrice de covariance $\Sigma_{\chi\chi}$ qui représente le bruit réel sur les données. Ceci n'est pas facile pour deux raisons :

– La matrice $\Sigma_{\chi\chi}$ ne sera plus diagonale par bloc. Aux termes diagonaux - qui représentent la covariance de chaque point - vont s'ajouter des termes non diagonaux, pour exprimer les covariances croisées de chaque paire de points. Le calcul de G sera considérablement complexifié.
– Le calcul de ces termes non diagonaux devra faire appel à un modèle de la corrélation du bruit entre les points. Il est difficile de voir quel modèle utiliser : faut-il modéliser explicitement la méthode d'acquisition des données ? Ou peut-on déduire un tel

modèle d'une étude empirique des données ? Ou simplement proposer un modèle sans justification, et l'ajuster jusqu'à ce que la prédiction soit correcte ?

Ayant pris conscience de la difficulté de la tâche, nous avons préféré contourner le problème, lorsque seuls les points proches sont corrélés (ce qui est le cas pour le scanner comme pour les pointés per-opératoires).

La décimation que nous appliquons dans l'ICP/EM multi-échelle va en-effet fusionner les points proches, et gommer ainsi les corrélations. Ceci permet à la fois d'améliorer la précision du recalage elle-même (voir résultats expérimentaux figure 6.9), et sa prédiction théorique. Mais le réglage du facteur α de la décimation a alors son importance : s'il est trop faible, on ne va pas regrouper tous les points corrélés (et donc sous-estimer l'incertitude). S'il est trop grand, on va regrouper des points non corrélés, perdre ainsi de l'information, et donc rendre l'algorithme moins précis (la prédiction fonctionnera certes correctement, mais à quoi bon?).

Résultats synthétiques

Nous avons dans un premier temps fait des expériences sur données synthétiques : nous avons pris la surface d'un ellipsoïde à peu près de la même taille que nos mâchoires (les grand axes font respectivement $5cm$, $3cm$ et $1cm$). Pour obtenir le modèle, nous l'avons échantillonnée régulièrement avec un pas de $0.25mm$. Pour la scène, nous avons échantillonné aléatoirement une centaine de points auxquels nous avons rajouté un bruit gaussien isotrope d'écart-type $0.3\,mm$ non-corrélé. L'immense avantage de cette approche synthétique est que l'on connaît parfaitement la bonne transformation (ici l'identité) et le modèle de bruit (ici non corrélé). De plus, on peut créer autant de scènes que nécessaire.

Nous avons mesuré la covariance empirique de la transformation par “bootstrap” (voir 3.9.4) : nous avons recalé, sur le même modèle, un millier de scènes, chacune échantillonnée et bruitée différemment, puis étudié la covariance des transformations obtenues autour de l'identité.

Nous avons ensuite appliqué la prédiction sur quelques une de ces scènes, pour vérifier sa stabilité (et, de fait, elle ne varie pas de plus de 5%), et la comparer la covariance prédite à la covariance empirique.

Dans le cas de l'ICP isotrope avec prédiction classique, nous avons obtenu les covariances suivants (les chiffres ont été multipliés par 10^6 pour plus de lisibilité) :

$$\text{Cov. predite} = \begin{pmatrix} 12 & 0 & 0 & 0 & -4 & -0 \\ 0 & 5 & 0 & 2 & 0 & 2 \\ 0 & 0 & 4 & 0 & -1 & 0 \\ 0 & 2 & 0 & 905 & 0 & 0 \\ -4 & 0 & -1 & 0 & 906 & 0 \\ 0 & 2 & 0 & 0 & 0 & 905 \end{pmatrix} \quad \text{Cov. mesuree} = \begin{pmatrix} 32 & -1 & -2 & 46 & 10 & 5 \\ -1 & 10 & 1 & 11 & -20 & -4 \\ -2 & 1 & 22 & 4 & -19 & -11 \\ 46 & 11 & 4 & 6935 & 497 & -308 \\ 10 & -20 & -19 & 497 & 4359 & 87 \\ 5 & -4 & -11 & -308 & 87 & 1342 \end{pmatrix}$$

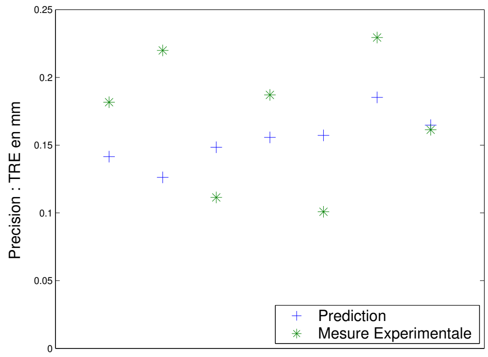
Comme on le voit, la covariance prédite est doublement mauvaise : d'une part elle n'a pas la bonne échelle (elle est bien trop faible), d'autre part elle n'a pas la bonne forme (les erreurs sur la translation, données par les termes inférieurs droits, apparaissent isotropes non corrélés, contrairement à ce que l'on mesure empiriquement).

Voyons maintenant ce qu'apportent nos améliorations. Appliquons donc l'EM sur les mêmes données avec les points semi-anisotropes (on utilise la même covariance pour le paramétrage de l'algorithme et pour le bruit réel de la prédiction : l'écart-type dans l'axe de la normale correspond à celui du bruit de mesure, $0.3mm$, et est doublé dans le plan tangent). La décimation est ici inutilisée, car les points ne sont pas corrélés. On obtient les covariances suivantes sur la transformation [12] :

$$\text{Cov. predite} = \begin{pmatrix} 33 & 1 & 9 & -17 & -51 & 8 \\ 1 & 9 & 0 & -1 & 16 & 5 \\ 9 & 0 & 88 & 17 & -57 & -18 \\ -17 & -1 & 17 & 11032 & -1360 & 646 \\ -51 & 16 & -57 & -1360 & 6523 & -216 \\ 8 & 5 & -18 & 646 & -216 & 1332 \end{pmatrix} \quad \text{Cov. mesuree} = \begin{pmatrix} 41 & -1 & 0 & -19 & -73 & 30 \\ -1 & 9 & 2 & 37 & -8 & 12 \\ 0 & 2 & 93 & 446 & -94 & -19 \\ -19 & 37 & 446 & 11454 & -518 & -423 \\ -73 & -8 & -94 & -518 & 6828 & 39 \\ 30 & 12 & -19 & -423 & 39 & 1290 \end{pmatrix}$$

On constate alors que les deux covariances se correspondent à peu près (en tout cas les termes diagonaux qui sont les plus importants) : l'échelle et la forme sont maintenant correctement prédites.

12. On pourra remarquer, en comparant les covariances mesurées, que l'EM semi-anisotrope semble expérimentalement moins précis que l'ICP classique sur cet exemple. Ce manque de précision ne provient pas de l'utilisation de l'EM (en général un peu plus précis que l'ICP), mais de l'utilisation d'un écart-type plus important dans le plan tangent (cette légère perte de précision avec l'utilisation des points semi-anisotropes est aussi visible sur la base de test, figure 6.6).



matlab/PredExpDents not found\protect !

FIG. 5.1 – *Comparaison de la précision prédite et mesurée pour un recalage pointés sur l'os / scanner (à gauche) pointés sur les dents / scanner (à droite) sur 7 cas de la base de tests*

Résultats réels

Nous présentons ici les résultats de la prédiction sur les recalages pointés sur l'os / scanner. Nous avons pris le réglage des paramètres de recalage qui permettait d'obtenir la meilleure précision (voir section 6.4.5), ici un ICP/EM avec un modèle de bruit isotrope d'écart-type $0.3mm$, et un facteur de décimation $\alpha = 1$ (soit un rayon pour les sphères de décimation de $0.3mm$).

Nous avons fait fonctionné la prédiction de la précision avec un modèle de bruit réel semi-anisotrope, d'écart-type $0.3mm$ dans l'axe de la normale et deux fois plus grand dans le plan tangent. Nous avons alors comparé la précision mesurée expérimentalement et la prédiction prédite (voir figure 5.1 gauche). Il semble que l'échelle soit à peu près bonne, mais la prédiction de la précision n'est pas très... précise.

De plus, si on applique maintenant la même prédiction au recalage pointés sur les dents / surface, les résultats sont maintenant très sur-estimés (de l'ordre de $0.6mm$), et il faut diminuer le facteur de décimation α (en fait prendre un facteur quasi-nulle, car les données sont indépendantes, voir les résultats figure 5.1 droite) pour obtenir une échelle correspondant aux résultats expérimentaux (de l'ordre de $0.2mm$).

Conclusion

Il nous semble qu'avec cette approche la forme de la covariance (qui indique les directions où il y a le plus d'imprécision) soit maintenant relativement bien estimée pour le recalage de surfaces (voir les expériences sur des données synthétiques). En revanche, nous n'avons pas vraiment résolu le problème d'échelle. La dernière expérience présentée montre que l'on peut obtenir une échelle à peu près correcte pour une catégorie de données précise en jouant sur le réglage des paramètres[13]. Mais les résultats sont encore assez

13. On pourra par exemple ajuster l'écart-type des bruits, qui interviennent directement dans la prédiction. On pourra aussi jouer sur le facteur de décimation α qui contrôle le nombre de points n. Notons

mauvais (ils varient du simple au double), et il reste de toute façon difficile d'affirmer quoi que ce soit étant donné le faible nombre d'expériences.

5.4.2 Guidage de l'acquisition des points per-opératoires

Le calcul théorique de la précision donne hélas des résultats sous-évalués. Mais s'ils ne sont pas utilisables pour leur valeur absolue, on peut peut-être utiliser leurs valeurs relatives. C'est l'idée fondamentale du guidage de l'acquisition : comparer le gain de précision théorique apporté par des points disposés à différents endroits de la mâchoire. On peut alors conseiller le chirurgien lorsqu'il acquiert de nouveaux points sur la mâchoire, en lui indiquant à quel endroit il faut prendre des points pour obtenir des gains significatifs dans la précision du recalage. Ceci pourrait rappeler le travail de [Simon, 1996, ch. 3], qui cherche aussi a déterminer la meilleure façon de répartir les points. Mais il le fait avant, et non après le recalage, et est ainsi obligé d'utiliser une méthode bien plus lourde que la nôtre.

Principe

Nous voulons donc profiter du fait que nous travaillons après un premier recalage : nous avons déjà recalé une première acquisition non-guidée[14] de la scène (les points per-opératoires) sur notre modèle (le scanner ou le Picza), et effectué le calcul de H et G qui permet la prédiction théorique de l'incertitude Σ_{TT} sur la transformation. On peut alors facilement voir comment évolue la prédiction lorsqu'on ajoute un nouveau point s' à la scène, en rajoutant sa contribution à H et G, et en en déduisant la nouvelle incertitude Σ'_{TT} sur la transformation. Si on effectue maintenant le même calcul pour plusieurs points répartis sur la mâchoire, on obtient de nombreuses nouvelles incertitudes prédites, qui présentent toutes le même problème d'échelle : leur valeurs absolues ne sont toujours pas utilisables, mais on peut au moins les comparer entre elles, et ainsi voir quel point apporte le meilleur gain en précision.

En pratique, on ne compare pas directement les matrices de covariances Σ'_{TT}, mais on les utilise pour en déduire la TRE (voir section 6.1.6), i.e. l'écart-type moyen σ' prédit sur le positionnement des points dans une zone cible (comme par exemple la totalité de la mâchoire ou la zone où l'on va opérer) :

$$\sigma' = Tr(K_{xx}.\Sigma'_{TT}) \quad \text{avec} \quad K_{xx} = E\left[\frac{d(T\star x)}{dT}^t . \frac{d(T\star x)}{dT}\right]$$

que l'écart-type de la précision prédite est grossièrement proportionnel à $\sigma/\sqrt{n}$.

14. En tout cas pas par le système que nous présentons ici. Dans VirtualScope, il y a cependant un premier guidage, basé sur des heuristiques, comme "il faut répartir les points sur les zones fortement courbées".

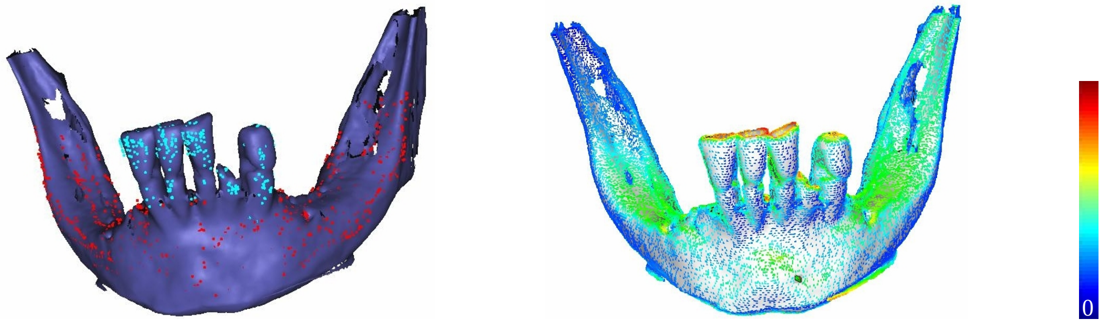


FIG. 5.2 – *Guidage de l'acquisition pour un recalage Pointés sur les Dents / scanner (la TRE est calculée sur la zone cible, située au niveau de l'incisive cassée)*
A gauche : la répartition des points per-opératoires (en bleu clair) conditionne mal le recalage dans l'axe y (gauche/droite) et très mal dans l'axe z (haut/bas)
A droite : le guidage de l'acquisition conseille de prendre maintenant des points en priorité au dessus (en rouge, jaune et orange, pour améliorer la précision en z) puis sur le côté (en vert et bleu clair)... et pas la peine d'en rajouter sur le devant (en bleu)

où x décrit les points de la zone cible.

Enfin, il faut savoir comment répartir les points potentiels s' pour lesquels on va calculer la nouvelle précision. Il est bien évident que, pour qu'un nouveau point est un intérêt, il ne faut pas qu'il soit considéré comme un point aberrant dans le recalage (auquel cas le résultat ne change pas), i.e. qu'il soit situé à proximité d'un point du modèle. Aussi, nous allons tout simplement utiliser ces points du modèles, comme nouveaux points à essayer.

L'algorithme est donc le suivant :

Recalage initial : Recaler la scène sur le modèle, calculer H et G (voir section précédente), calculer $K_{xx} = E\left[\frac{d(T\star x)}{dT}^t . \frac{d(T\star x)}{dT}\right]$ où x décrit les points de la zone cible.

Boucle : Pour chaque point du modèle m_j :

- Calculer le point de la scène qui lui correspond potentiellement : $s' = T^{-1} \star m_j$
- Calculer $H' = H + H(s')$ et $G' = G + G(s')$, et en déduire la nouvelle précision prédite sur la transformation Σ'_{TT}
- En déduire la TRE sur la cible composée des points x : $\sigma' = Tr(K_{xx}.\Sigma'_{TT})$

Reste à savoir comment présenter les résultats. On le fait graphiquement, en affichant le modèle en colorant les points en fonction de la TRE prédite (voir figure 5.2). L'utilisateur n'a plus qu'à viser les zones de la bonne couleur.

Discussion

Les résultats présentés sont très représentatifs de ce que l'on a pu observer sur les quelques expériences jusqu'ici réalisées : le guidage est proche de l'intuition, puisqu'il nous dit grosso-modo qu'il faut prendre des points sur des zones de la surface où la normale pointe vers une direction jusqu'ici délaissée. Il semble aussi défavoriser les points éloignés de la zone cible.

Pourtant nous n'avons pas véritablement cherché à valider expérimentalement cette technique. Pour y arriver, il faudrait probablement appliquer la stratégie suivante : il s'agirait d'enlever une partie des points dans un des nuage de pointés per-opératoires, évaluer expérimentalement la précision du recalage avec ce nuage amputé, puis rajouter des points dans une zone donnée, mesurer à nouveau la précision et en déduire le gain en précision. On pourrait alors comparer les gains expérimentaux et théoriques associés à différentes zones, et vérifier qu'il sont grossièrement proportionnels.

5.4.3 Conclusion

Nous n'avons pas réussi une adaptation pleinement satisfaisante des techniques de prédiction théorique de l'incertitude. Les rares expériences montrent que nos quelques améliorations (l'utilisation des points semi-anisotropes pour prendre en compte l'échantillonnage, l'utilisation de la décimation pour éliminer les corrélations) ont permis d'obtenir de bons résultats, mais il subsiste un problème d'échelle qu'il faut régler au cas par cas à l'aide des précisions mesurées expérimentalement au chapitre suivant. Nous avons donc mis au point un système de guidage de l'acquisition qui, en utilisant les précisions *relatives* apportées par différents points, s'abstrait de ce problème d'échelle et semble pouvoir donner des conseils pertinents au praticien. En ce qui concerne la précision *absolue* (i.e. la résolution du problème d'échelle) nous avons tout de même isolé les deux problèmes clés :

- La modélisation des appariements pour le recalage de surfaces, auquel nous avons apporté une première réponse, qui sera complétée dans la section 8.9.
- La modélisation du bruit sur les données : celle-ci doit être fidèle pour que la prédiction donne de bons résultats. Ceci signifie d'une part que la qualité de la prédiction est directement reliée à la qualité de l'estimation de la variance du bruit (voir section 6.4.2), et d'autre part qu'il faut prendre en compte la corrélation entre les différents points lorsqu'elle existe (c'est le cas sur les scanners et les rafales de pointés per-opératoires).

La résolution de ce dernier point est incontournable. En-effet, la précision du recalage dépend de manière évidente du bruit présent sur les données, et même si une méthode révolutionnaire de prédiction de l'incertitude était mise au point (ce qui est peu probable, tant celle que nous avons utilisé semble bien aborder le problème) elle devrait nécessairement se baser sur un modèle de bruit.

5.5 Conclusion

D'un point de vue théorique, nos trois variantes de l'ICP/EM montrent la généralité, la puissance et la souplesse du cadre statistique présenté au chapitre précédent : il a suffi de se demander comment exprimer l'idée initiale (utiliser la distance entre normales, prendre en compte un décalage, donner moins d'importance aux distances dans le plan tangent) au travers de la loi de probabilité $p(T \star s_i | m_j)$ pour pouvoir l'intégrer à l'ICP ou l'ICP/EM, et profiter ainsi de leur propriétés (et notamment la convergence assurée).

D'un point de vue pratique, ces trois techniques ont apporté encore une amélioration notable de la robustesse, et une légère amélioration de l'incertitude. Couplées à l'ICP/EM multi-échelle, elles ont donné un algorithme répondant au cahier des charges de Virtual-Scope.

En revanche, nous n'avons pas réussi une adaptation pleinement satisfaisante des techniques de prédiction théorique de l'incertitude. Quelques améliorations ont permis d'obtenir des résultats corrects (moins de 50% d'erreurs), mais il subsiste un problème d'échelle. Nous avons donc mis au point un système de guidage de l'acquisition qui, en utilisant les précisions *relatives* apportés par différents points, s'abstrait de ce problème d'échelle et semble pouvoir donner des conseils pertinents au praticien. En ce qui concerne la précision *absolue* (i.e. la résolution du problème d'échelle) nous avons tout de même isolé les causes de nos tracas et indiqué la marche à suivre inévitablement pour y remédier, d'un point de vue pratique en ajustant les paramètres en fonction des résultats expérimentaux, d'un point de vue théorique en modélisant mieux les appariements et les corrélations.

Chapitre 6

Évaluation, réglage et validation des algorithmes de recalage rigide

Sommaire

6.1 Introduction

6.1.1 Problématique

Notre quête de l'algorithme de recalage idéal aura été guidée par un seul critère : les performances mesurées expérimentalement sur les données AREALL. L'étude et l'optimisation de ces performances se devait d'être particulièrement poussée dans la mesure où nos algorithmes sont utilisés dans le cadre d'une application clinique. Nous présentons donc ici l'essentiel des techniques utilisées à cet effet.

Précisons que la plupart des idées développées dans ce chapitre sont en fait valables pour une large catégorie d'algorithmes de recalages, ceux qui fonctionnent par minimisation itérative d'un critère non convexe (nous suggérons au lecteur de relire la section 3.2.2 pour se familiariser avec leur problématique). Cette famille d'algorithme, dont l'ICP et l'ICP/EM sont deux cas particuliers, inclut aussi par exemple les algorithmes de recalage d'intensité par minimisation de l'information mutuelle [Roche *et al.*, 2000; Roche, 2001], et beaucoup d'autres.

6.1.2 État de l'art

Il est difficile de réaliser un état de l'art dans le domaine de la validation des algorithmes de recalages. En-effet, la plupart des articles présentant de tels algorithmes contiennent une section expérimentale, et on y trouve de nombreuses techniques d'évaluation des performances des algorithmes. Mais force est d'avouer que la plupart sont des techniques ad-hoc (elles ne sont pas véritablement justifiées et ne sont pas basées sur des travaux d'autres chercheurs) qui ont pour seul but de comparer plusieurs algorithmes. On y trouve d'ailleurs de nombreuses erreurs de raisonnement, par exemple :

- certains présentent la variation moyenne des résultats pour le même recalage et les mêmes données comme une mesure de la précision du recalage. Il s'agit en-fait d'une mesure de la répétabilité (voir section 6.3.1).
- certains donnent la valeur du critère[15] (i.e. l'écart-type total des distances entre points appariés) comme une mesure de la précision du recalage. Mais celle-ci fait aussi intervenir le bruit sur les données et dépend des erreurs d'appariements (voir section 3.2.1). Sa valeur est rarement représentative de la précision [Fitzpatrick *et al.*, 1999, Introduction].
- certains utilisent directement la partie translation de la transformation rigide trouvée pour estimer la précision du recalage. Hors celle-ci dépend de la façon dont est exprimée la transformation, et est couplée à la rotation (voir section 6.1.6).

15. Souvent appelée "Fiducial Registration Error" dans les articles.

A vrai dire, le problème d'une harmonisation rigoureuse des techniques de validation des algorithmes de recalage n'intéresse la communauté scientifique que depuis l'année 2000 [16] [Bowyer *et al.*, 2001; Jannin *et al.*, 2002; Woods, 2000], et n'a pas encore véritablement débouché.

Ce chapitre tente donc de justifier et compléter les techniques qui nous ont paru les plus pertinentes.

6.1.3 Organisation du chapitre

Pour utiliser un algorithme de recalage de façon sûre, il nous faut :

- Savoir évaluer le manque de robustesse de l'algorithme, i.e. savoir dans quelle mesure les défauts dans les données, les paramètres ou les estimées initiales des transformations peuvent mener à un résultat faux. C'est ce que nous verrons dans la section 6.2.
- Mettre en place des systèmes de sécurité pour éviter ou rejeter les résultats faux lorsque la parfaite robustesse de l'algorithme ne peut être garantie. Ce sera le sujet de la section 6.2.3.
- Savoir évaluer la précision de l'algorithme, une fois les problèmes de robustesse résolus. Nous verrons cela dans la section 6.3.
- Savoir comment régler les paramètres en fonction des données. Cette problématique est traitée dans la section 6.4.

Avant de rentrer dans le détail, nous devons faire quelques remarques générales. La section 6.1.4 oppose l'approche expérimentale présentée ici à la prédiction théorique. La section 6.1.5 présente la base de test AREALL, i.e. l'ensemble des cas sur lesquels nous avons effectué la plupart des expériences. La section 6.1.6 explique comment exprimer mathématiquement les erreurs entre plusieurs transformations.

6.1.4 Mesure expérimentale vs prédiction théorique

On peut envisager deux approches pour toutes nos problématiques : l'approche expérimentale et l'approche théorique.

Avec l'approche théorique (présentée pour l'incertitude aux chapitres précédents, voir sections 3.9, 4.5.3 et 5.4), nous sommes obligé, pour chaque type d'algorithme, de faire des calculs complexes, incluant souvent des simplifications et des hypothèses abusives. Les résultats sont quantitativement faux sur les surfaces, et donc inutilisables pour valider

16. Exception faite de la base Vanderbilt [West *et al.*, 1997], qui dès 1996 offrait la possibilité de tester la précision des algorithmes de recalage iconiques rigides multi-modaux sur 9 patients différents. La méthode était basée sur un étalon or obtenu par recalage de marqueurs fixés sur la tête du patient et visibles dans les images. Les résultats sont exprimés sous forme d'erreur sur des points cibles (TRE).

l'algorithme. En revanche, les équations obtenues sont très générales (elles prennent par exemple en compte les valeurs de certains paramètres), et permettent de prévoir rapidement (et non de mesurer à-posteriori) qualitativement les performances d'un algorithme, et donc de mieux comprendre son fonctionnement et de mieux pouvoir le régler.

Avec l'approche expérimentale (présentée dans ce chapitre), nous pouvons mettre en place des méthodes indépendantes du type d'algorithme (et a fortiori de son réglage). Nous ne pouvons en revanche pas faire d'étude générale, mais devons refaire tourner ces méthodes pour chaque type d'application (i.e. chaque type de protocole opératoire, mais aussi chaque type de morphologie, chaque type d'artefact, chaque type d'algorithme, chaque réglage de paramètres, ...) et devons parfois acquérir des données spécifiques. Certaines expériences impliquent de très nombreux recalages (plusieurs centaines voire plusieurs milliers), et sont donc très coûteuses en terme de temps de calcul. Nous obtenons en revanche des résultats quantitativement justes.

On comprend donc que, même si nous devons mettre l'accent sur l'approche expérimentale pour la validation (et forcément pour vérifier les prédictions théoriques), les deux approches sont très complémentaires, et aucune ne peut être ignorée.

Précisons enfin que les méthodes expérimentales de test de la robustesse et de la précision devront être intégrées dans un banc de test qui sera automatisé au maximum, car elles seront utilisées sur de nombreux cas différents avec de nombreux paramétrages différents, et les résultats devront pouvoir être facilement comparés, pour déterminer le paramétrage optimum.

6.1.5 Base de test AREALL

Pour faire des expériences, il faut des données ! Sauf exception, les résultats expérimentaux présentés dans cette thèse sont basés sur un jeu de données appelé Base de Test. Cette base inclut 9 cas différents, les cas qui nous semblaient les plus représentatifs des futures conditions d'utilisation de VirtualScope. Pour chaque cas, nous disposons de 3 scanners (de résolution $0.25 \times 0.25 \times 0.5mm$) segmentés (les surfaces contiennent plusieurs centaines de milliers de points), et de deux nuages de points per-opératoires acquis dans la même position, l'un contenant quelques centaines de points répartis sur les dents (entre 300 et 1100), l'autre contenant 1500 points répartis sur l'os de la mâchoire. On peut voir sur la figure 6.1 ces différents cas, avec, pour chacun, la surface segmentée du second scanner et les deux nuages de points per-opératoires dans leur position de référence.

Ces cas sont effectivement représentatifs des deux grosses variabilités observées d'un patient à l'autre :

- La répartition des dents : le nombre de dents, mais aussi leur répartition est importante pour le recalage. En-effet, si les dents restantes sont proches, les points per-opératoires pris sur ces dents vont être presque alignés, et le recalage sera alors

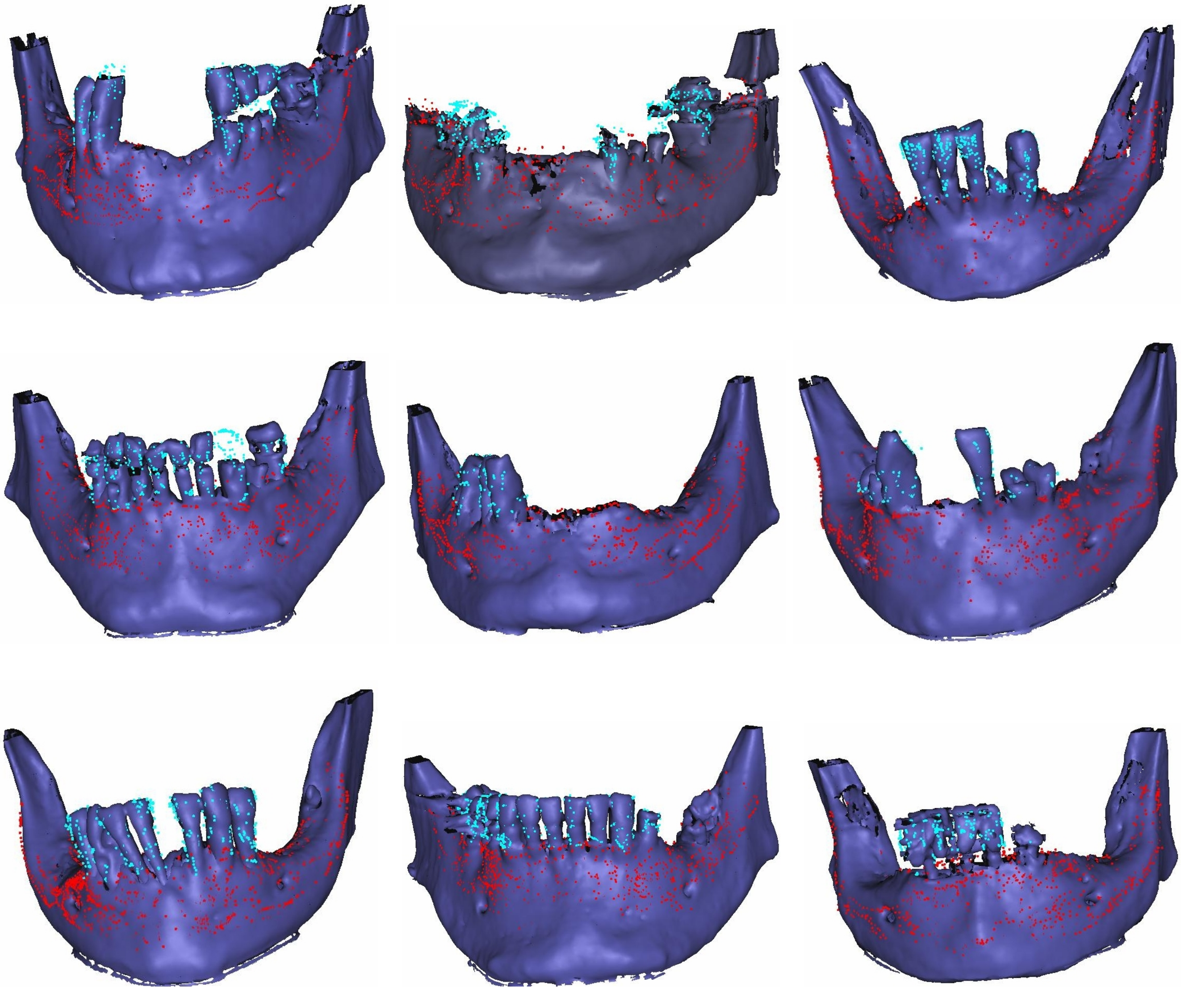


FIG. 6.1 – *La base de test AREALL : pour chaque cas, la surface segmentée de l'os dans le scanner, les nuage de points per-opératoire sur les dents et sur l'os dans leur position de référence.*

mal conditionné (cas du milieu). Ceci sera d'autant plus gênant lorsque le site à opérer est éloigné.

- Le niveau d'artefact : les artefacts, qui sont surtout visibles au niveau des dents, vont corrompre la surface segmentée et donc le recalage. Dans les situations les plus graves, des dents disparaissent en partie (cas milieu-gauche) ou en totalité (cas haut-milieu), ou sont complètement déformées (cas bas-droite).

Le recalage pointés sur les dents / scanner s'est d'ailleurs révélé impossible (ou beaucoup moins précis) sur les deux cas les plus pathologiques (haut-milieu et bas-droite), et nous ne les avons donc pas inclus dans les résultats d'expériences, sauf dans la section sur les systèmes de sécurités (voir 6.2.3) qui doivent bien-sûr être capables de détecter ces les échecs du recalage sur ces cas problématiques.

Pour effectuer les recalages, nous avons presque toujours utilisé un ICP/EM multi-échelle avec un bruit gaussien isotrope homogène (tel que présenté dans la section 4.4.1) en utilisant les normales (voir section 5.2.2) et les points semi-anisotropes gaussiens (voir section 5.3.2) mais sans utiliser la correction de décalage normal (voir section 5.1).

6.1.6 Mesure de l'erreur sur une transformation rigide

Nous devrons par la suite quantifier l'erreur entre une transformation rigide T_r obtenue à l'issu d'un recalage et une transformation de référence T_0, i.e. quantifier l'amplitude de la transformation résiduelle $T_d = T_r \star T_0^{-1}$.

Précisons que lorsqu'on aura plusieurs mesures d'erreurs pour un même type de recalage, on pourra les synthétiser en gardant :

- l'erreur maximale : celle-ci est plus parlante pour une validation clinique.
- la covariance des erreurs : celle-ci est plus commode et plus puissante d'un point de vue statistique. Elle permet en-effet d'examiner dans quelles directions se retrouve principalement l'erreur. Elle permet de plus de propager les erreurs (voir par exemple le calcul de l'erreur sur des points cibles à partir de l'erreur sur la transformation ci-après). Elle permet enfin de composer les erreurs (lorsqu'on compose les transformations de plusieurs recalages consécutifs par exemple : s'ils sont indépendant, leurs covariances s'additionnent).
- la variance des erreurs : on peut la déduire de la covariance ou la calculer directement. On perd alors l'information sur l'orientation de l'erreur, et on ne peut plus faire de calcul de propagation. La composition des erreurs reste en revanche valable.

Reste maintenant à savoir comment mesurer l'erreur entre deux transformations rigides. On utilisera par la suite deux mesures différentes, que nous présentons ici.

Erreur sur la transformation (TE) On pourra utiliser directement *l'erreur sur la transformation* (on utilisera la dénomination anglaise : "Transformation Error", TE en abrégé) : celle-ci est composée d'une erreur sur la rotation et d'une erreur sur la translation.

L'erreur sur la rotation permettra de quantifier directement les erreurs sur l'orientation (comme celle de l'outil chirurgical). L'erreur sur la translation sera en revanche difficilement utilisable, d'une part parce qu'elle est couplée à l'erreur sur la rotation, d'autre part parce qu'elle dépend de la façon dont est exprimée la transformation (et en particulier l'origine du repère qui a été choisi, lorsqu'on utilise la forme $T \star x = Rx + t$).

En termes mathématiques, il nous faudra exprimer la transformation résiduelle (la transformation $T_d = T_r \star T_0^{-1}$ qui sépare la transformation issue du recalage T_r et la transformation de référence T_0) en utilisant son vecteur (r,t) [17]. L'erreur sur la rotation est l'angle de la rotation différentielle, i.e. la norme de r. Lorsqu'on disposera de plusieurs transformations résiduelles, on pourra exprimer leur covariance comme la covariance des vecteurs les représentant (voir section D.2.2.4) :

$$\Sigma_{TT} = E\left[(r,t).(r,t)^t\right]$$

la variance de la transformation n'aurait aucun sens, puisqu'elle mélange rotation (en degré) et translation (en mm). On pourra en revanche calculer la variance de la rotation par la formule suivante :

$$\sigma_r^2 = E\left[\|r\|^2\right]$$

ou lorsqu'on dispose déjà de la covariance, en prenant les trois premiers termes diagonaux :

$$\sigma_r^2 = (\Sigma_{TT})_{11} + (\Sigma_{TT})_{22} + (\Sigma_{TT})_{33}$$

Erreur sur des points cibles (TRE) On pourra aussi calculer la conséquence de *l'erreur* de transformation *sur la position de certains points cibles* (on utilisera la dénomination anglaise : “Target Registration Error”, TRE en abrégé). Ces points cibles pourront être répartis sur la totalité des surfaces recalées ou être concentrés autour de la zone opératoire. On pourra alors directement quantifier les erreurs sur le positionnement d'un point dans cette zone (comme l'extrémité de l'outil chirurgical).

Concrètement, on peut calculer la TRE de deux façons. La première est directe : on mesure l'erreur entre la position d'un point cible auquel on applique la transformation de référence et le même point cible auquel on applique la transformation issue du recalage. Lorsqu'on a plusieurs points cibles ou plusieurs transformations, on peut ainsi calculer l'erreur max et la covariance ou la variance des erreurs.

La seconde s'appuie sur un calcul de propagation des erreurs : lorsqu'on a calculé la

17. r est le vecteur représentant la rotation (colinéaire à l'axe de rotation et de norme l'angle de cette rotation) et t est le vecteur de la translation. On utilise (r,t) car il représente la transformation rigide dans la carte principale centrée en T_0 (voir annexe D.1.4), et la covariance de la transformation est justement définie comme la covariance de cette représentation vectorielle (voir annexe D.2.2.4).

covariance Σ_{TT} sur la transformation comme indiqué dans le paragraphe précédent, on peut en déduire la covariance sur les points par la formule suivante (voir ??) :

$$\Sigma_{xx} = E\left[\frac{d(T \star x)}{dT} . \Sigma_{TT} . \frac{d(T \star x)}{dT}^t\right]$$

ou la variance sur les points par la formule suivante (voir ??) :

$$\sigma_x^2 = E\left[Tr(\Sigma_{xx})\right] = Tr(K_{xx}.\Sigma_{TT}) \quad \text{avec} \quad K_{xx} = E\left[\frac{d(T \star x)}{dT}^t . \frac{d(T \star x)}{dT}\right]$$

on ne pourra en revanche pas déduire ainsi une erreur maximum sur la position des points.

6.2 Robustesse

Nous allons étudier dans cette partie la robustesse des algorithmes, définie ici comme leur aptitude à converger à proximité de la bonne transformation.

Les algorithmes par minimisation de critère convergent toujours vers un minimum local du critère, mais celui-ci n'a aucune raison de se trouver à proximité du minimum global. En fait, la convergence dépend essentiellement de la qualité de l'estimée initiale de la transformation. Si elle est suffisamment proche de la bonne transformation, et en particulier si elle est située dans la cuvette du minimum global, la convergence à proximité du minimum global est assurée. Avec certaines variantes robustifiées de l'algorithme (en particulier avec la version multi-échelle, dans laquelle on modifie le critère en modifiant l'échelle) on arrivera même à converger à partir d'endroits situés à l'extérieur de cette cuvette. On parlera alors de *bassin de convergence, pour désigner l'ensemble des transformations initiales permettant d'atteindre le voisinage de la bonne transformation.* Cette définition permettra en outre de prendre en compte les cas extrêmes où le minimum global n'est lui-même pas proche de la bonne transformation (lorsque le manque de robustesse n'est pas dû à une mauvaise initialisation, mais à un mauvais réglage des paramètres ou à un problème dans les données, comme dans les deux cas pathologiques de notre base de test).

Nous étudierons donc dans cette partie la robustesse au travers des bassins de convergence. Nous présenterons ensuite des méthodes pour améliorer la robustesse. Nous verrons enfin comment, quand une parfaite robustesse n'est pas garantie, reconnaître un bon d'un mauvais recalage.

6.2.1 Évaluation expérimentale

Le concept fondamental dans l'étude expérimentale du bassin de convergence (et donc de la robustesse) est la carte des convergences d'un algorithme. Celle-ci est obtenue en faisant fonctionner de nombreuses fois un même algorithme sur les mêmes données en faisant varier l'estimée initiale de la transformation. On note alors les couples transformation initiale/transformation finale. On étudie ensuite la répartition des transformations finales, ce qui permet de dire quelles transformations peuvent être considérées comme à peu près bonnes (leur variabilité sera utilisée plus tard dans l'étude de la répétabilité, voir 6.3.1) et quelles transformations peuvent être considérées comme fausses. On peut alors regarder la répartition des transformations initiales qui ont mené à un résultat juste (i.e. étudier la forme du bassin de convergence) et en déduire la qualité nécessaire sur l'initialisation pour garantir une parfaite robustesse.

Calcul des cartes de convergence

Nous devons ici faire fonctionner le même algorithme (même méthode, même paramètres) sur les mêmes données, en faisant varier la transformation initiale autour de la bonne transformation. Ces variations devront être au moins aussi amples que celles produites par la méthode d'initialisation retenue dans le protocole opératoire (voir section 6.4.1).

Il nous faut une première estimée de la bonne transformation T_0. Celle-ci ne devra pas être nécessairement très précise, on peut même se contenter de la transformation obtenue par la méthode d'initialisation. Pour la faire varier, nous lui appliquerons une transformation marginale, que nous pourrons faire varier de plusieurs manières différentes :

- Par tirage aléatoire autour de l'identité : cette méthode est certes plus représentative de la réalité (si la loi de probabilité est bien choisie, ce qui est déjà un problème en soit), mais présente deux défauts majeurs : d'une part, rien ne garantit que les transformations sont bien réparties, d'autre part les transformations tirées ne seront pas reproductibles d'une expérience à l'autre (e.g. lorsqu'on voudra comparer deux réglages des paramètres de l'algorithme de recalage), interdisant toute comparaison détaillée des résultats.
- Par parcours d'une grille régulière dans l'espace des transformations : on utilisera les transformations de vecteur $(r_x,r_y,r_z,t_x,t_y,t_z)$, en faisant varier chaque coordonnée (t_y par exemple) entre une valeur minimale et maximale ($t_{y\,min}$ et $t_{y\,max}$) avec un pas donné ($t_{y\,pas} = \frac{t_{y\,max} - t_{y\,min}}{t_{y\,n}}$). On aura alors $r_{x\,n} * r_{y\,n} * r_{z\,n} * t_{x\,n} * t_{y\,n} * t_{z\,n}$ transformations à essayer.
- Par parcours régulier des axes de l'espace de des transformations : l'idée est proche du cas précédent, mais on ne fait varier qu'une seule coordonnée à la fois, en laissant les

autres à 0, pour limiter le nombre de transformation et donc de recalages à essayer, qui devient alors $r_{x\,n} + r_{y\,n} + r_{z\,n} + t_{x\,n} + t_{y\,n} + t_{z\,n}$.

Étant considérés les défauts de la première méthode, nous avons retenu les deux dernières méthodes.

Se pose alors la question de la taille et de la résolution de la grille (dont dépend le nombre de recalage et donc le temps de calcul). Comme nous avons remarqué que le recalage était bien plus robuste par rapport à des erreurs de rotation qu'à des erreurs de translation, nous ne faisons en général varier que cette dernière, en restant à rotation nulle. La largeur de la grille doit être un peu plus grande que le bassin de convergence, pour avoir un nombre significatif de bons et mauvais résultats. Aussi doit-on faire une première expérience rapide (parcours régulier des axes par exemple) pour l'évaluer grossièrement tout en vérifiant la faible sensibilité à la rotation. Dans nos applications, on utilise une grille qui va de -10 à $+10\,mm$ dans chaque direction, que nous parcourons ensuite intégralement. Nous avons essayé plusieurs résolutions de grille (jusqu'à 21 pas par axe, soit $1\,mm$ par pas), et avons constaté que les résultats étaient représentatifs dès 5 pas par axe (soit $4\,mm$ par pas). Nous utilisons donc en général cette résolution (et essayons alors 125 transformations), passant parfois à 11 pas par axe quand les recalages sont assez rapides (ce qui fait 1331 transformations).

Classification des transformations finales

Une fois que l'on a fait tourner l'algorithme avec ces différentes transformations initiales, on peut examiner les transformations finales obtenues. La figure 6.2 présente un exemple typique, où on a placé les transformations finales en fonction de la norme de leur translation et la norme de leur rotation.

On constate que les transformations sont réparties suivant deux ensembles bien distincts :

- Un premier ensemble proche de 0, i.e. de la transformation T_0, qui est proche de la bonne. Il s'agit donc des convergences que l'on peut considérer comme à peu près bonnes. On remarque en outre que ces transformations sont grosso-modo alignées, révélant que le recalage est plus mal conditionné suivant une direction couplant rotation et translation.
- Un deuxième ensemble plus éloigné et sans structure spéciale. Il s'agit des mauvaises convergences, uniformément réparties en dehors de la zone de bonne convergence, et plus nombreuses pour de fortes translations que pour de fortes rotations.

A partir de ces constatations, il devient assez aisé de faire le distinguo entre bonne et mauvaise convergence, à l'aide d'un simple seuil sur la norme de la rotation et la norme de la translation (en général 1 mm et 1 degré). Remarquons cependant qu'il faut pour cela que la transformation T_0 soit proche de la bonne solution. En-effet, si elle est elle-

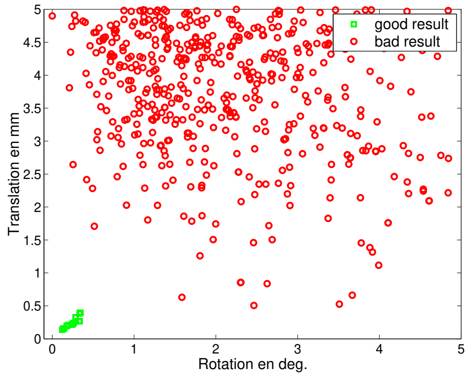


FIG. 6.2 – *Transformations finales d'un recalage Scanner/Pointés Dents placées en fonction de la norme de leur translation et de leur rotation*

même située à plus d'1 mm ou d'1 degré, notre stratégie, qui a l'avantage de pouvoir être automatisée, devient inutilisable. Nous ne pouvons donc plus nous contenter de l'estimée grossière utilisée précédemment. Plusieurs alternatives se présentent alors :

- Déterminer manuellement avant toute chose une estimée suffisamment précise de la bonne transformation et l'utiliser dans la totalité du protocole (i.e. y compris pour générer les transformations initiales).
- Affiner a-posteriori la transformation en prenant par exemple la moyenne des transformations qui semblent bonnes dans la carte des convergences.
- Utiliser la transformation finale de l'expérience qui semble avoir le mieux fonctionné, comme par exemple celle qui a donné le critère le plus faible.

Pour le banc de test automatisé, on a préféré utiliser la première méthode. Il suffit en-effet de réaliser une bonne fois pour toute un recalage correct (i.e. à 2-3 dixièmes de mm et de degrés) pour chaque jeu de données. On essaye pour cela différents paramétrages, on modifie manuellement l'initialisation si nécessaire jusqu'à obtenir un résultat visuellement satisfaisant.

Dans les cas où la vérification visuelle est impossible (par exemple pour des recalages avec quelques points isolés) ou ne permet pas d'atteindre la précision nécessaire (lorsque plusieurs transformations franchement différentes semblent bonnes), on calcule effectivement une première carte de convergence (voire plusieurs en utilisant des paramètres différents), et on recourt aux deux autres méthodes, jusqu'à obtenir une transformation satisfaisante, qui sera utilisée par la suite.

Étude de robustesse

Une fois que l'on a réussi à faire la classification bonne/mauvaise convergence, on peut évaluer la robustesse. Plusieurs niveaux de détail sont possibles.

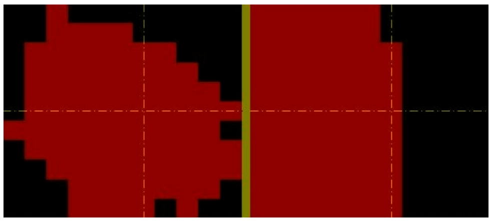


Fig. 6.3 – *Bassin de convergence d'un algorithme de recalage : On voit ici deux coupes (xy et xz) d'une image volumique binaire représentant un bassin de convergence. Les coordonnées de chaque voxel correspondent à la translation initiale (au centre une translation nulle), sa couleur indique si elle a mené à une convergence correcte (en rouge) ou non (en noir).*

On peut se contenter de regarder le pourcentage de convergences correctes. C'est très grossier, mais cela permet de comparer rapidement différents paramétrages.

On peut chercher la transformation initiale la plus proche de la bonne transformation qui a mené à un mauvais résultat. On obtient ainsi la largeur minimale du bassin de convergence, qui donne véritablement une estimation de la qualité nécessaire sur l'initialisation pour garantir une bonne convergence. C'est la technique généralement retenue dans notre banc de test automatisé.

On peut regarder les résultats en détail, pour une meilleure compréhension. Ainsi, lorsqu'on fait varier la translation initiale sur une grille, on peut construire une image volumique binaire dont chaque voxel correspond à une translation initiale et indique si le recalage a convergé. La figure 6.3 présente deux coupes d'une telle image. On peut voir sur cette figure la forme générale des bassins de convergence : ils ont une forme grossièrement elliptique (figure de gauche), mais ne sont pas forcément centrés autour de la bonne transformation. Ils peuvent être limités par un plan (figure de droite), qui correspond en fait à la crête entre le bassin de convergence de la bonne transformation et un bassin de convergence associé à une mauvaise transformation.

6.2.2 Amélioration de la robustesse

Lorsque la robustesse de l'algorithme ne donne pas satisfaction, on peut faire le raisonnement suivant : lorsque le critère forme une cuvette autour d'un mauvais minimum local, et que l'on se trouve dans cette cuvette, on a aucune chance d'en sortir en essayant de minimiser le critère, et la convergence vers le mauvais minima devient inéluctable. On peut alors penser à deux façons d'améliorer la robustesse : la première est de modifier le critère pour supprimer ces minima locaux, la deuxième est de modifier l'algorithme pour lui permettre de sortir d'une telle cuvette. Notre approche multi-échelle correspond à la première solution. Plusieurs approches correspondant à la deuxième solution ont été envisagées dans la littérature. Nous les présentons ici, car, contrairement à notre EM multi-échelle, elles ne sont pas spécifiques au recalage de surface, et peuvent ainsi être reprises pour d'autres algorithmes. Nous ne rentrerons cependant pas dans les détails car ces techniques se sont avérées peu efficaces dans notre cas.

Monte-Carlo et algorithmes génétiques

Dans cette première approche, l'idée est d'effectuer plusieurs recalages à partir d'initialisations différentes (i.e. situées dans des cuvettes différentes du critère) et de ne garder que le meilleur (en espérant que ce soit le minimum global). Il faut donc savoir comment répartir les transformations initiales pour avoir une chance que l'une d'elle se situe dans la bonne cuvette.

Plusieurs approches ont été présentées autour de ces idées : des approches déterministes (les transformations initiales sont réparties sur une grille régulière [Hügli and Schütz, 1997]), des approches de type Monte-Carlo (les transformations initiales sont tirées suivant une loi de probabilité connue, voir par exemple [Simon, 1996]), des approches basées sur des algorithmes génétiques (voir par exemple [Brunnstrom and Stoddart, 1995]), ...

Le principal problème de ces méthodes est qu'elles exigent d'effectuer de nombreux recalages, et multiplient d'autant le temps de calcul. Elles sont donc incompatibles avec les contraintes de temps qui doivent être respectées en per-opératoire.

Recuit-simulé sur la transformation

Cette approche a été largement appliquée aux algorithmes par minimisation de critère non-convexe. L'idée est la suivante : pour éviter de rester coincé dans un minimum local, on va "secouer" la transformation, i.e. la perturber en lui appliquant une transformation marginale suffisamment importante pour sortir du bassin de convergence du minimum local [Blais and Levine, 1995].

Recuit-simulé sur les données

Dans cette approche, introduite par [Penney *et al.*, 2001], on ne bruite plus directement la transformation, mais on bruite les données, en ajoutant à chaque point un bruit gaussien. Ce bruit est variable d'une itération à l'autre, de variance décroissante.

Discussion

Nous avons très vite abandonné ces méthodes en faveur de notre approche multi-échelle. Les méthodes de type Monte-Carlo n'ont même pas été testées en raison de leur lenteur. Notre approche multi-échelle s'est expérimentalement révélée plus efficace que le recuit-simulé sur la transformation, lui-même plus efficace que le recuit simulé sur les données, comme le montre les résultats sur un recalage Pointés-Picza (les paramètres des recuits simulés ont été réglés automatiquement pour optimiser la robustesse, voir section 6.4.5) :

	ICP	ICP+recuit transfo.	ICP+recuit données	EM
Robustesse (%)	23.20	31.20	39.20	61.6
Temps moyen (s)	0.75	1.19	1.11	1.5

Nous avons finalement mis ces méthodes de côté pour les cas extrêmes, et nous sommes focalisé sur notre approche multi-échelle.

6.2.3 Systèmes de sécurité

Lorsque la parfaite robustesse ne peut être garantie (ce qui est en général hélas le cas, cf les deux cas pathologiques de notre base de test que nous prenons ici en compte dans les expériences), il faut pouvoir dire, à l'issu d'un recalage, si l'on a affaire à un résultat correct [18]. Il nous faut donc mettre au point un système de sécurité permettant de classer, pour une application donnée, les bons et les mauvais résultats.

Intuitivement, il semble que la valeur finale du critère de l'algorithme de recalage puisse donner une telle indication. En-effet, pour un modèle et une scène donnée, le critère est en général plus faible pour un bon recalage que pour un mauvais recalage. Nous allons ici développer cette idée, en présentant tout d'abord les diagrammes de sensibilité/sélectivité qui permettent facilement de savoir comment régler un seuil sur la valeur du critère pour classer au mieux les résultats des recalages. Nous verrons alors que le classement obtenu ne sera pas forcément parfait, en particulier lorsqu'on cherchera un seul seuil pour un certain type de données (e.g. pointés sur l'os/scanner) et non pour un modèle et une scène connue. Mais nous ne chercherons pas à trouver un système de sécurité universel, préférant choisir un critère qui fonctionne parfaitement pour une application donnée.

Diagrammes de sensibilité / sélectivité

Nous allons maintenant présenter les diagrammes de sensibilité / sélectivité ("ROC-curve" en anglais [Metz, 1978]) qui permettent d'étudier rapidement les performances d'un classificateur basé sur le seuillage d'une valeur scalaire (pour l'instant, la valeur finale du critère de l'algorithme de recalage), et de régler au mieux le seuil.

Le but de notre classificateur est de distinguer les bonnes des mauvaises transformations. Ce qui nous intéresse, c'est qu'il fasse un minimum d'erreurs. On définit donc :

- La sensibilité : il s'agit du pourcentage de vrais positifs (i.e. de transformation classées comme bonnes qui sont effectivement bonnes) parmis les positifs (les transfor-

18. Dans le cas contraire, on pourra appliquer une des méthodes de la section précédente, ou tout simplement recommencer la procédure d'initialisation. Dans VirtualScope, le système de navigation et de guidage du geste chirurgical ne sera de toute façon pas utilisable tant que le système de sécurité n'aura pas été passé avec succès.

mations classées comme bonnes). Plus il est élevé, moins il y aura de faux positifs (des transformations classées bonnes alors qu'elles sont mauvaises).

– La sélectivité : il s'agit du pourcentage de vrais négatifs (i.e. de transformation classées comme mauvaises qui sont effectivement mauvaises) parmis les négatifs (les transformations classées comme mauvaises).

On veut rejeter systématiquement les mauvais recalages, i.e. avoir un classificateur parfaitement sensible (qui ne produit aucun faux positif). Il est de plus préférable qu'il ne rejette que peu de bons recalages, i.e. qu'il ait une sélectivité aussi élevée que possible. Ceci évitera le travail supplémentaire entraîné par un échec du système de sécurité.

Lorsque la classificateur dépend d'un paramètre, ici le seuil sur la valeur du critère, on peut tracer le diagramme de sensibilité / sélectivité empirique du classificateur. Il nous faut pour cela un ensemble d'expériences, qui soient représentatives de notre application et des résultats possibles de l'algorithme, et pour lesquelles on connaît la bonne classification. Ici, nous pouvons directement utiliser les résultats de l'étude de la robustesses. Nous avons en-effet déjà fait beaucoup de recalages représentatifs de notre application à la section 6.2.1 (nous avons 125 résultats de recalages pour chaque patient, 10 patients dans notre base de test, et avons échantillonné nos transformations initiales sur une grille dont la largeur permet d'avoir un nombre significatif de bons et de mauvais résultats) et les avons classés dans la section 6.2.1.

On fait alors varier la valeur du seuil, et on compte, pour chaque valeur, le nombre de vrais positifs (les résultats de recalages que l'on sait bons et dont le critère à une valeur inférieure au seuil) et faux positifs (les résultats de recalages que l'on sait bons mais dont le critère à une valeur supérieure au seuil), on en déduit la sensibilité, puis la sélectivité à l'aide des vrais et faux négatifs. Enfin, on trace le diagramme, qui est la courbe paramétrée par la valeur du seuil, avec en abscisse la sélectivité et en ordonnée la sensibilité :

$$(x,y)(seuil) = (select(seuil), sensib(seuil))$$

Deux exemples d'un tel diagramme sont donnés dans la figure 6.4. Sur le premier, les paramètres du recalage ont été mal réglés, et il est impossible d'obtenir un classificateur parfait en utilisant la valeur du critère. Sur le second, le recalage a été bien réglé, et on peut trouver un seuil permettant une classification parfaite des bons et mauvais recalages.

Perspectives

Dans notre dernier exemple, nous avons obtenu un classificateur parfait en nous basant directement sur la valeur du critère de l'algorithme de recalage, pour un réglage des paramètres et un cadre applicatif donné (le recalage per-opératoire utilisé dans VirtualScope). On peut cependant se demander s'il ne serait pas possible d'obtenir un classificateur plus général, capable de s'adapter à d'autres problèmes de recalage. Pour cela, il faut d'abord

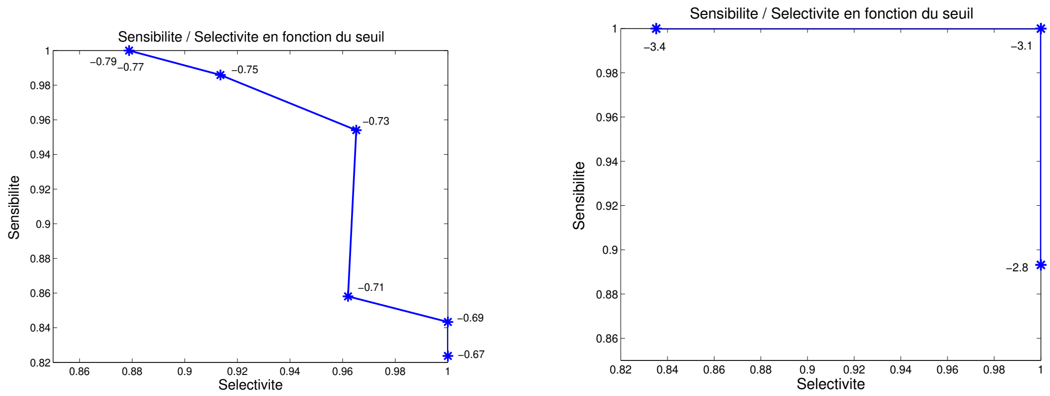


FIG. 6.4 – *Diagrammes de sensibilité / sélectivité pour les recalages pointés sur les dents /scanner sur la base de test AREALL, pour l'ICP/EM avec une variance sur-estimée (à gauche) et une variance correcte (à droite).*
Les courbes sont paramétrées par la valeur du seuil sur le critère de l'algorithme de recalage (très différent entre les deux cas dépendant de la variance du bruit)
Dans le cas de gauche, on ne peut obtenir une classification parfaite : il faut fixer le seuil à -0.77 *pour qu'aucun mauvais recalage ne soit accepté (sensibilité égale à* 100%*), mais on aura alors* 12% *de bons recalages rejetés. En revanche, dans le cas de droite, un seuil à* -3.1 *permettra d'obtenir une sensibilité et une sélectivité parfaite.*

mettre en évidence les différents éléments qui conditionnent les performances des systèmes de sécurité basé sur le seuillage du critère.

Choix et réglage de l'algorithme de recalage L'exemple précédent montre à quel point, sur les mêmes données, un mauvais réglage de l'algorithme peut compromettre l'existence d'un système de sécurité parfait. Mais quel est alors le bon réglage? Nous avons pu constater expérimentalement que le réglage des paramètres qui optimise la robustesse (voir section 6.4.5) est souvent celui qui permet la meilleure classification.

Choix du critère de notre système de sécurité Pourquoi utiliser le même critère que celui du recalage? Une première idée, pour obtenir un classificateur plus général, est de s'affranchir de l'échelle du bruit sur les données, en utilisant un critère qui la prenne en compte. Il pourra par exemple s'agir de la RMS de la distance de Mahalanobis entre points appariés. Mais pour les surfaces, nous savons de plus que la distance tangente n'est pas très pertinente, que seule la distance dans l'axe de la normale doit être faible pour un bon recalage. Aussi le critère qui semble le plus pertinent est la RMS de la distance de Mahalanobis dans l'axe de la normale, i.e. la RMS de la distance entre points appariés dans l'axe de la normale divisée par l'écart-type du bruit normal.

Enfin, on peut finalement songer à s'affranchir complètement de l'algorithme de recalage, en utilisant un critère qui n'est pas basé sur les appariements trouvés par l'algorithme.

Nous développerons cette idée dans la section 8.9.

Le domaine d'application Le seuil optimal de notre système de sécurité, une fois l'algorithme et le critère choisi, dépend du type de données utilisé. Il est en particulier sensible à l'échantillonnage et au bruit de mesure. Il est peu vraisemblable qu'un même seuil soit applicable pour des données d'origines très différentes. On cherchera donc en général à mettre en évidence non pas un seuil universel, mais plusieurs seuils fonctionnant chacun parfaitement sur une catégorie d'application aussi large que possible. Ainsi, dans VirtualScope, on a pu regrouper les recalages de Pointés sur les dents et de Pointés sur l'os, mais avons par contre dû faire la distinction entre les scanners sans ou avec peu d'artefacts (comme ceux de la base de test), et les scanners avec des artefacts spéciaux (tels certains scanners ne figurant pas dans la base de test, pour lesquels l'algorithme de reconstruction tomographique, différent, générait des artefacts à la fois plus nombreux, plus chaotiques et plus diffus dans l'image).

Conclusion

Nous avons ici mis au point un système de sécurité simple et pleinement satisfaisant pour VirtualScope, en nous limitant à l'étude du critère de l'ICP/EM pour un seul type de données et un seul réglage de l'algorithme de recalage. Nous avons ensuite soulevé la question de la mise au point de systèmes de sécurité moins spécialisés, en proposant d'utiliser des critères plus pertinents et prenant en compte l'échelle du bruit. Nous pensons toutefois qu'ils ne seront applicables que pour un algorithme bien réglé (au sens de la robustesse). Nous doutons enfin de la possibilité d'établir un seuil universel (fonctionnant sur des données très différentes), préférant le régler pour chaque application à l'aide du diagramme de sensibilité / sélectivité tracé à partir des résultats des expériences de robustesse.

6.3 Précision

Supposons maintenant que l'algorithme ait bien convergé : la transformation trouvée est au voisinage de la bonne transformation. Mais à quelle distance est-elle ?

Il nous faut pour cela distinguer les différentes sources d'erreurs de l'algorithme :

- le bruit interne : nous avons vu dans le partie consacrée à la robustesse qu'il peut y avoir plusieurs minima locaux, et donc plusieurs convergences possibles à proximité de la bonne solution. De ce fait, avec les mêmes données et les mêmes paramètres, plusieurs résultats sont possibles, ce qui introduit une première source d'imprécision. Celle-ci peut-être interprétée comme due aux erreurs dans l'estimation des appariements.

– le bruit externe : il s'agit de la conséquence du bruit sur les données.
– le bruit de modélisation : il est dû aux défauts de l'algorithme (rappelons par exemple que nous avons supposés les points identifiés, ce qui n'est pas vrai dans le cas des surfaces), qui peuvent entraîner des erreurs même en l'absence de bruit interne et externe, i.e. en présence d'un seul minimum global et de données parfaites.

Le premier bruit peut être isolé. On dit alors que l'on mesure la répétabilité de l'algorithme. Les autres bruits se cumulent pour former l'imprécision globale. On peut alors se demander si ces 3 formes de bruits sont indépendantes les unes des autres.

6.3.1 Évaluation expérimentale des bruits internes et externes

Bruit interne ou répétabilité

Le répétabilité a été définie comme la variabilité introduite par la possibilité d'atteindre plusieurs minima locaux différents aux abords de la bonne transformation. Or nous avons déjà obtenu une carte des convergences de l'algorithme lorsque nous avons étudié expérimentalement la robustesse (voir 6.2.1). On peut donc mesurer directement la répétabilité de l'algorithme en regardant la variabilité des transformations classées bonnes autour de leur moyenne. Ainsi, sur les recalages de notre base de test, on a pu vérifier que la répétabilité de l'ICP était mauvaise (elle induit une TRE de l'ordre de $0.15mm$, et augmente ainsi notoirement l'imprécision de l'algorithme), alors que celle de l'ICP/EM était très bonne (inférieure à $0.05mm$).

Bruit externe

On ne peut en fait pas isoler expérimentalement le bruit externe, car le bruit interne est toujours présent. On peut en revanche mesurer la somme de ces deux bruits.

Rappelons que le bruit externe est la conséquence du bruit de mesure sur les points des deux nuages. Nous allons ici montrer comment étudier expérimentalement la conséquence du bruit sur les points de la scène pour un modèle fixé. Pour cela, il nous faut plusieurs scènes contenant la mesure des mêmes points. Deux possibilités sont offertes pour obtenir ces scènes :

– On peut faire plusieurs acquisitions. C'est ce que nous avons fait avec les nuages de points isolés : nous avons marqué sur une empreinte de mâchoire 40 points, et avons fait 10 acquisitions différentes de ces points, dans le même repère. Nous avons ainsi obtenu 10 nuages contenant les mêmes points avec un bruit de mesure représentatif de la réalité.
– Lorsqu'on ne dispose que d'une seule acquisition, on crée à partir de cette seule acquisition plusieurs nuages différents en bruitant les points suivant un modèle de

bruit aussi représentatif que possible du véritable bruit de mesure (ces méthodes sont déjà décrites à la section 3.9.4).

Ces deux méthodes permettent d'obtenir un jeu de scènes contenant les mêmes points (mais bruités de façon indépendantes suivant le bruit de mesure) dans la même position. Il suffit alors de les recaler toutes sur le même modèle, et de mesurer la variabilité des transformations obtenues autour de leur moyenne, pour obtenir une estimation de la somme du bruit interne et externe.

6.3.2 Évaluation expérimentale de la précision globale

Pour estimer la précision globale du recalage, il faut comparer la transformation obtenue à la véritable transformation. Il faut donc disposer d'une estimation de cette dernière, appelée transformation étalon. Il existe plusieurs méthodes d'évaluation de la précision globale basées sur des transformations étalons différentes. Ces méthodes se distinguent par deux aspects :

- La qualité de la transformation étalon obtenue : cet étalon ne correspond jamais exactement à la bonne transformation. Aussi, lorsqu'on comparera la transformation obtenue dans un recalage à la transformation étalon, on mesurera à la fois l'erreur de recalage et l'erreur sur la transformation étalon (si les deux méthodes sont statistiquement indépendantes, la variance mesurée sera la somme de la variance du recalage et de la variance de l'étalon). Ainsi, si la précision du recalage est meilleure que celle de l'étalon, on mesurera essentiellement cette dernière, et les résultats apparaîtront bien plus mauvais qu'ils ne le sont.
- Les biais systématiques ne pouvant être pris en compte : lorsque la méthode d'évaluation de l'étalon et le méthode de recalage sont corrompues par un même biais (e.g. lorsqu'ils s'appuient sur la même image biaisée), la comparaison de l'étalon et du résultat du recalage ne permettra pas de faire apparaître ce biais. Les résultats apparaîtront alors meilleurs qu'ils ne le sont.

Nous allons ici présenter plusieurs méthodes que nous avons utilisé, basées sur plusieurs qualités d'étalons différents.

Étalon or : utilisation d'une méthode certifiée

On parlera d'étalon or lorsque la transformation est imposée ou mesurée à l'aide d'une méthode certifiée, i.e. dont la précision est connue, et dont le biais est négligeable. Ce peut-être à l'aide d'instruments de métrologie 3D (robot passif, système optique, capteur magnétique), ou à l'aide de méthodes de recalage validées (recalage manuel, recalage basé sur des marqueurs). Nous allons détailler ici deux de ces méthodes que nous avons utilisées, puis abandonnées parce que les étalons obtenus étaient trop peu précis.

Marqueurs radiopaques La méthode basé sur des recalages de marqueurs (billes ou tiges radiopaques, casques stéréotaxiques [West *et al.*, 1997], ...) a souvent été utilisée comme méthode étalon dans la validation des algorithmes de recalage. Nous avons donc testé une telle technique, que nous présentons ici.

Nous avons utilisé en guise de marqueurs des billes radiopaques d'environ 1 mm de diamètre, placées sur la mâchoire lors du passage au scanner (ils peuvent être directement collés lorsqu'on travaille sur des mâchoires nues, ou placés sur une gouttière pour un patient). Les marqueurs sont alors bien visible et faciles à segmenter dans l'image scanner. Ils doivent ensuite être replacés sur la mâchoire (lorsqu'ils ne sont pas collés, dans le cas d'une gouttière par exemple), puis mesurés (en les pointant avec notre outil de navigation par exemple). On réalise enfin un recalage entre les marqueurs radiopaques, appariés manuellement, pour obtenir notre transformation.

On obtient ainsi un étalon or. En-effet, la méthode de recalage de points appariés que nous avons utilisé [Pennec, 1996] est reconnue par la communauté scientifique, et les biais sont négligeables (le scanner et l'outil de navigation et donc les images sont certifiés sans biais, ainsi que la méthode de recalage). Enfin, on peut calculer sa précision en fonction de la précision de mesure des points [Pennec, 1996]. Celle-ci est hélas très mauvaise (de l'ordre du voxel pour la segmentation dans le scanner, probablement supérieure au mm avec l'outil de navigation), et il faut aussi rajouter l'erreur supplémentaire introduite par le replacement des marqueurs sur la mâchoire. L'erreur sur la transformation étalon ainsi estimée serait de l'ordre du millimètres, et donc trop importante pour notre application.

Rotation imposée On a aussi essayé, dans certains cas, d'imposer mécaniquement la transformation. On a par exemple fixé une mâchoire sur un support cubique, posé sur un marbre et aligné contre la face d'une équerre fixée. On a ensuite imposé la rotation (la translation étant elle difficile à imposer ou mesurer précisément) en alignant le support de la mâchoire sur une seconde face de l'équerre. Nous avons appliqué une seule fois ce protocole, sur des données Picza. Les erreurs mesurées sur la rotation se sont révélées pertinentes (de l'ordre de 0.5 degrés, ce qui est compatible avec les erreurs mesurées par d'autres méthodes comme l'étalon bronze ou l'étalon zinc présentés dans les sections suivantes). Mais nous avons rapidement abandonné cette méthode qui ne donne aucune indication sur la qualité des translations.

Étalon bronze : Comparaison de plusieurs méthodes de recalage

On parlera d'étalon bronze lorsque la transformation étalon est obtenue en examinant les résultats d'algorithmes de recalages très différentes sur les mêmes images. On est alors tributaire d'un biais systématique sur les images elles-mêmes : si l'image du modèle est décalée par exemple de 2 cm vers le haut, toutes les méthodes donneront une transformation décalée de 2 cm vers le haut, et on ne pourra pas repérer de biais systématique. Remarquons que dans notre système, ce n'est en fait pas gênant, puisqu'on travaille directement dans les repères des images (image scanner et image per-opératoire) lors de la navigation.

L'avantage d'une telle approche est double. D'une part, si l'un des algorithmes de recalage a un biais systématique, on pourra le repérer (sauf s'ils ont tous le même ce qui serait malheureux). D'autre part, les algorithmes de recalages étant très différents les uns des autres, on peut raisonnablement supposer que leurs résultats sont statistiquement indépendants.

Recalage Scanner/Scanner Nous avons utilisé une approche de type étalon bronze pour quantifier la précision des recalages scanners/scanners. Nous disposions en-effet de trois scanners différents pour les patients de la Base de Test (voir 6.1.5).

Pour chaque patient, nous avons recalé les scanners entre eux en utilisant des algorithmes iconiques (block-matching [Ourselin, 2002; Ourselin *et al.*, 2001], rapport de corrélation, Information Mutuelle [Roche *et al.*, 2000; Roche, 2001]) et géométriques (lignes de crête [Pennec *et al.*, 2000], ICP/EM).

Nous avons ensuite estimé les précisions de chaque méthode, en appliquant la méthode du "Leave One Out" : pour chaque patient, nous avons calculé l'étalon (comme la moyenne des transformations obtenues) SANS utiliser le résultat de la méthode à évaluer (pour garantir l'indépendance statistique), puis avons comparé l'étalon et la transformation obtenue par la méthode à évaluer. Les résultats (les chiffres donnés sont l'écart-type de la rotation et de la TRE sur les 9 patients, les cibles étant placées au sommet de la boite englobante des mâchoires dans les images) ont montré que l'ICP/EM était le plus précis (TE rotation 0.08 deg, TRE 0.10 mm) bien qu'il ait fallu ajuster les réglages pour les cas très artefactés; les autres algorithmes ont tous donné des résultats comparables (0.13 deg, 0.20 mm).

On a finalement calculé un étalon global pour chaque patient (en faisant la moyenne des transformations obtenues par toutes les méthodes), et estimé que sa précision était de l'ordre de 0.03 deg et 0.05 mm (en utilisant la formule classique de la covariance de la moyenne de n mesures indépendantes).

Boucles de consistances

Nous allons ici présenter la technique des boucles de consistances. Elles ont été introduites pour pouvoir évaluer l'incertitude cumulée (la consistance) de plusieurs recalages. Elles peuvent être utilisées en l'absence totale d'étalon (auquel cas il est difficile de déduire une incertitude pour chaque recalage), ou en présence d'un étalon de précision connue sur un des recalages (auquel cas on peut déduire une précision pour les autres recalages).

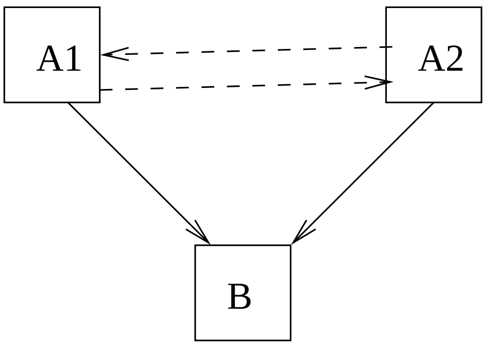
Nous allons ici présenter ce dernier cas.

Pour les boucles de consistances, il nous faut disposer de trois images A, B et C que l'on puisse recaler deux à deux. La composition des trois transformations obtenues $T_{boucle} = T_{A \to B} \star T_{B \to C} \star T_{C \to A}$ doit recaler l'image A sur elle même, et donc être égale à l'identité. L'erreur par rapport à l'identité donne alors une indication de la consistance, i.e. des erreurs cumulées des trois recalages. Si les recalages sont statistiquement indépendants, les variances (ou covariances) des erreurs (TE ou TRE) des trois recalages s'additionnent et on a donc :

$$\sigma^2_{boucle} = \sigma^2_{A \to B} + \sigma^2_{B \to C} + \sigma^2_{C \to A}$$

Cette formule ne permet pas dans le cas général de déduire l'erreur pour chaque recalage, mais donne au moins un majorant et un ordre de grandeur.

Aussi allons nous présenter ici une variante, basée sur l'étalon bronze obtenu précédemment, qui permet véritablement d'évaluer la précision d'un recalage. L'idée est d'utiliser deux images A_1 et A_2 de même type (scanner en l'occurrence) et une image de type éventuellement différent B (nuage de points per-opératoires dans notre cas) :



Nous avons vu ci-dessus comment obtenir un étalon de précision connue pour les recalages entre A_1 et A_2. Une fois cet étalon connu, on peut mettre en place un banc de test automatique dont le principe est le suivant : on recale chaque donnée B sur les différentes données A_i. On mesure alors la consistance de la boucle $T_{boucle} = T_{B \to A_2} \star T_{A_2 \to A_1} \star T^{-1}_{B \to A_1}$. On a alors :

$$\sigma^2_{boucle} = \sigma^2_{A_i \to A_j} + \sigma^2_{B \to A_i} + \sigma^2_{B \to A_j}$$

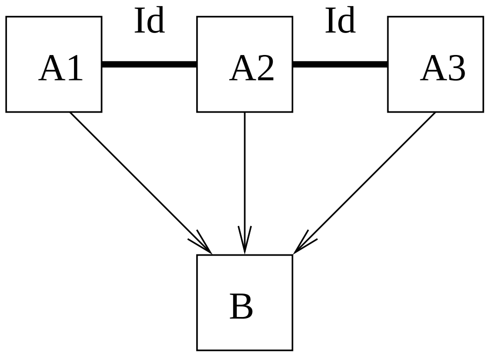
On retrouve une fois l'erreur de recalage entre les A_i et deux fois l'erreur de recalage entre B et A_i. On peut négliger la première (on ne fait alors que sur-estimer l'erreur, comme lorsqu'on utilise un mauvais étalon), et considérer que les recalages entre B et A_i suivent la même loi (ils font intervenir des images du même type, avec les mêmes niveaux de bruits et la même configuration anatomique, puisqu'il s'agit du même patient) et ont donc la même variance. On mesure donc au final à peu près le double de l'erreur de recalage entre données de type A et données de type B :

$$\sigma^2(T_{A_i \to B}) = \sigma^2_{boucle}/2$$

Étalon zinc : comparaison de résultats de la même méthode

Nous abordons maintenant le cas le plus difficile : on ne dispose d'aucune méthode alternative pour estimer la bonne transformation, et on en est réduit à comparer des transformation obtenues par la même méthode de recalage (en faisant éventuellement varier les paramètres). Le principal problème d'une telle approche est qu'elle ne permet pas de repérer des biais systématiques dans la méthode de recalage même (par exemple si elle travaille avec une mauvaise origine du repère qui a pour effet de décaler tous les résultats).

Le cas que nous présentons ici est celui où l'on dispose d'une seule donnée de type B (le scanner) et de plusieurs données du même type A_1, A_2, ..., A_n prises dans la même position (les nuages de points per-opératoires dans notre cas). Ces données A_i doivent être représentatives de la variabilité réelle des acquisitions et statistiquement indépendants. Il ne faut par exemple pas chercher à acquérir les mêmes points. L'idéal est que les différentes acquisitions soient effectuées par des personnes différentes :



On peut bien-sûr appliquer la méthode des boucles de consistances (la transformation entre les A_i est parfaitement connue, c'est l'identité), en composant les transformations entre A_i et B deux à deux. On peut aussi prendre directement en compte toutes les transformations entre A_i et B, car elles doivent être identiques.

On procède alors de la façon suivante : on effectue tous les recalages entre entre A_i et B. Nous supposerons alors que les transformations $T_{A_i \to B}$ suivent la même loi (cela semble réaliste, dans la mesure où les données A_i ont été acquises de la même façon) et sont statistiquement indépendantes. On peut alors estimer la bonne transformation comme la moyenne $\overline{T}$ des $T_{A_i \to B}$, et calculer la variance non-biaisée des transformations :

$$\sigma^2(T_{A_i \to B}) = \frac{1}{n-1} \sum_{i=1}^{n} d^2(T_{A_i \to B}, \overline{T})$$

La limite de cette méthode, outre son incapacité à repérer les biais systématiques (la transformation moyenne calculée est alors fausse), réside dans le nombre de transformations nécessaires. Il faut en faut au moins une dizaine pour avoir un résultat statistiquement pertinent. Or il est long et difficile de réaliser les nombreuses acquisitions nécessaires (nous n'en n'avons que deux jeux : 40 nuages de 30 à 60 points sur les dents d'un plâtre, et 10 nuages de 50 points orientés sur les dents d'une mâchoire sèche).

Variante "Jack-Knife" Il existe une variante appelée "Jack-Knife" [Duda *et al.*, 2001, Section 9.4] de ces méthodes lorsqu'on dispose d'un seul nuage de points A : il s'agit de diviser (de façon équitable) ce nuage en n sous nuages A_i, que l'on considère comme n acquisitions différentes dans la même position. On mesure alors la précision du recalage $\sigma^2(T_{A_i \to B})$ entre les sous-nuages et l'image B. On peut alors conclure en se disant que la précision du recalage $\sigma^2(T_{A \to B})$ avec le nuage entier sera au moins meilleure (puisqu'il y a plus de points), probablement de l'ordre de $\sigma^2(T_{A_i \to B})/n$ (lorsque la précision du recalage suit à peu près la loi des grands nombres, ce qui semble être en général le cas).

6.3.3 Conclusion

Nous avons présenté ici plusieurs méthodes permettant de faire des mesures de précision sur des données réelles en l'absence de mesure étalon. Ces méthodes sont basées sur l'acquisition de plusieurs images pour une même modalité, dans des positions identiques (étalon zinc) ou différentes (étalon bronze). Elles ont en commun de reposer sur une estimation de la transformation exacte aux vues des différents recalages, si bien qu'elles ne permettent pas de repérer un biais systématique, et peuvent ainsi sous-estimer l'imprécision globale du recalage.

Ces méthodes soulèvent plusieurs grandes perspectives :

- L'hypothèse d'indépendance entre différentes méthodes de recalages : nous avons supposé, lors du calcul de l'étalon bronze et de sa précision, que les différentes méthodes étaient statistiquement indépendantes les unes des autres. Ceci est probablement vrai dans notre cas, car nous utilisons des algorithmes très différentes. Mais il serait intéressant de pouvoir quantifier et prendre en compte l'éventuelle corrélation entre deux méthodes de recalage, ou deux réglages différents des paramètres pour une même méthode. Ceci permettrait probablement de mieux comprendre le fonctionnement des algorithmes et les effets des différentes variantes et des différents paramètres.
- L'hypothèse d'indépendance entre les différents recalages effectués sur un même patient : cette hypothèse n'est bien-sûr pas vérifiée lorsqu'on utilise la même image ou le même nuage de points dans deux recalages différents (le même nuages de points per-opératoires avec l'étalon bronze, le même scanner avec l'étalon zinc). Dans notre cas, ce défaut peut être traité comme un biais systématique, puisque l'un des nuages est fixé une bonne fois pour toute. Mais lorsqu'on va multiplier le nombre d'images (et en particulier lorsqu'on va avoir plusieurs modèles et plusieurs scènes), il va falloir réfléchir à la bonne façon de généraliser les boucles de consistances : va-t-on

choisir des boucles qui fassent intervenir des images toutes différentes pour garantir leur indépendance (mais limiter ainsi le nombre de boucles possibles), ou va-t-on tenter de quantifier et prendre en compte la corrélation entre deux boucles faisant intervenir une image commune ?

– La prise en compte de plusieurs patients : nous avons jusqu'ici présenté des méthodes permettant d'évaluer la précision d'un recalage pour un patient donné. Lorsque nous disposions de plusieurs patients, nous avons gardé la précision moyenne ou le cas le plus mauvais. Mais il faudrait à l'avenir une approche un peu plus sophistiquée, permettant aussi de quantifier les corrélations entre recalage sur plusieurs patients, ou de prendre en compte des niveaux de bruits différents entre deux patients (ce qui permettrait d'établir une formule empirique liant le niveau de bruit sur les données et la précision du recalage, à comparer aux formules théoriques de la section 5.4).

Les réponses à ces questions permettront probablement de mettre en place un système basé sur les boucles de consistances bien plus sophistiqué, permettant une estimation des transformations exactes (et donc des erreurs de recalages) plus sûre à partir de plusieurs méthodes de recalages (ou de variantes de la même méthode) et sur plusieurs patients.

6.4 Évaluation de la qualité des données et réglage des paramètres

Nous allons maintenant voir comment régler les différents paramètres de l'algorithme. Nous le ferons en deux étapes :

– nous les réglerons tout-d'abord manuellement. Ceci sera pour nous l'occasion de résumer le rôle des différents paramètres de notre algorithme ICP/EM multi-échelle, et de donner leurs valeurs types.
– nous les optimiserons ensuite à partir de cette première estimation, de façon complètement automatique, de manière à maximiser la robustesse et la précision du recalage, que l'on peut mesurer grâce aux bancs de tests présentés aux sections précédentes.

Pour la première étape, nous aurons en particulier besoin d'évaluer le niveau de bruit sur les données. Nous en profitons donc ici pour présenter les méthodes permettant d'évaluer le bruit sur les données et la transformation initiale.

6.4.1 Évaluation de la qualité de l'initialisation

Il nous faut dans un premier temps évaluer la qualité de notre initialisation, i.e. de notre estimée initiale de la transformation. Rappelons que celle-ci est calculée par un recalage de points identifiés. On peut donc envisager deux façons d'évaluer la précision de ce recalage et donc de la transformation initiale obtenue :

- Expérimentalement, en appliquent une stratégie identique à celle présentée dans la section 6.3.2 : on acquiert plusieurs fois les points per-opératoires utilisés pour l'initialisation, et on regarde la variabilité du résultat.
- Théoriquement, en utilisant la formule permettant de prévoir l'incertitude d'un tel recalage (voir section 3.9). Il faut pour cela utiliser une estimation de la variance du bruit de mesure sur l'acquisition de chacun des points.

Les deux méthodes donnent des résultats comparables, une imprécision de l'ordre de $2mm$.

6.4.2 Évaluation du bruit sur les données

Il nous faut maintenant évaluer le niveau de bruit sur les données (i.e. sur la position des points et l'orientation des normales), pour savoir comment fixer, dans l'algorithme, les distances et angles caractéristiques. Nous le ferons dans un premier temps de façon très heuristique. Nous verrons ensuite comment obtenir une valeur plus rigoureuse en regardant les premiers résultats (et en particulier les valeurs des critères) de recalages.

Approche Heuristique On connaît toujours un ordre de grandeur de l'écart-type σ du bruit sur la position des points :

- Pour les scanners, il est de l'ordre de la taille de voxel (quelques dixièmes de mm).
- Pour les nuages per-opératoires, il est de l'ordre de la précision de l'outil utilisé pour prendre les points (précision que l'on connaît par des méthodes de métrologies indépendantes développées par AREALL).

On peut alors faire un premier recalage en utilisant un ICP tout simple, tel que présenté dans la section 3.2 (sans décimation, sans utiliser les normales, avec des poids unitaires, une distance quadratique isotrope, et une distance de coupure égale au triple de σ). On vérifiera visuellement que le résultat n'est pas complètement faux (rappelons que l'ICP simple est peu robuste). Il est important, pour la suite, de ne pas sous-estimer σ. On peut repérer de tels cas, en regardant le taux d'appariements : s'il parait trop faible par rapport au taux de recouvrement de la scène par rapport au modèle, c'est que σ a été sous-estimé. Il faut alors l'augmenter et relancer l'algorithme jusqu'à avoir un taux d'appariements satisfaisant.

Utilisation des valeurs des résultats des recalages Nous avons déjà expliqué, dans la section 3.2.1, que la valeur du critère sous sa forme la plus simple $C(T,A) = \frac{1}{\sum_{ij} A_{ij}} . \sum_{ij} A_{ij} . d^2(T \star s_i, m_j)$ était, pour un bon recalage, une estimation statistique de la variance des erreurs. Les différentes modifications que nous avons depuis apporté à l'algorithme vont cependant biaiser cet estimateur. L'annexe A présente les calculs nécessaires pour obtenir une valeur non biaisée. Elle peut être utilisée pour estimer les niveaux de bruits pour un type de donnée.

6.4.3 Réglage des paramètres d'échelles

Nous allons maintenant détailler le rôle des différents paramètres de notre algorithme ICP/EM multi-échelle, et montrer comment les régler, pour obtenir un algorithme précis et robuste. Nous illustrerons à chaque fois nos propos avec des résultats expérimentaux obtenus sur la base de test AREALL, figure 6.1. La précision est mesurée par la technique des boucles de consistances avec les scanners, le chiffre donné est l'écart-type de la TRE (les cibles étant réparties sur l'ensemble de la mâchoire). La robustesse est le pourcentage de convergences pour des translations initiales réparties sur une grille régulière de 1 cm de demi-largeur.

Nous ferons la distinction entre :

- les paramètres qui indiquent l'échelle de base (l'écart-type sur la position des points, sur les orientations) et qui peuvent être déduits de la section précédente. Attention, car le rôle et le réglage de ces paramètres pourra changer sensiblement pour des types de données très différents (présentant des bruits ou des échantillonnages particuliers).
- les paramètres invariants par changement d'échelle (distance de Mahalanobis maximale, coefficient α de la décimation, coefficient d'échelle initial et final de l'approche multi-échelle) pour lesquels on exhibera des valeurs utilisables dans la plupart des applications.

Écart-type sur la position des points (σ)

Il s'agit du paramètre le plus important de l'algorithme, car il conditionne la forme du critère (voir section 4.3.1). Comme le montre la section précédente, il doit être égal au niveau de bruit estimé sur les données, et nous avons déjà vu comment le déterminer. Pour mieux comprendre son rôle, nous allons détailler ici les conséquences d'une sous-estimation ou d'une sur-estimation de sa valeur.

La sous-estimation produit les effets les plus dramatiques. La forme du critère dépend en-effet de l'écart-type qui définit l'échelle de travail. Si l'échelle est sous-estimée, le critère devient irrégulier, les minima locaux se multiplient. Les conséquences sont doubles : d'une part l'algorithme a plus de chance de rester piégé dans un minimum, et est donc beaucoup moins robuste. D'autre part, le nombre de minima locaux autour du minimum global augmente, et la répétabilité (et donc la précision) se détériore.

La sur-estimation est moins grave. Elle améliore la robustesse, mais fait chuter la précision. En-effet, les poids accordés aux appariements sont moins sélectifs (ils favorisent moins les points les plus proches), et donc moins pertinents.

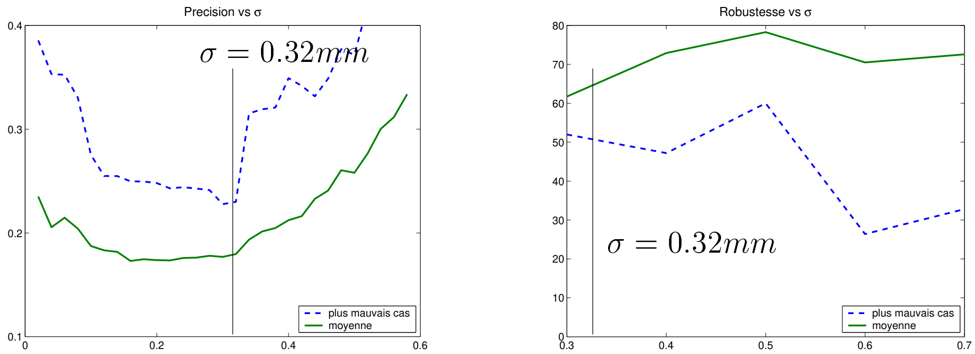


FIG. 6.5 – *Précision et robustesse sur la base de test AREALL de l'ICP/EM multi-échelle en fonction de l'échelle* σ*, ici de l'ordre de* $0.32\,mm$*.*

Points semi-anisotropes σ_t

Lorsqu'on utilise les points semi-anisotropes gaussiens, il y a deux écart-types à régler. Le premier, l'écart-type dans la direction normale, joue exactement le même rôle que l'écart-type présenté précédemment. Le second, l'écart-type dans le plan tangent, doit prendre une valeur au moins égale (pour avoir une bonne précision), voire deux ou trois fois supérieure, en prenant en compte le pas d'échantillonnage (voir section 5.3.3). La précision sera légèrement détériorée, mais la robustesse considérablement améliorée.

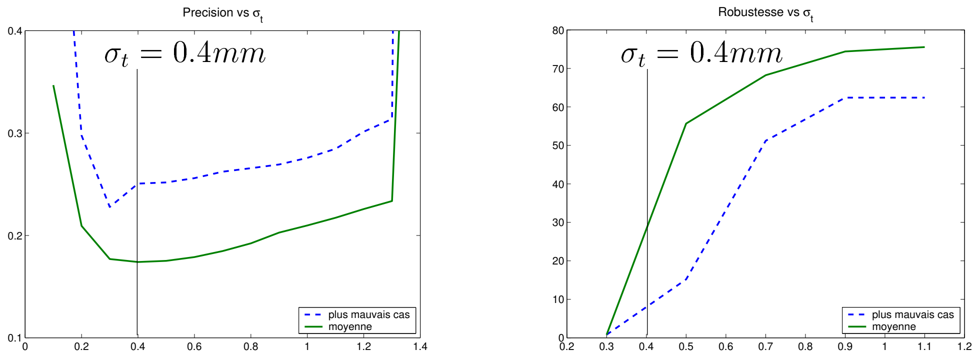


FIG. 6.6 – *Précision et robustesse sur la base de test AREALL de l'ICP/EM multi-échelle avec utilisation des points anisotropes gaussiens en fonction de l'écart-type dans le plan tangent* σ_t*.*

Écart-type sur l'orientation des normales σ_n

L'écart-type sur l'orientation des normales joue un rôle moins important. Les normales ont en-effet une influence faible sur le calcul de la transformation. En revanche, elles jouent un rôle dans le calcul des poids des différents appariements, favorisant les points dont les normales se correspondent, dans le but de rejeter plus de points aberrants et être un peu plus robuste et précis.

Aussi, sous-évaluer l'écart-type multipliera le nombre de points rejetés, ce qui peut être désastreux surtout pour la précision de l'algorithme.

Le sur-évaluer atténuera le rejet des points aberrants, et donc les gains en précision (et dans une moindre mesure en robustesse).

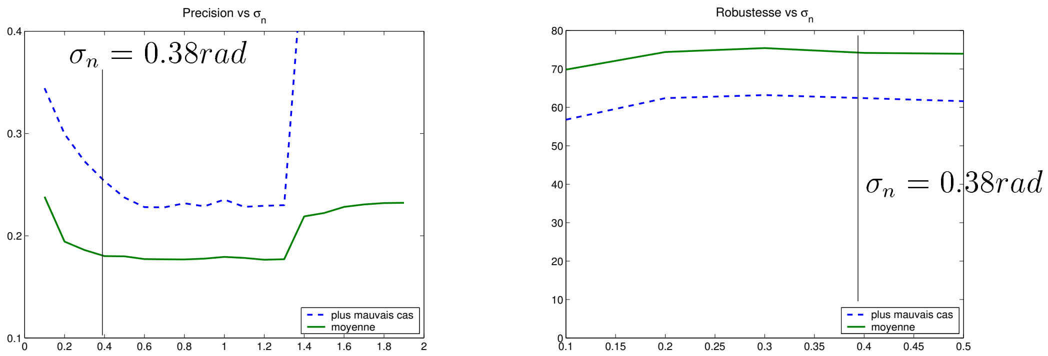


FIG. 6.7 – *Précision et robustesse sur la base de test AREALL de l'ICP/EM multi-échelle avec utilisation des normales en fonction de l'écart-type σ_n sur l'orientation des normales, ici de l'ordre de* $0.38\,rad$.

6.4.4 Réglage des paramètres sans échelle

Nous continuons ici l'analyse des différents paramètres, en nous intéressants maintenant aux paramètres sans échelle (i.e. sans unités). Il semble que l'influence de ces paramètres dépende peu du type de données utilisées, dès lors que les paramètres d'échelles sont correctement réglés. Nous exhiberons donc pour chacun une plage de valeur utilisable dans la plupart des applications.

Rejet des points aberrants μ_{max}^2

Le paramètre qui contrôle le rejet des points aberrants est la distance de Mahalanobis maximale μ_{max}^2. Nous avons déjà expliqué à la section 3.6.2 comment la régler, aux alentours de 3^2.

On peut préciser ici les effets d'un mauvais réglage, qui sont contradictoires. En-effet, sous-évaluer μ_{max}^2 aura peu d'influence sur l'incertitude (on prend en compte que les appariements les plus proches et donc les plus pertinents, mais le nombre d'appariements diminue), réduira le temps de calcul (puisque la distance de recherche diminue) mais détériorera la robustesse (lorsque la transformation initiale est assez éloigné, le nombre d'appariements trouvés sera trop faible pour que l'on se rapproche de la bonne solution). Bien évidement, pour une valeur très faible (inférieure à 1), le nombre d'appariements chute dramatiquement, et l'algorithme se comporte très mal.

Lorsqu'on sur-estimera la distance de recherche, le nombre de points aberrants pris en compte va se multiplier. Jusqu'à une valeur de 4^2, les performances se détériorent légèrement. Au delà, l'algorithme devient très vite inutilisable.

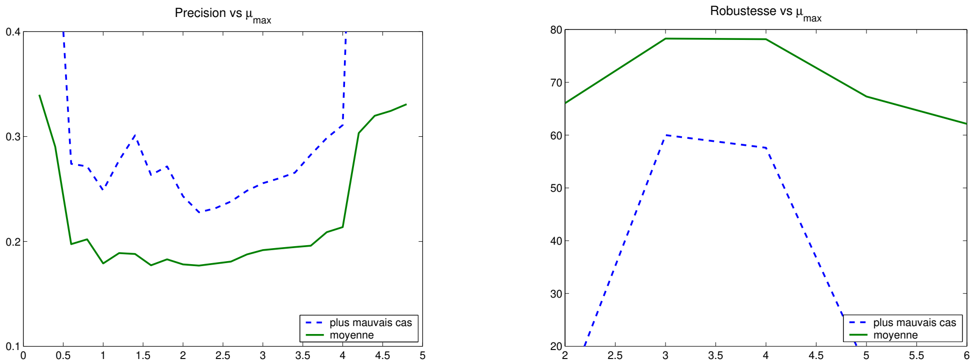


FIG. 6.8 – *Précision et robustesse sur la base de test AREALL de l'ICP/EM multi-échelle en fonction de la racine carré de la distance de Mahalanobis maximale* μ_{max}^2.

Décimation

La décimation est contrôlé par le paramètre α qui donne le rayon des sphères en fonction de l´échelle. Comme expliqué dans la section 4.4.3, ce paramètre doit être fixé entre 1 et 2.

Le sous-estimer a pour principal effet d'augmenter fortement le temps de calcul. De plus, sur des données échantillonnées de façon irrégulières, la décimation permet de ré-homogénéiser l'échantillonnage, et donc de limiter les déséquilibres entre différentes zones de la surface qui entraîne parfois une détérioration de la robustesse et de l'incertitude. Un coefficient α trop faible ne permettra pas de profiter de ses effets.

Une sur-estimation de α (au delà de 3) aura des effets plus graves : elle fera chuter le nombre de points de façon trop importantes, et la précision comme la robustesse se détérioreront rapidement.

Multi-échelle

L'approche multi-échelle est contrôlée par deux paramètres : le facteur d'échelle initial (le facteur final est toujours égal à 1), et la vitesse de diminution de l'échelle. Elle n'a absolument aucune influence sur la précision (qui ne dépend que de l'échelle finale), mais a la plus grosse influence sur la robustesse.

Le choix du facteur d'échelle initial dépend de la robustesse souhaitée, i.e. de la qualité de la transformation initiale disponible. Nous n'avons hélas pas réussi à établir de lien quantitatif entre les deux. Bien évidement la robustesse va en augmentant avec l'échelle de départ. Il est cependant inutile d'utiliser une échelle trop grande qui augmente le nombre

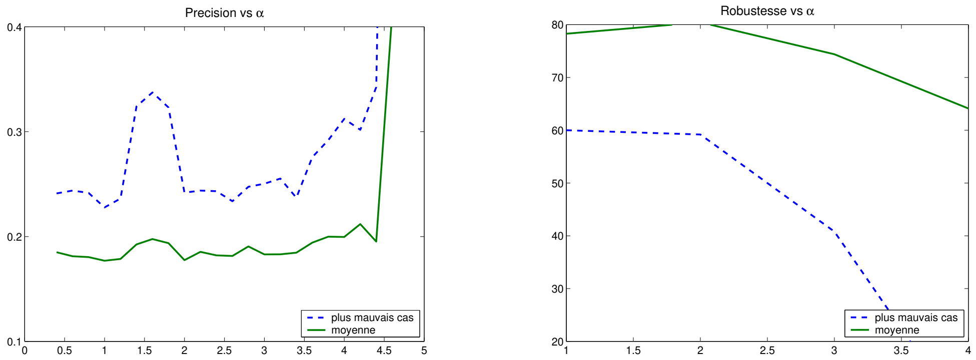


FIG. 6.9 – *Précision et robustesse sur la base de test AREALL de l'ICP/EM multi-échelle en fonction du facteur de décimation α.*

d'itérations et donc le temps de calcul. De plus, sur certains cas, nous avons constaté expérimentalement qu'une échelle initiale trop grande a pour effet d'amener le recalage sur une transformation trop éloignée de la bonne qui mène ensuite vers un mauvais minimum local lorsqu'on diminue l'échelle. On utilise donc en général un facteur d'échelle initial de l'ordre de 8^2.

En ce qui concerne la vitesse de diminution de l'échelle, nous multiplions en général l'échelle σ^2 par 0.95 après chaque itération. Une valeur plus faible entraînerait une diminution trop rapide de l'échelle, et l'approche multi-échelle perdrait sont intérêt. Une valeur plus forte entraîne une diminution trop lente du facteur d'échelle, qui n'apporte guère de gain en robustesse, mais augmente fortement le nombre d'itérations nécessaires pour atteindre l'échelle finale, et donc le temps de calcul.

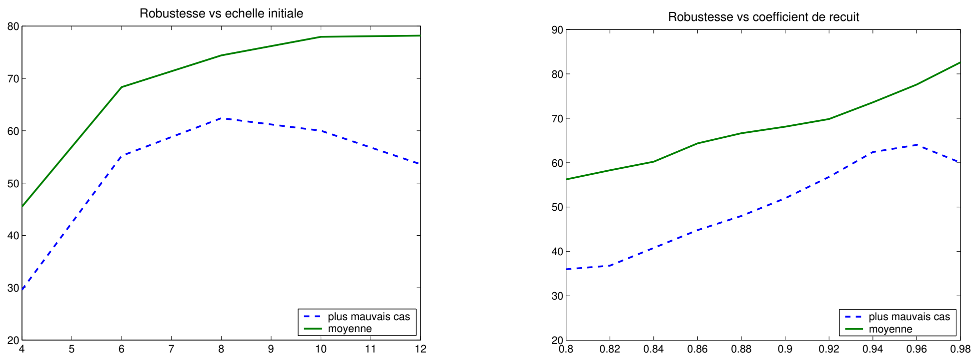


FIG. 6.10 – *Robustesse sur la base de test AREALL de l'ICP/EM multi-échelle en fonction du coefficient d'échelle initial (à gauche) et du coefficient de recuit simulé (à droite)*

6.4.5 Optimisation des paramètres

Maintenant que nous connaissons le rôle de chaque paramètre, nous pouvons, pour chaque type de recalage, faire un premier réglage encore grossier mais déjà efficace. Nous allons ici montrer comment, à partir de ce premier réglage, effectuer une optimisation automatique des paramètres pour obtenir les résultats les plus précis ou les plus robustes.

Nous allons pour cela optimiser l'un des indicateurs expérimentaux mesurés par nos bancs de test automatisés :

- Précision : nous utiliserons les méthodes d'évaluation basées sur les boucles de consistances avec pour indicateur, l'écart-type de la TRE.
- Robustesse : nous utiliserons la méthode basée sur les cartes de convergences, avec pour indicateur le taux d'expériences ayant convergé.

Nous ferons à chaque fois tourner notre banc sur la base de test. Notre mesure finale correspond à l'expérience ayant le moins bien fonctionné. On pourra ainsi garantir une précision ou une robustesse minimum.

Pour réaliser l'optimisation de l'indicateur, on alterne l'optimisation suivant chaque paramètre [19] : pour chacun, on essaye plusieurs valeurs de paramètres différentes, réparties dans la zone où ce paramètre peut donner de bons résultats, et on garde la meilleure valeur.

6.5 Conclusion

Nous avons dans ce chapitre expliqué comment utiliser l'algorithme mis au point dans les chapitres précédents (i.e. l'ICP/EM multi-échelle avec utilisation de la distance entre normales et des points semi-anisotropes), en nous focalisant sur les deux éléments qui font sa qualité : la robustesse et la précision, notions que nous avons expliqué en détail.

Nous avons montré comment estimer le bruit sur les données, puis avons détaillé le rôle des différents paramètres, et les valeurs à leur donner pour un fonctionnement correct de l'algorithme.

Nous avons surtout montré comment évaluer expérimentalement la robustesse et la précision des algorithmes de recalage rigide. La particularité des techniques présentées sont les suivantes :

- Elles sont entièrement automatisées, ce qui a permis de leur adjoindre une sur-couche permettant le réglage automatique des paramètres pour un fonctionnement optimal de l'algorithme.
- Elles donnent des chiffres clairs : pour la convergence, le taux d'échec en fonction de la qualité de la transformation initiale, et pour la précision, l'écart-type ou la valeur maximale des erreurs en orientation et en positionnement au niveau d'une ou plusieurs cibles choisies par l'utilisateur (typiquement autour du site où l'on va opérer).

19. Il s'agit donc d'un algorithme de minimisation alternée, à l'image de l'ICP. Il possède le même défaut, celui de pouvoir rester coincé dans un minimum local. On le fait donc en général tourner à partir de plusieurs estimées initiales des paramètres très différentes, pour explorer les différents minima locaux, et on ne garde que le meilleur résultat.

- Elles ne nécessitent pas de connaître une mesure étalon. Pour la robustesse, il suffit d'effectuer un premier recalage grossier pour chaque patient. Pour la précision, il faut disposer pour chaque patient d'au moins deux images de l'une des modalités (dans des positions éventuellement différentes si on peut les recaler entre elles, comme dans le cas du scanner) pour pouvoir construire des "boucles de recalages" dont on mesure la consistance.
- Elles peuvent prendre en compte (de manière encore simpliste) plusieurs patients différents.

Nous avons enfin abordé le dernier élément nécessaire au bon fonctionnement du système VirtualScope, à savoir la mise en place de systèmes de sécurité permettant de vérifier dans un contexte clinique que les recalages ont correctement fonctionné.

Deuxième partie

Surfaces aléatoires

Avec la première partie de ce manuscrit, il semble bien qu'on ait répondu à la problématique du recalage dans le cadre de VirtualScope. Nous lui avons donné une base théorique statistique et l'avons inséré dans un schéma multi-échelle au chapitre 4. Nous l'avons testé, optimisé, sécurisé et validé dans le cadre de l'application VirtualScope au chapitre 6. Il reste bien-sûr beaucoup de perspectives (ne serait-ce que travailler sur d'autres applications et types de données) et de détails à résoudre, mais une question centrale reste latente dans tous ces développements :

Comment modéliser, par le biais des points orientés, des éléments de surfaces échantillonnées aléatoires ?

Nous utilisons des points orientés dans le recalage, notamment avec la distance entre deux points orientés (dans la section 5.2) qui permet d'améliorer l'élimination des points aberrants. Mais on considère alors que la répartition des points et des normales sont indépendantes ce qui est certes vrai pour les amers géométriques, mais manifestement faux pour les surfaces échantillonnées. Il y a bien-sûr un bruit de mesure sur le modèle comme sur la scène, que l'on peut souvent supposer comme non-corrélé. Mais derrière ce bruit se cache un second phénomène aléatoire qui est dû à l'échantillonnage et impose une corrélation entre la position des points et des normales : le point de la scène est une mesure d'un point orienté de la surface au voisinage du point du modèle et non au même endroit. Il faudrait alors prendre en compte deux choses : la façon dont a été échantillonnée la surface et la forme de la surface.

Nous avons justement, dans la section 5.3, développé un modèle d'appariement semi-anisotrope, qui tient compte de la normale au point du modèle et en tire un modèle de bruit gaussien non-isotrope (vaguement) justifié par une approximation plane de la surface et gaussienne de l'échantillonnage. Ce modèle ne tient en revanche pas compte de la normale au point de la scène. Les améliorations qu'il apporte à la prédiction (section 5.4) sont insuffisantes mais significatives et encourageantes.

Il nous faut donc dans un premier temps disposer de vrais outils statistiques pour manipuler les points orientés aléatoires et, avant cela, les normales aléatoires. Nous présenterons, pour ces derniers, l'approche théorique, justifierons des approches un peu plus pratiques, et présenterons un premier exemple d'application au chapitre qui suit.

Enfin, il nous reste à proposer dans le chapitre 8 un vrai modèle probabiliste, qui prédise où peut se trouver un point de la scène connaissant un point du modèle, en prenant correctement en compte les trois phénomènes (à peu près) indépendants que l'on a mis en évidence :

– Celui lié à la forme de la surface qui indique où peut se trouver un point de la scène situé à telle distance du point du modèle. Il doit au moins prendre en compte une échelle, qui correspond à la vitesse de variation de la normale (i.e. la courbure), mais aussi à la vitesse à laquelle les points peuvent se disperser autour du plan tangent.

- Celui lié à l'échantillonnage qui indique à quelles distances le point de la scène peut se trouver du point du modèle. Le modèle sera en général relativement simple, faisant intervenir un seul paramètre d'échelle, la densité de l'échantillonnage. Mais les positions relatives des points sur une même surface vont parfois présenter des distances privilégiées en fonction de la régularité et du pas de l'échantillonnage.
- Celui enfin lié à la mesure du point qui vient s'ajouter au produit des deux premiers. On le modélisera avec un simple bruit gaussien paramétré par la seule échelle du bruit.

Nous verrons alors comment utiliser ces différents modèles de bruit dans le recalage, et dans les différentes applications correspondant aux nombreux travaux qui ont inspiré ces deux chapitres.

Chapitre 7

Normales aléatoires

Sommaire

7.1 Introduction

L'espace des normales (i.e. l'espace des directions) ne forme pas un espace vectoriel, mais une variété différentielle analogue à une sphère en 3D (ou une hypersphère en dimension quelconque). Le calcul des probabilités et des statistiques usuel ne peut donc s'appliquer simplement dans un tel espace.

Les travaux de Pennec [Pennec, 1996] ont permis de dégager un cadre rigoureux permettant de généraliser les notions usuelles de probabilités et statistiques sur une telle variété différentielle. Il montre en particulier qu'il existe une généralisation cohérente dans les variétés différentielles globalement invariantes par un groupe de transformation (rigide, affine, ...). Nous présentons dans l'annexe D un résumé plus intuitif que rigoureux de ces travaux, et détaillons le cas des normales orientées et non-orientées dans la section 7.2.

Nous verrons alors que la moyenne et la covariance ainsi définies pour les normales sont difficiles à calculer et manipuler (le calcul de la moyenne est itératif, la représentation de la covariance varie suivant cette moyenne). Nous reviendrons alors sur deux techniques couramment utilisées pour calculer la normale "moyenne" : la somme normalisée (dans le cas orienté), et la technique du tenseur symétrique (dans le cas non-orienté). Nous justifierons ces techniques comme des approximations au deuxième ordre, et verrons en particulier que la technique du tenseur permet de calculer et manipuler à la fois la moyenne et la covariance de façon extrêmement pratique. Ensuite, nous comparerons expérimentalement ces techniques en faisant le parallèle avec un article présentant des travaux analogues sur les rotations.

Enfin, nous verrons une première application du tenseur symétrique qui permettra d'introduire le vote de tenseur sur lequel nous reviendrons plus longuement dans le chapitre suivant. Il s'agit de l'estimation des normales pour un nuage de points non structurés et non orientés provenant d'une variété quelconque (une courbe en 2D, une courbe ou une surface en 3D). Cette technique est très fortement inspiré des travaux de Médioni sur la reconstruction de courbes et surfaces par vote de tenseurs [Médioni *et al.*, 2000]. En fait, c'est en essayant de modéliser son approche efficace mais très heuristique que nous avons développé les notions de ce chapitre. Et la technique d'interpolation des normales que l'on justifie ici théoriquement est, à quelques corrections près, identique à la sienne.

7.2 Statistiques exactes sur les normales aléatoires

L'ensemble des normales orientées $\mathcal{S}_{\mathfrak{D}-1}$ ou non-orientées $\mathcal{P}_{\mathfrak{D}-1}$ (analogues respectivement à une sphère et un hémisphère) sont tous deux des sous-variétés C^∞ de dimension $\mathcal{D} = \mathfrak{D} - 1$. Nous allons ici redonner les résultats de l'annexe D pour ces ensembles. Nous nous limiterons en général au cas 3D, le cas 2D étant assez évident, et les dimensions supérieures sans intérêt immédiat pour nous.

7.2.1 Normales et transformations rigides

Il nous faut dans un premier temps examiner les propriétés de l'ensemble des normales par rapport au groupe des transformations rigides, i.e. définir une normale origine, le sous-groupe d'isotropie et les fonctions de placement.

Rappelons tout d'abord l'effet d'une transformation rigide sur une normale :

$$(R,t) \star \mathtt{n} = R.\mathtt{n}$$

Aucune normale n'ayant de propriété particulière, l'origine de l'espace des normales peut être fixé de manière arbitraire. Nous utiliserons $\mathtt{o} = (1,0,0)$ par la suite.

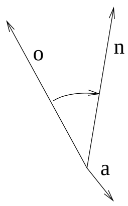
Le sous-groupe d'isotropie est constitué de l'ensemble des transformations rigides qui laissent cette normale inchangée. La translation n'ayant aucune importance, reste à savoir quelles rotations laissent une normale inchangée. Il s'agit bien-sûr des rotations ayant pour axe cette normale, et un angle quelconque, qui seront donc notées $R_{\mathtt{o},\theta}$. En résumé, le sous-groupe d'isotropie est :

$$\mathcal{H} = \{(R_{\mathtt{o},\theta},t); t,\theta\, quelconque\}$$

dans le cas non-orienté, il faudra de plus rajouter la composition de ces rotations par les symétries (i.e. les rotations d'angle $180\, deg$) d'axe perpendiculaire à $\mathtt{o}$.

Reste à trouver une fonction de placement simple permettant d'atteindre la normale $\mathtt{n}$ à partir de la normale origine $\mathtt{o}$. Nous prendrons une translation nulle, et la rotation autour d'un axe perpendiculaire à ces deux normales est un choix simple, car cet axe est obtenu par produit vectoriel : $a = \frac{\mathtt{o}\wedge\mathtt{n}}{\|\mathtt{o}\wedge\mathtt{n}\|}$, et l'angle de la rotation est alors l'angle entre les deux normales $(\widehat{\mathtt{o},\mathtt{n}})$. Finalement, la fonction de placement canonique est :

$$\mathtt{f}_{\mathtt{n}} = R_{(\frac{\mathtt{o}\wedge\mathtt{n}}{\|\mathtt{o}\wedge\mathtt{n}\|},(\widehat{\mathtt{o},\mathtt{n}}))}$$



Notons que dans le cas où la normale $\mathtt{n}$ est égale à $\mathtt{o}$ ou à son opposée, le vecteur $\mathtt{o}\wedge\mathtt{n}$ est nul, et la formule ci dessus ne peut être appliquée. Lorsque $\mathtt{n} = \mathtt{o}$, l'identité est la fonction de placement la plus simple. Lorsque $\mathtt{n} = -\mathtt{o}$, n'importe quelle rotation d'angle

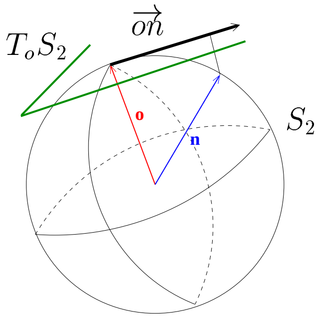


FIG. 7.1 – *Plan tangent et carte locale pour les normales*

180 degrés et d'axe perpendiculaire à $\mathtt{o}$ peut convenir :

$$\mathtt{f}_{\mathtt{o}} = Id$$

$$\mathtt{f}_{-\mathtt{o}} = R_{(a,\pi\, rad)} \quad avec\ a \perp \mathtt{o}$$

7.2.2 Espace tangent et carte locale

Nous allons maintenant définir les éléments de bases de la géométrie différentielle des normales, i.e. identifier les espaces tangents et proposer des cartes locales.

L'espace des normales est identifié à la sphère (demi-sphère dans le cas non-orienté) unité. L'espace tangent en une normale $\mathtt{n}$ est, comme le suggère l'intuition, le plan tangent à la sphère, d'équation $(\mathtt{x}|\mathtt{n}) = 1$ (voir fig. 7.1). Nous identifierons, pour simplifier, ce plan au plan d'équation $(\mathtt{x}|\mathtt{n}) = 0$ (i.e. le plan parallèle mais passant par l'origine de l'espace vectoriel ambiant).

Essayons maintenant de définir une carte locale centrée autour d'une normale $\mathtt{o}$, dans le plan tangent en $\mathtt{o}$. Soit donc une normale $\mathtt{n}$. Il est naturel de représenter cette normale, dans la carte locale, par un vecteur colinéaire, et de même sens que le projeté orthogonal $\Pi_{\mathtt{o}\perp}(\mathtt{n}) = \mathtt{n} - \mathtt{o}(\mathtt{n}|\mathtt{o})$ de $\mathtt{n}$ sur le plan tangent ($\Pi_{\mathtt{o}\perp}$ est la projection orthogonale d'axe $\mathtt{o}$). Ce vecteur sera donc de la forme : $\overrightarrow{on} = \lambda.\Pi_{\mathtt{o}\perp}(\mathtt{n}) = \lambda.(\mathtt{n} - \mathtt{o}(\mathtt{n}|\mathtt{o}))$ (voir fig. 7.1).

Plusieurs possibilités sont offertes pour la longueur de ce vecteur :

1. La distance angulaire : $\|\overrightarrow{on}\| = (\widehat{\mathtt{o},\mathtt{n}})$.
2. La distance vectorielle, définie à partir de la distance quadratique de $\mathbb{R}^{\mathfrak{D}}$: $\|\overrightarrow{on}\| = \|\mathtt{o} - \mathtt{n}\| = 2\sin(\frac{(\widehat{\mathtt{o},\mathtt{n}})}{2})$.
3. La distance tensorielle, définie par $\lambda = 1$, soit $\overrightarrow{on} = \Pi_{\mathtt{o}\perp}(\mathtt{n}) = (\mathtt{n} - \mathtt{o}(\mathtt{n}|\mathtt{o}))$ et $\|\overrightarrow{on}\| = \sin(\widehat{\mathtt{o},\mathtt{n}})$.

Nous utiliserons dans un premier temps la distance angulaire, qui est la distance intrinsèque comme le montre la section suivante. Nous reviendrons sur les deux autres distances dans la partie traitant des statistiques approchées (section 7.3).

7.2.3 Géodésiques et Distance intrinsèque

Nous allons maintenant définir une métrique Riemanienne sur l'ensemble des normales et trouver les géodésiques et la distance intrinsèque associée. Pour définir une métrique sur l'ensemble des normales, il faut définir une métrique sur chaque espace tangent. Nous souhaitons bien-sûr utiliser une métrique invariante par transformation rigide. Il est alors naturel d'utiliser la restriction de la métrique euclidienne usuelle de l'espace 3D à la sphère des normales.

Les géodésiques (i.e. les chemins les plus courts entre deux points) sont alors les grands cercles de la sphère. Une géodésique minimisante est donc un arc d'un de ces cercles unitaires. Sa longueur est proportionnelle à l'angulation entre les deux extrémités. La distance intrinsèque entre normale est donc bien l'angulation :

$$d(\mathtt{n}_1,\mathtt{n}_2) = (\widehat{\mathtt{n}_1,\mathtt{n}_2}) \tag{7.1}$$

7.2.4 Carte exponentielle et lieu de coupure

Nous avons présenté trois cartes locales pour l'ensemble des normales. La première (avec la distance intrinsèque $\|\overrightarrow{on}\| = (\widehat{\mathtt{o},\mathtt{n}})$) est la carte exponentielle. Nous avons aussi expliqué à cette occasion comment calculer $\overrightarrow{on}$ à partir de $\mathtt{o}$ et $\mathtt{n}$, i.e. comment calculer le logarithme :

$$\overrightarrow{on} = \log_{\mathtt{o}}(\mathtt{n}) = (\widehat{\mathtt{o},\mathtt{n}}) \frac{\mathtt{n} - \mathtt{o}(\mathtt{n}|\mathtt{o})}{\|\mathtt{n} - \mathtt{o}(\mathtt{n}|\mathtt{o})\|} \tag{7.2}$$

Pour exprimer l'exponentielle, i.e. calculer $\mathtt{n}$ à partir de $\mathtt{o}$ et $\overrightarrow{on}$, il faut faire tourner la normale $\mathtt{o}$ d'un angle de $\|\overrightarrow{on}\| = (\widehat{\mathtt{o},\mathtt{n}})$, dans la direction indiquée par $\overrightarrow{on}$. Pour cela, il suffit d'appliquer à $\mathtt{o}$ la rotation d'axe $\mathtt{o} \wedge \overrightarrow{on}$ et d'angle $\|\overrightarrow{on}\|$. Étant donné que $\mathtt{o}$ est de norme unitaire et que $\mathtt{o}$ et $\overrightarrow{on}$ sont par construction perpendiculaires, le produit vectoriel $\mathtt{o} \wedge \overrightarrow{on}$ a effectivement pour norme $\|\overrightarrow{on}\|$. Finalement, il suffit d'appliquer à $\mathtt{o}$ la rotation de vecteur $\mathtt{o} \wedge \overrightarrow{on}$:

$$\mathtt{n} = \exp_{\mathtt{o}}(\overrightarrow{on}) = R_{\mathtt{o}\wedge\overrightarrow{on}}(\mathtt{o}) \tag{7.3}$$

Le lieu de coupure[20] de cette carte exponentielle correspond à un quart (pour les normales non-orientées) ou un demi-tour (dans le cas orienté) sur la sphère, i.e. à $\|\overrightarrow{on}\| = \pi/2$ (l'ensemble des normales perpendiculaires à $\mathtt{o}$) pour les normales non-orienté et $\|\overrightarrow{on}\| = \pi$ (la normale $-\mathtt{o}$) pour le cas orienté.

Rappelons maintenant la principale propriété de la carte exponentielle. La norme (i.e. la distance au centre de la carte) est égal à la distance intrinsèque :

$$\|\overrightarrow{on}\| = d(\mathtt{o},\mathtt{n}) = (\widehat{\mathtt{o},\mathtt{n}})$$

20. Le lieu de coupure est le contour du domaine où les géodésiques sont minimisantes, i.e. où la carte est bijective.

7.2.5 Forme volume

Il nous reste enfin à savoir comment définir la forme volume qui est nécessaire pour intégrer des fonctions définies sur l'ensemble des normales.

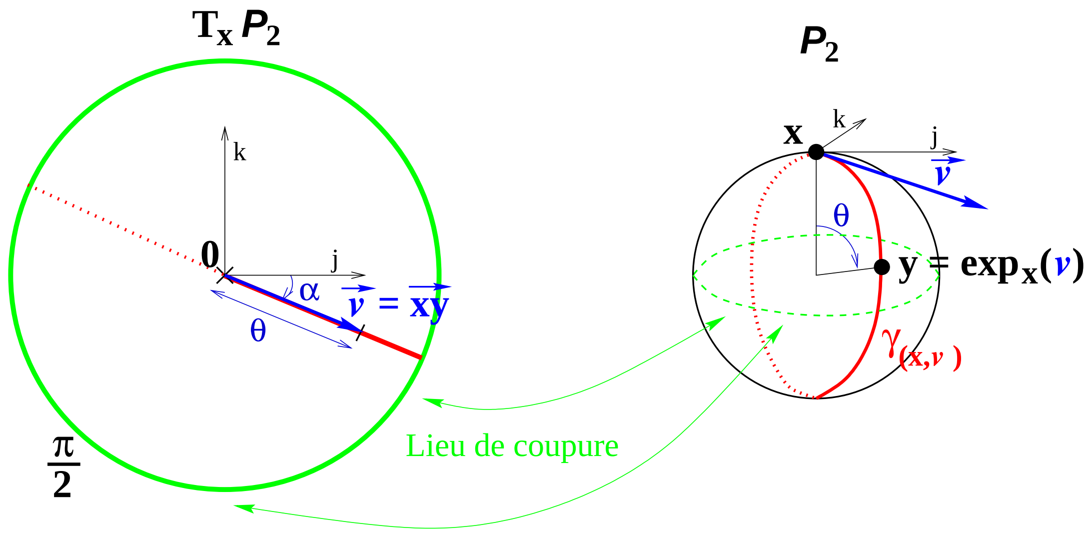
Il faut pour cela calculer le déterminant du tenseur métrique. On utilisera la base canonique de $\mathbb{R}^3$, notée $\{i,j,k\}$. Le plan tangent en l'origine $\mathtt{o} = (1,0,0)$ est donc le plan (j,k). La carte principale est donc définie de la manière suivante :

$$\exp \quad : \quad \begin{array}{ccc} \{\overrightarrow{n} = (x_1,x_2)/x_1^2 + x_2^2 < \frac{\pi}{2}\} & \to & \mathcal{P}_2 \\ \overrightarrow{n} = (\theta\cos\alpha,\theta\sin\alpha) & \mapsto & \exp(\overrightarrow{n}) = (\cos\theta, \sin\theta\,\cos\alpha, \sin\theta\,\sin\alpha) \end{array}$$

Pour alléger les notations, on utilisera plutôt la carte équivalente (en coordonnées polaires) :

$$\mathtt{n} \quad : \quad \begin{array}{ccc} \overrightarrow{n} = (\theta,\alpha)/\theta \in [0,\pi/2[,\alpha \in]-\pi,\pi] & \to & \mathcal{P}_2 \\ \overrightarrow{n} = (\theta,\alpha) & \mapsto & (\cos\theta, \sin\theta\,\cos\alpha, \sin\theta\,\sin\alpha) \end{array}$$

Tout-ceci est illustré sur la figure suivante :



On peut alors calculer les dérivés de la carte locale :

$$\begin{pmatrix} \frac{\partial \mathtt{n}}{\partial \theta} & \frac{\partial \mathtt{n}}{\partial \alpha} \end{pmatrix} = \begin{pmatrix} -\sin\theta & 0 \\ \cos\theta\,\cos\alpha & -\sin\theta\,\sin\alpha \\ \cos\theta\,\sin\alpha & \sin\theta\,\cos\alpha \end{pmatrix}$$

puis calculer le déterminant du tenseur métrique (en adaptant la section D.1.3.2) :

$$[G_{\mathtt{n}(\theta,\alpha)\bullet\bullet}] = \begin{pmatrix} \frac{\partial \mathtt{n}}{\partial \theta} & \frac{\partial \mathtt{n}}{\partial \alpha} \end{pmatrix}^t . \begin{pmatrix} \frac{\partial \mathtt{n}}{\partial \theta} & \frac{\partial \mathtt{n}}{\partial \alpha} \end{pmatrix}$$

$$[G_{\mathtt{n}(\theta,\alpha)\bullet\bullet}] = \begin{pmatrix} \cos^2\theta\,\cos^2\alpha + \cos^2\theta\,\sin^2\alpha + \sin^2\theta & \dots \\ -\cos\theta\,\cos\alpha\sin\theta\,\sin\alpha + \cos\theta\,\sin\alpha\sin\theta\,\cos\alpha & \sin^2\theta\,\sin^2\alpha + \sin^2\theta\,\cos^2\alpha \end{pmatrix}$$

$$[G_{\mathtt{n}(\theta,\alpha)\bullet\bullet}] = \begin{pmatrix} 1 & 0 \\ 0 & \sin^2\theta \end{pmatrix}$$

et donc

$$\sqrt{\left|[G_{\mathtt{n}(\theta,\alpha)\bullet\bullet}]\right|} = \sin\theta \tag{7.4}$$

On peut alors intégrer sans problème les fonctions données sous la forme $f(\theta,\alpha)$ en utilisant l'équation D.13 :

$$\int_{\mathcal{P}_2} f(\mathtt{n}).d\mathcal{P}_2(\mathtt{n}) = \int f(\theta,\alpha).\sin\theta.d\theta.d\alpha$$

7.2.6 Normale aléatoire, densité et espérance d'un observable

On peut toujours définir une normale aléatoire au travers de sa mesure de probabilité $Pr_{\mathbf{n}}(A) = Pr(\mathbf{n} \in A)$ qui indique la probabilité que $\mathbf{n}$ se trouve dans le sous-ensemble A de l'ensemble des normales. Mais cette approche est peu pratique, et on préférera utiliser la métrique pour définir une densité de probabilité, i.e. une fonction $p_{\mathbf{n}}(\mathtt{n})$ telle que :

$$Pr_{\mathbf{n}}(A) = \int_A p_{\mathbf{n}}(\mathtt{n}).d\mathcal{P}_2(\mathtt{n})$$

et on peut alors facilement calculer l'espérance d'une observable, i.e. une fonction φ de l'ensemble des normales et à valeurs réelles ou vectorielles :

$$E_{\mathbf{n}}[\varphi] = \int_{\mathcal{P}_2} \varphi(\mathtt{n}).p_{\mathbf{n}}(\mathtt{n}).d\mathcal{P}_2(\mathtt{n})$$

On peut aussi faire les calculs dans la carte exponentielle (voir eq. D.16 et 7.4) :

$$p_{\mathbf{n}}(\theta,\alpha) = p_{\mathbf{n}}(\mathtt{n}).\sin\theta$$

$$E_{\mathbf{n}}[\varphi] = \int_{\mathcal{P}_2} \varphi(\mathtt{n}).p_{\mathbf{n}}(\mathtt{n}).d\mathcal{P}_2(\mathtt{n}) = \int \varphi(\theta,\alpha).p_{\mathbf{n}}(\theta,\alpha).d\theta.d\alpha = \int \varphi(\theta,\alpha).p_{\mathbf{n}}(\mathtt{n}).\sin\theta.d\theta.d\alpha$$

Exemple : normale aléatoire uniforme en 3D La demi-sphère des normales non-orientées est de volume (en l'occurrence surface) fini. On peut donc définir une normale aléatoire uniforme sur la totalité de l'espace des normales.

Sa densité de probabilité sera de la forme $p_{\mathbf{n}}(n) = k$. Nous travaillerons dans la carte habituelle (la carte exponentielle en coordonnées polaires), où cette densité prend la forme $p_{\mathbf{n}}(\theta,\alpha) = k.\sin\theta$, on peut alors calculer la constante de normalisation :

$$E_{\mathbf{n}}[1] = 1$$
$$\int_{\theta=0}^{\frac{\pi}{2}} \int_{\alpha=-\pi}^{\pi} k.\sin\theta \, d\theta \, d\alpha = k.2\pi. \int_{\theta=0}^{\frac{\pi}{2}} \sin\theta \, d\theta = k.2\pi = 1$$

Finalement : $k = 1/(2\pi)$.

On peut aussi calculer sa variance totale (que nous définirons dans la section suivante) par rapport à la normale correspondant à $\theta = 0$. Il s'agit de l'espérance de l'observable θ^2 :

$$E\left[\theta^2\right] = \frac{1}{2\pi} \int_{\theta=0}^{\frac{\pi}{2}} \int_{\alpha=-\pi}^{\pi} \theta^2 \sin\theta \, d\theta \, d\alpha = \int_{\theta=0}^{\frac{\pi}{2}} \theta^2 \sin\theta \, d\theta$$
$$= \left[-x^2\cos x + 2x.\sin x + 2\cos x\right]_0^{\pi/2} = \pi - 2 \tag{7.5}$$

une normale uniforme a donc un écart-type total (non-infini !) de $\sqrt{\pi - 2} = 61.2\, deg$.

Algorithme : tirage d'une normale aléatoire uniforme en nD Pour obtenir une normale aléatoire uniforme, la méthode la plus simple consiste à effectuer un tirage de vecteur uniforme à l'intérieur d'un volume à symétrie sphérique (comme une boule), et normaliser ensuite ce vecteur. Les directions étant réparties uniformément dans le volume, les normales ainsi obtenues le seront aussi.

Reste donc à tirer un vecteur aléatoire dans ce volume. Pour cela, il suffit de tirer un vecteur aléatoire dans n'importe quel ensemble contenant ce volume, et rejeter les vecteurs situés à l'extérieur.

L'ensemble où il sera le plus simple de tirer un vecteur aléatoire est le cube, car on peut tirer les coordonnées facilement (il s'agit d'un tirage uniforme) et indépendamment les unes des autres. Le volume à symétrie sphérique le plus simple sera la boule inscrite dans ce cube auquel on aura retiré une petite boule concentrique (pour éviter les divisions par une norme trop faible lors de la normalisation).

L'algorithme est donc le suivant :

Faire

. tirer un vecteur aléatoire uniforme : $x = (UniformRnd([-1,1]),UniformRnd([-1,1]),...)$

. calculer sa norme au carré : $r^2 = \sum_i x_i^2$

Jusqu'à $(r^2 > 0.01)$ et $(r^2 < 1)$

Normaliser $n = x/r$

7.2.7 Espérance et covariance

On peut maintenant calculer la variance totale d'une normale aléatoire par rapport à une normale connue. Il s'agit de l'espérance de la distance au carré à cette normale :

$$E_{\mathbf{n}}\left[d^2(\mathtt{n},\mathtt{m})\right]$$

On définit alors l'espérance de Fréchet (resp. Karcher) de $\mathbf{n}$ comme l'ensemble des minima globaux (resp. locaux) de la variance totale. On peut alors calculer l'espérance de Karcher $\mathtt{m}$ d'une normale aléatoire $\mathbf{n}$ par descente de gradient :

Algorithme : Calcul de l'espérance de Karcher par descente de gradient

- Initialisation : $\mathtt{m}^0$
- A l'itération I : on calcule l'espérance de $\overrightarrow{m^I n}$ dans la carte exponentielle centrée en $\mathtt{m}^I$, et on en déduit la nouvelle estimée : $\mathtt{m}^{I+1} = \exp_{\mathtt{m}^I}\left(E_{\mathbf{n}}\left[\overrightarrow{m^I n}\right]\right)$
- Jusqu'à convergence : $d(\mathtt{m}^I,\mathtt{m}^{I+1}) < \varepsilon$

Rappelons que l'espérance de Karcher inclut l'espérance de Fréchet, équivalent de la moyenne dans le cas vectoriel, et que les deux sont restreintes à une seule normale lorsque la normale aléatoire est suffisamment concentrée. On pourra dans ce cas utiliser l'algorithme précédent pour la calculer. En ce qui concerne l'initialisation, on pourra par exemple utiliser l'approximation au sens vectoriel de l'espérance (voir 7.3.2.1).

Enfin, on pourra définir la covariance d'une normale aléatoire par rapport à une normale connue en utilisant la carte exponentielle :

$$\Sigma_{\mathbf{nn}}(\mathtt{m}) = E_{\mathbf{n}}\left[\overrightarrow{mn}.\overrightarrow{mn}^t\right]$$

et en particulier définir la covariance par rapport à l'espérance :

$$\Sigma_{\mathbf{nn}} = \Sigma_{\mathbf{nn}}(\overline{\mathtt{n}})$$

7.2.8 Lois gaussiennes

Nous définissons ici les lois gaussiennes sur les normales. La loi gaussienne d'espérance $\overline{\mathtt{n}}$ et de covariance $\Sigma_{\mathbf{nn}}$ est définie par la densité de probabilité de la forme :

$$N_{(\overline{\mathtt{n}},\Sigma_{\mathbf{nn}})}(\mathtt{m}) = \frac{e^{-\frac{\overrightarrow{\overline{n}m}^t.\Delta_{\mathbf{nn}}^{-1}.\overrightarrow{\overline{n}m}}{2}}}{k} \tag{7.6}$$

où k (constante de normalisation) est solution de :

$$k = E_{\mathbf{n}}[1]$$

et $\Delta_{\mathbf{nn}}$ (matrice de dispersion) est solution de :

$$\Sigma_{\mathbf{nn}} = E_{\mathbf{n}}\left[\overrightarrow{\overline{n}m}.\overrightarrow{\overline{n}m}^t\right]$$

Il n'existe pas de solution explicite, et il n'y a en particulier plus égalité entre la matrice de dispersion et la covariance, comme dans un espace vectoriel. Mais les formules du cas vectoriels constituent de bonnes approximations pour des normales faiblement dispersées :

$$\Sigma_{\mathbf{nn}}\,\text{faible} \quad \Rightarrow \quad \begin{array}{c} \Delta_{\mathbf{nn}} \simeq \Sigma_{\mathbf{nn}} \\ k \simeq (2\pi)^{(\frac{\mathcal{D}}{2})}.\sqrt{\Sigma_{\mathbf{nn}}} \end{array} \tag{7.7}$$

Notons aussi qu'une dispersion infinie (i.e. d'inverse nulle) permettra de représenter une loi uniforme.

On peut se demander alors ce qu'il se passe lorsque l'on ajoute une erreur gaussienne centrée à une normale aléatoire qui suit déjà une loi gaussienne. Il semble que le résultat soit une loi gaussienne dont la matrice de dispersion est la somme des deux matrices de dispersions.

Pour bien distinguer variances et dispersions, rappelons que nous utilisons le vocabulaire suivant dans ce manuscrit :

- Covariance : La covariance $\Sigma_{\mathbf{nn}} = E_{\mathbf{n}}\left[\overrightarrow{\overline{n}m}.\overrightarrow{\overline{n}m}^t\right]$
- Variance totale : La trace de la covariance, i.e. l'espérance de la distance carré : $Tr(\Sigma_{\mathbf{nn}}) = E\left[d(\overline{n},n)^2\right]$
- Variance : La variance par direction, qui correspond au paramètre $\sigma_{\mathbf{n}}^2$ pour une covariance isotrope : $\Sigma_{\mathbf{nn}} = \sigma_{\mathbf{n}}^2.Id$. La variance totale est alors donnée par $Tr(\Sigma_{\mathbf{nn}}) = \mathfrak{D}.\sigma_{\mathbf{n}}^2$
- Écart-type : La racine carré de la variance $\sigma_{\mathbf{n}} = \sqrt{\sigma_{\mathbf{n}}^2}$
- Matrice de dispersion : La matrice $\Delta_{\mathbf{nn}}$ qui apparaît dans une loi gaussienne.
- Dispersion totale : La trace de la matrice de dispersion $Tr(\Delta_{\mathbf{nn}})$.
- Dispersion : La dispersion par direction, qui correspond au paramètre $\delta_{\mathbf{n}}^2$ pour une matrice de dispersion isotrope : $\Delta_{\mathbf{nn}} = \delta_{\mathbf{n}}^2.Id$. La dispersion totale est alors donnée par $Tr(\Delta_{\mathbf{nn}}) = \mathcal{D}.\delta_{\mathbf{n}}^2$
- Dispersion-type : La racine carré de la dispersion $\delta_{\mathbf{n}} = \sqrt{\delta_{\mathbf{n}}^2}$

Lien entre variance et dispersion pour les normales 3D gaussiennes isotrope
Nous allons ici illustrer le lien entre covariance et matrice de dispersion pour des normales

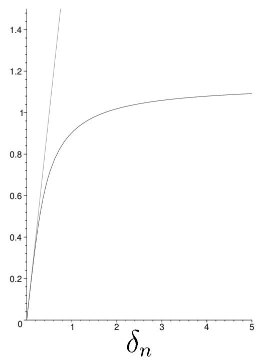
qui suivent une loi gaussienne isotrope. Il s'agit des lois pour lesquelles la matrice de dispersion est proportionnelle à l'identité : $\Delta_{\mathbf{nn}} = \delta_{\mathbf{n}}^2.Id$. Leur loi est donc de la forme :

$$p(\mathtt{m}) = \frac{1}{k}\exp\left(-\frac{(\widehat{\overline{\mathtt{n}},\mathtt{m}})^2}{2.\delta_{\mathbf{n}}^2}\right)$$

Nous pouvons calculer la variance totale par la formule suivante (au dénominateur : formule analogue à l'équation 7.5, au numérateur : la constante de normalisation) :

$$Tr(\Sigma_{\mathbf{nn}}) = E\left[\theta^2\right] = \frac{\int_{\theta=0}^{\frac{\pi}{2}}\int_{\alpha=-\pi}^{\pi}\ \theta^2\sin\theta\exp\left(-\frac{\theta^2}{2.\delta_{\mathbf{n}}^2}\right)\,d\theta\,d\alpha}{\int_{\theta=0}^{\frac{\pi}{2}}\int_{\alpha=-\pi}^{\pi}\ \sin\theta\exp\left(-\frac{\theta^2}{2.\delta_{\mathbf{n}}^2}\right)\,d\theta\,d\alpha}$$

Nous ne donnons pas ici les formules explicites (obtenues avec un logiciel de calcul symbolique), trop complexes, mais avons tracé la variance totale $Tr(\Sigma_{\mathbf{nn}})$ en fonction de la dispersion $\delta_{\mathbf{n}}^2$:



On constate alors que $Tr(\Sigma_{\mathbf{nn}})$ et $Tr(\Delta_{\mathbf{nn}}) = 2\delta_{\mathbf{n}}^2$ sont bien équivalents au voisinage de 0, comme le prévoyait l'eq. 7.7. En revanche, la variance totale tend vers une valeur finie lorsque la dispersion tend vers l'infini, i.e. lorsque la loi de probabilité tend vers une distribution uniforme. On vérifie que la limite est $\pi - 2$, la variance totale d'une distribution uniforme, que nous avions déjà calculée (eq. 7.5).

Algorithme : Tirage d'une normales 3D gaussiennes isotrope D'après ce qui précède, il faut, pour tirer une normale 3D gaussienne centrée autour de $\overline{\mathtt{o}}$ et de dispersion Δ, tirer le vecteur $\overrightarrow{on}$ de l'espace tangent suivant une loi gaussienne de covariance Δ (voir eq. 7.6) et calculer $\mathtt{n} = \exp_{\overline{\mathtt{o}}}(\overrightarrow{on}) = R_{\overline{\mathtt{o}}\wedge\overrightarrow{on}}(\overline{\mathtt{o}})$. Dans le cas d'une loi isotrope avec $\Delta = \delta^2.Id$, on peut encore simplifier les choses. En effet, dans ce cas $v = \overline{\mathtt{o}} \wedge \overrightarrow{on}$ suit aussi une loi gaussienne isotrope de variance δ^2 dans l'espace tangent. On peut donc tirer directement v, sans passer par $\overrightarrow{on}$. Pour cela, on tire un vecteur aléatoire dans l'espace 3D, et on le projette dans le plan tangent pour éliminer la composante perpendiculaire à celui-ci. L'algorithme est donc le suivant :

Algorithme Tirage d'une normale aléatoire gaussienne isotrope

- Tirer un vecteur aléatoire a suivant une gaussienne centrée isotrope d'écart-type δ^2.
- Le projeter dans l'espace tangent : $v = a - \overline{\mathtt{o}}.(a|\overline{\mathtt{o}})$.
- Calculer la normale : $\mathtt{n} = R_v(\overline{\mathtt{o}})$

7.2.9 Distance de Mahalanobis

La distance de Mahalanobis simple entre une normale $\mathtt{m}$ et une normale aléatoire centrée en $\overline{\mathtt{n}}$ et de covariance $\Sigma_{\mathbf{nn}}$ est donnée par :

$$\mu^2(\mathbf{n},\mathtt{m}) = \overrightarrow{\overline{n}m}^t.\Sigma_{\mathbf{nn}}^{-1}.\overrightarrow{\overline{n}m}$$

dans le cas d'une loi gaussienne isotrope de covariance $\Sigma_{\mathbf{nn}} = \sigma_{\mathbf{n}}^2.Id$, on retrouve :

$$\mu^2(\mathbf{n},\mathtt{m}) = \frac{d^2(\overline{\mathtt{n}},\mathtt{m})}{\sigma_{\mathbf{n}}^2}$$

7.3 Statistiques approchées sur les normales aléatoires

Les notions statistiques rigoureuses que nous venons de définir se révèlent assez peu maniables dans le cas des normales : l'espérance ne peut-être calculée que de façon itérative, il n'est pas évident de savoir lorsqu'elle n'est pas unique, et la matrice de covariance est définie dans l'espace tangent à l'espérance, qui est variable.

Il a donc semblé utile de justifier ou créer des méthodes de calcul approchées mais plus maniables de l'espérance et la covariance d'une primitive. Ces méthodes sont basées sur l'utilisation d'une approximation de la distance intrinsèque dans les formules développées dans la partie précédente. Précisons que nous ne modifierons pas la formule donnant l'espérance d'une observable (nous utiliserons toujours la métrique invariante, même si nous n'utilisons plus la distance associée).

Nous présenterons d'abord un cadre général à ce genre d'approximation, permettant d'apprécier leurs qualités. Nous verrons ensuite rapidement le cas de la distance vectorielle, qui permet de justifier l'approximation la plus simple de l'espérance au sens de Fréchet d'une normale aléatoire. Nous définirons ensuite le tenseur d'une normale aléatoire et verrons comment l'utiliser pour retrouver l'approximation de l'espérance au sens de Fréchet et de la covariance relative à la distance tensorielle. Nous comparerons enfin ces différentes méthodes.

7.3.1 Principe

Nous définirons par la suite une distance entre normales notée d_a et reliée à la distance intrinsèque d par une formule du type :

$$d_a = \Psi_a(d) \tag{7.8}$$

Pour être, comme d, une distance, d_a doit être :

- symétrique : ce sera toujours le cas ici car d est symétrique.
- définie positive : d étant elle-même définie positive, il faut et il suffit que Ψ_a soit définie positive $((d > 0) \Rightarrow (\Psi_a(d) > 0)$ et $\Psi_a(0) = 0)$.
- conforme à l'inégalité triangulaire : il suffit que Ψ_a soit croissante et de dérivé décroissante (et vérifie donc $\Psi_a(x + y) \leq \Psi_a(x) + \Psi_a(y)$). On a alors $d_a(\mathtt{n}_1,\mathtt{n}_2) = \Psi_a(d(\mathtt{n}_1,\mathtt{n}_2)) \leq \Psi_a(d(\mathtt{n}_1,\mathtt{n}_3) + d(\mathtt{n}_3,\mathtt{n}_2)) \leq \Psi_a(d(\mathtt{n}_1,\mathtt{n}_3)) + \Psi_a(d(\mathtt{n}_3,\mathtt{n}_2)) = d_a(\mathtt{n}_1,\mathtt{n}_3) + d_a(\mathtt{n}_3,\mathtt{n}_2)$. On remarquera que dans ce cas, notre distance approchée est une M-estimée [Rousseeuw and Leroy, 1987] de la distance intrinsèque, et est donc censée être plus robuste.

On supposera en général la dérivé de Ψ_a égale à 1 en 0 : $\frac{\partial \Psi_a}{\partial d}(0) = 1$, pour que les deux distances d et d_a soient directement comparables au voisinage de 0.

La distance d étant invariante, la distance d_a sera elle aussi invariante. En revanche, elle n'est reliée à aucune métrique Riemanienne invariante.

Nous utiliserons par la suite aussi la fonction ψ_a définie comme le rapport entre la distance approchée et la distance rigoureuse. On a donc :

$$\psi_a(d) = \Psi_a(d)/d \qquad d_a = \psi_a(d).d$$

7.3.1.1 Approximation de l'espérance

L'espérance de Fréchet (resp. Karcher) d'une normale aléatoire $\mathbf{n}$ est définie comme l'ensemble des normales $\mathtt{m}$ minimisant globalement (resp. localement) la variance totale $E_{\mathbf{n}}\left[d^2(\mathtt{n},\mathtt{m})\right]$ (voir 7.2.7). Ces normales sont regroupées dans l'ensemble $\mathbb{E}[\mathbf{n}]$.

Nous approximerons donc l'espérance au sens de Fréchet (resp. Karcher) en prenant l'ensemble des normales $\mathtt{m}$ minimisant globalement (resp. localement) la variance totale approchée :

$$E_{\mathbf{n}}\left[d_a^2(\mathtt{n},\mathtt{m})\right] \tag{7.9}$$

et regrouperons ces normales dans l'ensemble $\mathbb{E}_a[\mathbf{n}]$.

7.3.1.2 Approximation de la covariance

La covariance d'une primitive aléatoire $\mathbf{x}$ par rapport à une primitive fixe $\mathtt{y}$ est définie par : $\Sigma_{\mathbf{nn}}(\mathtt{m}) = E_{\mathbf{n}}[\overrightarrow{\mathtt{nm}}.\overrightarrow{\mathtt{nm}}^t]$. Ainsi, elle vérifie $E_{\mathbf{n}}[d^2(\mathtt{n},\mathtt{m})] = Tr(\Sigma_{\mathbf{nn}}(\mathtt{m}))$.

Or la variance totale approchée peut se ré-écrire $E_{\mathbf{n}}[\Psi_a(d(\mathtt{n},\mathtt{m}))^2] = E_{\mathbf{n}}[\Psi_a(\|\overrightarrow{mn}\|)^2] = E_{\mathbf{n}}\left[\psi_a^2(\|\overrightarrow{mn}\|).\,\|\overrightarrow{mn}\|^2\right]$. Il semble alors assez naturel de généraliser la fonction Ψ_a au plan tangent en $\mathtt{m}$ en écrivant :

$$\Psi_a(\overrightarrow{mn}) = \overrightarrow{mn}.\frac{\Psi_a(\|\overrightarrow{mn}\|)}{\|\overrightarrow{mn}\|} = \overrightarrow{mn}.\psi_a(\|\overrightarrow{mn}\|) \tag{7.10}$$

On a alors $E_{\mathbf{n}}[d^2(\mathtt{n},\mathtt{m})] = E_{\mathbf{n}}[Tr\,(\Psi_a(\overrightarrow{mn}).\Psi_a(\overrightarrow{mn})^t)] = Tr\,(E_{\mathbf{n}}[\Psi_a(\overrightarrow{mn}).\Psi_a(\overrightarrow{mn})^t])$. On définit donc la matrice de covariance approchée de la façon suivante :

$$\Sigma_{a\,\mathbf{nn}}(\mathtt{m}) = E_{\mathbf{n}}\left[\Psi_a(\overrightarrow{mn}).\Psi_a(\overrightarrow{mn})^t\right]$$

$$\Sigma_{a\,\mathbf{nn}} = E_{\mathbf{n}}\left[\Psi_a(\overrightarrow{\overline{n}n}).\Psi_a(\overrightarrow{\overline{n}n})^t\right] \tag{7.11}$$

7.3.1.3 Caractérisation de l'espérance de Karcher

Nous avons montré (eq. D.19) que l'espérance $\overline{n}$ au sens de Karcher de la normale vérifiait :

$$E\left[\overrightarrow{\overline{n}n}\right] = 0$$

Nous pensons que ceci s'adapte pour notre approximation (la démonstration doit être analogue[21], bien que certaines difficultés pourraient apparaître au niveau du lieu de coupure). L'approximation de l'espérance $\overline{n}$ au sens de Karcher de la normale vérifierait alors :

$$E\left[\dot{\Psi}_a(\left\|\overrightarrow{\overline{n}n}\right\|).\Psi_a(\overrightarrow{\overline{n}n})\right] = 0 \tag{7.12}$$

21. Il nous faut la dérivée, dans la carte locale, de la distance $d_a^2(\mathtt{n},\mathtt{m}) = \Psi_a(\|\overrightarrow{mn}\|)^2$:

$$\frac{\partial \Psi_a(\|\overrightarrow{mn}\|)^2}{\partial \overrightarrow{n}} = 2.\Psi_a(\|\overrightarrow{mn}\|).\frac{\partial \Psi_a(\|\overrightarrow{mn}\|)}{\partial \overrightarrow{n}} = 2.\Psi_a(\|\overrightarrow{mn}\|).\dot{\Psi}_a(\|\overrightarrow{mn}\|)\frac{\partial\,\|\overrightarrow{mn}\|}{\partial \overrightarrow{n}}$$

$$= 2.\Psi_a(\|\overrightarrow{mn}\|).\dot{\Psi}_a(\|\overrightarrow{mn}\|).[G_{\exp\,\overrightarrow{y}\bullet\bullet}].\frac{\overrightarrow{mn}}{\|\overrightarrow{mn}\|} = 2.\dot{\Psi}_a(\|\overrightarrow{mn}\|).[G_{\exp\,\overrightarrow{y}\bullet\bullet}].\Psi_a(\overrightarrow{mn})$$

La suite du raisonnement est alors identique.

7.3.1.4 Propagation des approximations

Avec des distances de ce type, les propriétés de propagation par une transformation fixe de l'espérance et de la covariance sont conservées :

$$\mathbb{E}_a[\mathtt{f} \star \mathbf{n}] = \mathtt{f} \star \mathbb{E}_a[\mathbf{n}]$$

$$\Sigma_{\mathtt{f}\star\mathbf{n}\,\mathtt{f}\star\mathbf{n}} = J(\mathtt{f}).\Sigma_{\mathbf{nn}}.J(\mathtt{f})^t$$

7.3.1.5 Approximation de l'espérance d'une gaussienne

Nous allons essayer de montrer que l'approximation de l'espérance se révèle être exacte pour une loi gaussienne.

Loi symétrique par rapport à une normale Soit une normale aléatoire $\mathbf{n}$ définie par une densité de probabilité $p_{\mathbf{n}}(\mathtt{n})$. Nous dirons qu'elle suit une loi symétrique décroissante s'il existe une normale $\widetilde{\mathtt{n}}$ telle que pour tout $\mathtt{m}$ on ait : $p_{\mathbf{n}}(\exp_{\widetilde{\mathtt{n}}} \overrightarrow{\widetilde{\mathtt{n}}m}) = p_{\mathbf{n}}(-\exp_{\widetilde{\mathtt{n}}} \overrightarrow{\widetilde{\mathtt{n}}m})$, i.e. si la loi est invariante par une symétrie d'axe $\widetilde{\mathtt{n}}$.

Or la symétrie de la loi de $\mathbf{n}$ implique que $E\left[\dot{\Psi}_a(\left\|\overrightarrow{\widetilde{n}n}\right\|).\Psi_a(\overrightarrow{\widetilde{n}n})\right] = E\left[-\dot{\Psi}_a(\left\|\overrightarrow{\widetilde{n}n}\right\|).\Psi_a(\overrightarrow{\widetilde{n}n})\right] = -E\left[\dot{\Psi}_a(\left\|\overrightarrow{\widetilde{n}n}\right\|).\Psi_a(\overrightarrow{\widetilde{n}n})\right]$ et donc :

$$E\left[\dot{\Psi}_a(\left\|\overrightarrow{\widetilde{n}n}\right\|).\Psi_a(\overrightarrow{\widetilde{n}n})\right] = 0$$

La normale $\widetilde{\mathtt{n}}$ a donc des chances de faire partie de l'espérance de Karcher voire de l'espérance de Fréchet. Il faut cependant vérifier qu'il s'agit bien d'un minimum local ou global (et non d'un maximum par exemple) de la variance.

Loi symétrique mono-modale Nous pensons que, lorsque la loi est mono-modale et symétrique, le mode, qui vérifie $E\left[\dot{\Psi}_a(\left\|\overrightarrow{\widetilde{n}n}\right\|).\Psi_a(\overrightarrow{\widetilde{n}n})\right] = 0$, est nécessairement le minimum de la variance et de la variance approchée.

Conclusion Lorsque la loi est symétrique mono-modale (ce qui est en particulier le cas des gaussiennes), on a donc vraisemblablement l'équivalence entre les différentes approximations de l'espérance de Fréchet et sa véritable valeur.

7.3.1.6 Cas général

Dans le cas général (loi de probabilité quelconque), on peut donner un majorant de l'erreur commise en utilisant nos statistiques approchées. Il faut pour cela faire appel à des travaux mathématiques plus généraux.

Nous nous appuierons ici sur l'article [Picard, 1994]. Cet article est basé sur le concept de connecteur, qui généralise les cartes locales définies dans le plan tangent : en résumé, un connecteur est une fonction γ qui à un couple $(\mathtt{x},\mathtt{y})$ de primitives associe un vecteur de l'espace tangent $\overrightarrow{xy}$. Il peut s'agir de la fonction log (on parle alors de connecteur géodésique), d'une fonction convexe sur les géodésiques (on parle alors de connecteur convexe), ou de fonctions plus générales. Dans notre cas, $\gamma(\mathtt{m},\mathtt{n}) = \Psi_a(\overrightarrow{mn})$ définit effectivement un connecteur convexe (car la fonction Ψ_a est elle-même convexe). L'article définit la notion de γ-barycentre d'une primitive aléatoire (partie 1), qui est en fait équivalente à notre espérance de Fréchet au sens de la distance d_a. Il montre ensuite (partie 2) que l'erreur commise sur le γ-barycentre est de la forme $O(\sigma)$, i.e. inférieure à $k\sigma$, où σ est l'écart-type de la primitive aléatoire.

En résumé, l'erreur commise sur l'estimation de l'espérance de Fréchet d'une normale aléatoire par les méthodes de statistiques approchées est de l'ordre de σ où σ est l'écart-type de la normale aléatoire.

7.3.2 Utilisation de la distance vectorielle

Nous allons voir ici quels sont les effets de l'utilisation de la distance vectorielle entre deux normales définie comme la distance au sens vectoriel entre les deux représentations unitaires des normales. Il s'agit donc, pour deux normales $\mathtt{m}$ et $\mathtt{n}$ de :

$$d_v(\mathtt{m},\mathtt{n}) = \|\mathtt{m} - \mathtt{n}\| = 2\sin(\frac{(\widehat{\mathtt{m},\mathtt{n}})}{2}) = d(\mathtt{m},\mathtt{n}) + O(d(\mathtt{m},\mathtt{n})^3)$$

où $\|\|$ est la norme quadratique usuelle de $\mathbb{R}^{\mathfrak{D}}$. Cette distance vérifie aussi :

$$d_v(\mathtt{m},\mathtt{n})^2 = \|\mathtt{m} - \mathtt{n}\|^2 = 2(1 - (\mathtt{m}|\mathtt{n}))$$

où $(.|.)$ est la produit scalaire usuel de $\mathbb{R}^{\mathfrak{D}}$.

On a donc :

$$\Psi_v(d) = 2\sin(\frac{d}{2}) \qquad \psi_v(d) = \frac{\sin(\frac{d}{2})}{\frac{d}{2}}$$

Remarquons au passage que cette distance est bien adaptée au cas orienté.

7.3.2.1 Calcul de l'approximation vectorielle de l'espérance

Nous allons effectuer ce calcul dans le cas discret, et retrouver ainsi une technique souvent utilisée pour calculer la moyenne de plusieurs normales. Soit donc un ensemble de normales $\mathtt{n}_i$ représentées par les vecteurs unitaires de coordonnées $\mathtt{n}_i^k$ et munies de leur probabilité $P(\mathtt{n}_i)$. L'approximation vectorielle de l'espérance est l'ensemble des normales

$\mathtt{m}$ représentées par les vecteurs unitaires de coordonnées $\mathtt{m}^k$ minimisant la variance totale :

$$\sum_i P(\mathtt{n}_i).\, \|\mathtt{m}-\mathtt{n}_i\|^2_{\mathbb{R}^{\mathfrak{D}}} = \sum_i P(\mathtt{n}_i).(\mathtt{m}-\mathtt{n}_i|\mathtt{m}-\mathtt{n}_i)$$

En utilisant un multiplicateur de Lagrange α pour prendre en compte la contrainte $\|\mathtt{m}\| = 1$, on a donc :

$$\mathtt{m} = \arg\min_{\mathtt{m},\alpha} \frac{\left[\sum_i P(\mathtt{n}_i).(\mathtt{m}-\mathtt{n}_i|\mathtt{m}-\mathtt{n}_i) + \alpha.(\|\mathtt{m}\|^2 - 1)\right]}{2}$$

En dérivant par rapport à $\mathtt{m}$, on obtient : $\sum_i P(\mathtt{n}_i).(\mathtt{m}-\mathtt{n}_i)+\alpha.\mathtt{m} = 0$ et donc : $(1+\alpha)\mathtt{m} = \sum_i P(\mathtt{n}_i).\mathtt{n}_i$. On calcule alors la valeur de α grâce à la contrainte $\|\mathtt{m}\| = 1$. Finalement :

$$\mathtt{m} = \frac{\sum_i P(\mathtt{n}_i).\mathtt{n}_i}{\|\sum_i P(\mathtt{n}_i).\mathtt{n}_i\|}$$

On a donc la proposition suivante :

Proposition 7.1 *Espérance de Fréchet au sens de la distance vectorielle*
On obtient l'approximation vectorielle de l'espérance au sens de Fréchet d'une normale aléatoire en normalisant la moyenne des normales considérées comme de simples vecteurs de $\mathbb{R}^{\mathfrak{D}}$.

Ce procédé, souvent utilisé, est donc une approximation de la véritable espérance d'une normale aléatoire. Il échoue cependant lorsque $\sum_i P(\mathtt{n}_i).\mathtt{n}_i = 0$.

7.3.2.2 Approximation de la matrice de covariance

L'approximation de la covariance ne présente ici aucun intérêt pratique. Nous ne détaillerons pas les calculs.

7.3.3 Tenseur d'une normale aléatoire

La distance tensorielle et l'approximation de l'espérance et la covariance qu'elle permet de définir est étroitement reliée au tenseur d'une normale aléatoire (d'où son nom). Nous allons donc dans un premier temps nous familiariser avec ce tenseur, avant de revenir, à la section suivante à nos approximations.

Ce tenseur peut être défini comme la covariance de la normale aléatoire, calculée comme pour un vecteur aléatoire de l'espace nD classique, i.e. en oubliant qu'il s'agit d'un vecteur unitaire. Cet oubli est, inutile de le préciser, lourd de conséquence, et nous ne ferons plus allusion à cette matrice de covariance dans la suite, pour éviter les confusions avec la vraie matrice de covariance des normales, définie dans l'espace tangent à la moyenne.

7.3.3.1 Tenseur associé à une direction

Plaçons nous dans $\mathbb{R}^{\mathfrak{D}}$ muni de sa norme canonique. On peut associer à tout vecteur le tenseur suivant :

Définition 7.1 *Tenseur associé à un vecteur de* $\mathbb{R}^{\mathfrak{D}}$

Soit $x \in \mathbb{R}^{\mathfrak{D}}$.

Le tenseur associé à x *et noté* $T(x)$ *est le tenseur d'ordre 2 donné par :*

$$T(x) = x \otimes x$$

soit, sous forme matricielle :

$$T(x) = x.x^t$$

$$(T(x))_{ij} = x_i.x_j \qquad (T(x))^i_j = x^i.x_j \qquad \ldots$$

On remarque que $(y \otimes y = x \otimes x) \Leftrightarrow (y = \pm x)$. Autrement dit, l'utilisation des tenseurs nous fait perdre l'information sur le sens de l'orientation. Il est alors naturel de définir la relation d'équivalence suivante : $x \sim y \Leftrightarrow (x = \pm y)$, et la définition du tenseur reste valable dans l'espace non orienté $\mathbb{R}^{\mathfrak{D}}/\sim$. Elle y est même plus intéressante, car l'application $x \mapsto T(x)$ est alors injective.

On peut de plus restreindre cette définition à l'ensemble des normales orientées de $\mathbb{R}^{\mathfrak{D}}$ qui forment le sous-ensemble $\mathcal{S}_{\mathfrak{D}-1}$ identifié ensuite à la sphère unité $\{n \in \mathbb{R}^{\mathfrak{D}} / \|n\| = 1\}$, ou l'ensemble des directions (i.e. des normales non-orientées) de $\mathbb{R}^{\mathfrak{D}}$, qui forment l'ensemble $\mathcal{P}_{\mathfrak{D}-1} = \mathcal{S}_{\mathfrak{D}-1}/\sim$. Nous présentons le cas non-orienté dans la suite, mais ces définitions restent bien-sûr valable (mais ne sont plus biunivoque) dans le cas orienté.

Définition 7.2 *Tenseur associé à une normale non-orientée*

Soit une direction de $\mathcal{P}_{\mathfrak{D}-1}$, *représentée par le vecteur unitaire non orienté* $\pm n$.

Le tenseur associé à n *et noté* $T(n)$ *est le tenseur d'ordre 2 donné par :*

$$T(n) = n \otimes n$$

soit, sous forme matricielle :

$$T(n) = n.n^t$$

$$(T(n))_{ij} = n_i.n_j \qquad (T(n))^i_j = n^i.n_j \qquad \ldots$$

7.3.3.2 Tenseur associé à une direction aléatoire

Soit maintenant une direction aléatoire $\mathbf{n}$. On lui associe le tenseur aléatoire $T(\mathbf{n})$. Or l'ensemble des tenseurs d'ordre 2 forment un espace vectoriel réel de dimension finie. Le tenseur aléatoire $T(\mathbf{n})$ est donc un vecteur aléatoire, dont nous pouvons calculer l'espérance (en passant par exemple par les coordonnées de ce tenseur dans une base qui sont des variables aléatoires réelles). Nous utiliserons cette espérance pour représenter $\mathbf{n}$:

Définition 7.3 *Tenseur associé à une élément aléatoire*
Soit $\mathbf{n}$ une direction aléatoire de $\mathcal{P}_{\mathfrak{D}-1}$.
Le tenseur associé à $\mathbf{n}$ et noté $\overline{T(\mathbf{n})}$ est le tenseur d'ordre 2 donné par :

$$\overline{T(\mathbf{n})} = E\left[T(\mathbf{n})\right] = E\left[\mathbf{n} \otimes \mathbf{n}\right]$$

soit sous forme matricielle :

$$\overline{T(\mathbf{n})} = E\left[\mathbf{n}\mathbf{n}^t\right]$$

7.3.3.3 Propriétés du tenseur associé à une direction aléatoire

La première propriété importante d'un tenseur de la forme $\overline{T(\mathbf{n})}$ est la forme de son application à deux vecteurs. On a en effet, par linéarité : $a\overline{\otimes}\overline{T(\mathbf{n})}\overline{\otimes}b = E\left[a\overline{\otimes}(\mathbf{n} \otimes \mathbf{n})\overline{\otimes}b\right]$. Or $a\overline{\otimes}(n \otimes n)\overline{\otimes}b = (a\overline{\otimes}n) \otimes (n\overline{\otimes}b) = (a|n)(n|b)$. Finalement :

Proposition 7.2 *Application de $\overline{T(\mathbf{n})}$ à deux vecteurs*
Soient $\overline{T(\mathbf{n})}$ tenseur associé à une direction aléatoire de $\mathcal{P}_{\mathfrak{D}-1}$ et a et b deux vecteurs de $\mathbb{R}^{\mathfrak{D}}$. On a :

$$a\overline{\otimes}\overline{T(\mathbf{n})}\overline{\otimes}b = E\left[(a|\mathbf{n}).(\mathbf{n}|b)\right]$$

$$a\overline{\otimes}\overline{T(\mathbf{n})}\overline{\otimes}a = E\left[(a|\mathbf{n})^2\right] \geq 0$$

On voit alors immédiatement que $\overline{T(\mathbf{n})}$ est un tenseur symétrique. Or il existe un important résultat sur les tenseurs symétriques réels d'ordre 2 : un tenseur T symétrique réel d'ordre 2 sur un espace euclidien E de dimension $\mathfrak{D}$ possède exactement $\mathfrak{D}$ valeurs propres réelles λ_1, λ_2, ... , $\lambda_{\mathfrak{D}}$ (distinctes ou non). Les espaces propres étant orthogonaux entre-eux, il peut donc être diagonalisé dans une base orthonormale $\{v_i\}$ constituée de vecteurs propres.

De plus, on constate que $\overline{T(\mathbf{n})}$ est positif. Ses valeurs propres sont donc positives, et nous les mettrons sous forme carré : $\lambda_i = \alpha_i^2$.

En résumé :

Proposition 7.3 *Diagonalisation de $\overline{T(\mathbf{n})}$*
Soient $\overline{T(\mathbf{n})}$ tenseur associé à une direction aléatoire de $\mathcal{P}_{\mathfrak{D}-1}$.
$\overline{T(\mathbf{n})}$ est diagonalisable de valeurs propres α_1^2, α_2^2, ..., $\alpha_{\mathfrak{D}}^2$, avec $\alpha_i \in \mathbb{R}^+$.

On peut de plus construire une base orthonormale $\{v_i\}$ de $\mathbb{R}^{\mathfrak{D}}$, où chaque v_i est un vecteur propre associé à la valeur propre α_i^2. Une telle base est appelée base propre de $\overline{T(\mathbf{n})}$.

L'expression de $\overline{T(\mathbf{n})}$ dans cette base est : $\overline{T(\mathbf{n})} = \sum_i \alpha_i^2 (v_i \otimes v_i) = \sum_i (\alpha_i v_i) \otimes (\alpha_i v_i)$.

Une dernière propriété importante de $\overline{T(\mathbf{n})}$ porte sur sa trace (égale à la somme des valeurs propres) : par linéarité, $Tr(\overline{T(\mathbf{n})}) = E\left[Tr(\mathbf{n} \otimes \mathbf{n})\right]$. Or $Tr(n \otimes n) = n \overline{\otimes} n = (n|n) = 1$. On a donc :

Proposition 7.4 *Trace de $\overline{T(\mathbf{n})}$*

Soient $\overline{T(\mathbf{n})}$ tenseur associé à une direction aléatoire de $\mathcal{P}_{\mathfrak{D}-1}$. La trace de $\overline{T(\mathbf{n})}$, qui est aussi la somme de ses valeurs propres, est unitaire :

$$Tr(\overline{T(\mathbf{n})}) = \sum_i \alpha_i^2 = 1$$

7.3.3.4 Représentation sous forme d'ellipsoïdes

Il est commode de représenter les tenseurs symétriques positifs par des ellipsoïdes. En effet, une fois notre tenseur diagonalisé $\overline{T(\mathbf{n})} = \sum_i \alpha_i^2 (v_i \otimes v_i)$, on peut lui associer de façon biunivoque l'ellipsoïde dont les axes principaux sont colinéaires à v_i et de demi-longueur α_i.

Dans le cas où les valeurs propres sont toutes non nulles, il s'agit donc de l'ellipsoïde d'équation $\sum \frac{(x|v_i)^2}{\alpha_i^2} = 1$. Le tenseur est alors inversible, d'inverse $\overline{T(\mathbf{n})}^{-1} = \sum_i \frac{1}{\alpha_i^2} (v_i \otimes v_i)$, si bien que l'équation de notre ellipsoïde n'est autre que :

$$x \overline{\otimes} (\overline{T(\mathbf{n})}^{-1}) \overline{\otimes} x = Cte$$

Dans le cas où au moins l'une de nos valeurs propres est nulle, il s'agit d'un ellipsoïde dégénéré. Nous donnerons des exemples de tels tenseurs en présentant les tenseurs bâton et plateau de Médioni (voir pages suivantes).

Prenons un exemple en 2D. Soit le tenseur de matrice $\begin{bmatrix} 1/2 & 1/4 \\ 1/4 & 1/2 \end{bmatrix}$ dans la base canonique. Ses valeurs propres sont 1/4 et 3/4, et les vecteurs propres unitaires associés sont par exemple respectivement $\begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}$ et $\begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$. Ils forment bien une base orthonormale, et le tenseur est alors représenté par l'ellipse de la figure 7.2 (les vecteurs propres sont représentés par les flèches).

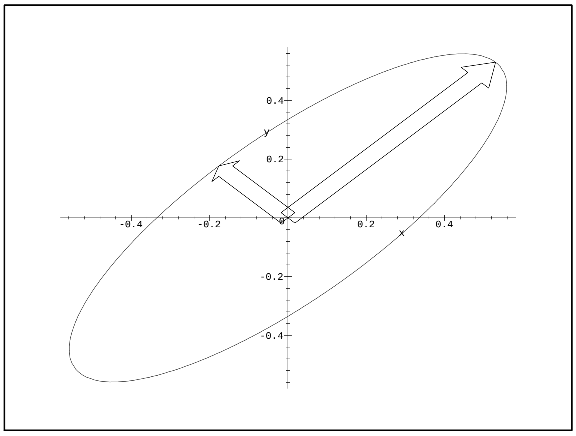


FIG. 7.2 – *Exemple de représentation sous forme d'ellipsoïde du tenseur d'une normale aléatoire*

7.3.3.5 Exemples de tenseurs associés à une direction aléatoire

Le principal intérêt des tenseurs utilisés ici est l'effet tout-à-fait particulier que le passage à l'espérance a sur eux. Avant de le traiter de manière rigoureuse, nous allons essayer d'illustrer cet effet en prenant quelques exemples de normales aléatoires simples, où l'espérance d'une fonction réelle est simple à définir. Commençons par examiner le cas d'une loi binomiale en 2D. Soient donc deux normales parfaitement déterminées. En notant $\{e_1,e_2\}$ la base canonique de $\mathbb{R}^2$, on prend par exemple les normales représentées par $n_1 = \sin(-\theta)e_1 + \cos(-\theta)e_2$ et $n_2 = \sin(\theta)e_1 + \cos(\theta)e_2$. Les tenseurs associés ont pour matrice dans la base canonique :

$$[T(n_1)] = \begin{bmatrix} \sin^2\theta & -\sin\theta\cos\theta \\ -\sin\theta\cos\theta & \cos^2\theta \end{bmatrix} \quad et \quad [T(n_2)] = \begin{bmatrix} \sin^2\theta & \sin\theta\cos\theta \\ \sin\theta\cos\theta & \cos^2\theta \end{bmatrix}$$

Le tenseur $T(n_1)$ a pour valeurs propres 1 et 0, et les vecteurs propres associés sont n_1 et $n_{1\perp}$, vecteur perpendiculaire à n_1. Il en va de même pour $T(n_2)$. $T(n_1)$ et $T(n_2)$ sont donc associés aux ellipses dé-générées de la figure 7.3 gauche (On a ici pris $\theta = \frac{\pi}{12}$, et on a exagéré la largeur du petit axe, qui est en principe nulle). Le tenseur associé à la normale aléatoire $\mathbf{n}$ qui suit une loi binomiale de la forme $Pr(\mathbf{n} = n_1) = Pr(\mathbf{n} = n_2) = 1/2$, est tout simplement la moyenne des deux tenseurs qui précèdent. Sa matrice est donc :

$$\left[\overline{T(\mathbf{n})}\right] = \left[\frac{T(n_1)}{2} + \frac{T(n_2)}{2}\right] = \begin{bmatrix} \sin^2\theta & 0 \\ 0 & \cos^2\theta \end{bmatrix}$$

Les vecteurs propres et valeurs propres apparaissent immédiatement, l'ellipse associée est représenté sur la figure 7.3 droite.

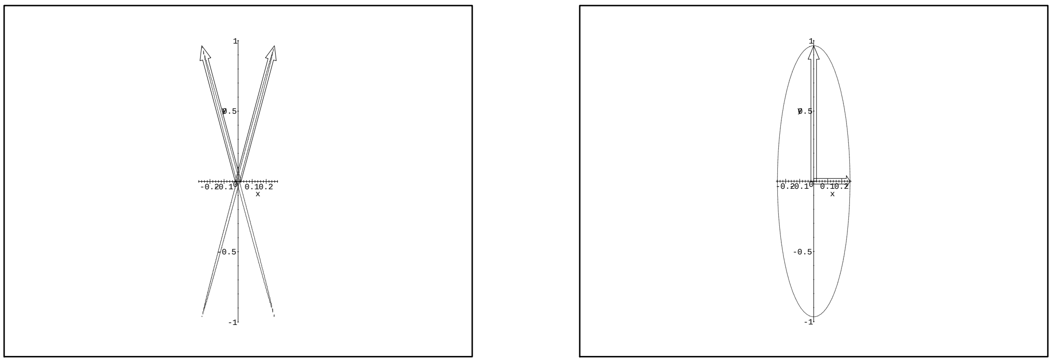


FIG. 7.3 – *Tenseur d'une normale aléatoire suivant une loi binomiale*

On constate donc que l'aspect bimodal de $\mathbf{n}$ est voilé. En revanche, le grand-axe de l'ellipse (et donc la vecteur propre associée à la plus grande valeur propre) correspond à l'espérance de la normale telle qu'on se la figure intuitivement. De plus, le fait que le petit axe ne soit plus de longueur nulle semble exprimer le fait que les réalisations de la variable aléatoire sont dispersées autour d'une valeur. Sa longueur, qui augmente avec la dispersion (lorsque θ augmente dans notre exemple), suggère un lien avec la covariance.

Soit maintenant 4 normales en 3D. On utilise la base canonique $\{e_1,e_2,e_3\}$, et les 4 normales sont de la forme $n_1 = \sin(-\theta_1)e_1 + \cos(-\theta_1)e_3$, $n_2 = \sin(\theta_1)e_1 + \cos(\theta_1)e_3$, $n_3 = \sin(-\theta_2)e_2 + \cos(-\theta)e_3$ et $n_4 = \sin(\theta_2)e_2 + \cos(\theta_2)e_3$. Les tenseurs associés sont donnés par leurs matrices :

$$[T(n_1)] = \begin{bmatrix} \sin^2\theta_1 & 0 & -\sin\theta_1\cos\theta_1 \\ 0 & 0 & 0 \\ -\sin\theta_1\cos\theta_1 & 0 & \cos^2\theta_1 \end{bmatrix}$$

$$[T(n_2)] = \begin{bmatrix} \sin^2\theta_1 & 0 & \sin\theta_1\cos\theta_1 \\ 0 & 0 & 0 \\ \sin\theta_1\cos\theta_1 & 0 & \cos^2\theta_1 \end{bmatrix}$$

$$[T(n_3)] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sin^2\theta_2 & -\sin\theta_2\cos\theta_2 \\ 0 & -\sin\theta_2\cos\theta_2 & \cos^2\theta_2 \end{bmatrix}$$

$$[T(n_4)] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sin^2\theta_2 & \sin\theta_2\cos\theta_2 \\ 0 & \sin\theta_2\cos\theta_2 & \cos^2\theta_2 \end{bmatrix}$$

Les ellipsoïdes dé-générés sont représentés dans la figure 7.4 gauche.

Soit maintenant une normale aléatoire $\mathbf{n}$ définie par $Pr(\mathbf{n} = n_1) = Pr(\mathbf{n} = n_2) = Pr(\mathbf{n} = n_3) = Pr(\mathbf{n} = n_4) = 1/4$. Le tenseur associé est donc donné par la matrice :

$$\left[\overline{T(\mathbf{n})}\right] = \left[\frac{T(n_1)}{4} + \frac{T(n_2)}{4} + \frac{T(n_3)}{4} + \frac{T(n_4)}{4}\right] = \begin{bmatrix} \frac{\sin^2\theta_1}{2} & 0 & 0 \\ 0 & \frac{\sin^2\theta_2}{2} & 0 \\ 0 & 0 & \frac{\cos^2\theta_1 + \cos^2\theta_2}{2} \end{bmatrix}$$

Il est représenté par l'ellipsoïde de la figure 7.4 droite.

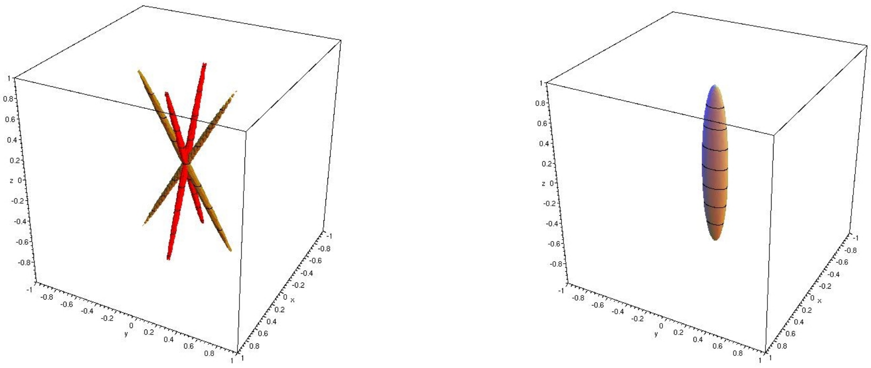


FIG. 7.4 – *Tenseur d'une normale aléatoire suivant une loi quadrimodale*

On constate qu'une fois de plus, les vecteurs propres associés à la plus grande valeur propre semble correspondre à la notion intuitive d'espérance. Les 2 autres vecteurs propres sont dans la plan perpendiculaire au premier, et les valeurs propres associées semblent aussi représenter une divergence (plus importante suivant l'axe y, dans notre cas). Ceci suggère un lien entre la sous-matrice de $\overline{T(\mathbf{n})}$ dans la base formée par les vecteurs propres obtenue après élimination de la première coordonnée (première ligne et première colonne donc) et une certaine matrice de covariance.

Tenseurs bâton (stick), plateau (plate) et boule (ball) Certains tenseurs particuliers peuvent retenir l'attention : il s'agit des tenseurs dont certaines valeurs propres sont nulles et dont les autres valeurs propres sont égales. Nous donnerons ici l'exemple en 3D, avec ce que l'équipe de Médioni a appelé les tenseurs Stick, Plate et Ball [Médioni *et al.*, 2000] (que nous traduirons ici par bâton, plateau et boule), mais on pourrait faire de même en n'importe quelle dimension.

Nous travaillerons dans la carte définie dans la partie 7.2.5. Nous considérerons en général la densité de probabilité sous la forme $p_{\mathbf{n}}(n(\theta,\alpha))$, et non dans la carte locale. On aura alors :

$$\overline{T(\mathbf{n})} = \int_{\theta=0}^{\frac{\pi}{2}} \int_{\alpha=-\pi}^{\pi} \begin{bmatrix} \cos\theta \\ \sin\theta\,\cos\alpha \\ \sin\theta\,\sin\alpha \end{bmatrix} \otimes \begin{bmatrix} \cos\theta \\ \sin\theta\,\cos\alpha \\ \sin\theta\,\sin\alpha \end{bmatrix} \; p_{\mathbf{n}}(\mathtt{n}(\theta,\alpha))\;\sin\theta\,d\theta\,d\alpha$$

La matrice de $\overline{T(\mathbf{n})}$ est alors :

$$\left[\overline{T(\mathbf{n})}\right] = \int_{0}^{\frac{\pi}{2}} \int_{-\pi}^{\pi} \begin{bmatrix} \cos^2\theta & \cos\theta\,\sin\theta\,\cos\alpha & \cos\theta\,\sin\theta\,\sin\alpha \\ ... & \sin^2\theta\,\cos^2\alpha & \sin^2\theta\,\cos\alpha\,\sin\alpha \\ ... & ... & \sin^2\theta\,\sin^2\alpha \end{bmatrix} \; p_{\mathbf{n}}(\mathtt{n}(\theta,\alpha))\;\sin\theta\,d\theta\,d\alpha$$

Tenseur associé à une normale déterministe - Tenseur bâton L'équipe de Médioni associe à une donnée du type normale à une surface le tenseur $n \otimes n$ (qui possède une valeur propre unitaire et les autres nulles). Ceci correspond donc à l'hypothèse que la normale est déterministe, ce qui est le cas. Nous avons représenté dans la figure 7.5 la surface et sa normale, puis le tenseur correspondant.

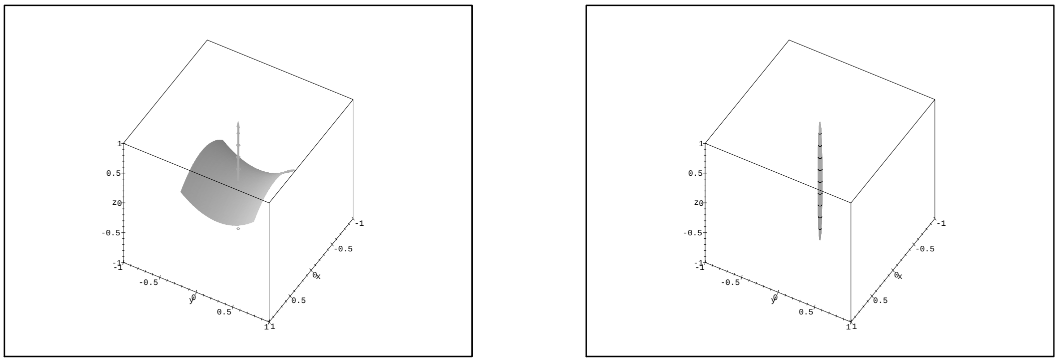


FIG. 7.5 – *Un élément de surface et sa normale (gauche) et le tenseur bâton associé à cette normale (droite)*

Tenseur associé à un élément de courbe - Tenseur plateau Médioni associe à un élément de courbe, défini par sa tangente t le tenseur $e_1 \otimes e_1 + e_2 \otimes e_2$ (qui possède deux valeurs propres égales et une dernière nulle), où (t,e_1,e_2) forme un repère orthonormal.

Or un élément de courbe déterministe nous apporte l'information suivante : la normale à la courbe est perpendiculaire à la tangente. Autrement dit, dans le repère (t,e_1,e_2), on a $\theta = \frac{\pi}{2}$, mais on ne dispose d'aucune information sur γ. L'hypothèse la plus faible est alors de supposer que γ suit une loi de probabilité uniforme sur $\{\frac{\pi}{2}\} \times [-\pi,\pi]$.

On a donc $p_{\mathbf{n}}(\mathtt{n}(\theta,\gamma)) = \delta_{\frac{\pi}{2}}(\theta)\,\frac{1}{2\pi}$, et en reportant, on trouve bien $\overline{T(\mathbf{n})} = \frac{e_1 \otimes e_1 + e_2 \otimes e_2}{2}$, soit le tenseur utilisé par Lee et Médioni après normalisation.

La figure 7.6 représente une courbe et sa tangente, puis le tenseur correspondant.

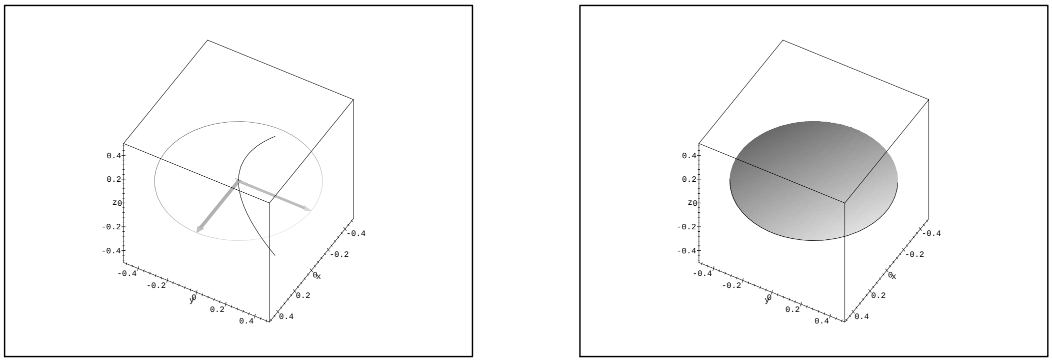


FIG. 7.6 – *Un élément de courbe et sa tangente (gauche) et le tenseur plateau associé à cette tangente (droite)*

Tenseur associé à un point - Tenseur boule Dans le cas d'un point, on ne dispose d'aucune information sur la normale. L'hypothèse la plus faible est alors de supposer qu'elle suit une loi de probabilité uniforme: $p_{\mathbf{n}}(m) = \frac{1}{2\pi}$. Le résultat, que l'on peut obtenir par calcul direct ou par simple considération de symétrie, est alors $\overline{T(\mathbf{n})} = \frac{e_1 \otimes e_1 + e_2 \otimes e_2 + e_3 \otimes e_3}{3}$ (qui possède trois valeurs propres égales) pour n'importe quelle base (e_1,e_2,e_3) orthonormée, qui est, au coefficient de normalisation prêt le tenseur utilisé par Lee et Médioni.

Le tenseur boule est représenté sur la figure 7.7.

Décomposition d'un tenseur associée à une normale aléatoire Nous avons déjà vu qu'un tenseur pouvait se décomposer suivant ses 3 valeurs propres: $\overline{T(\mathbf{n})} = \alpha_1^2(v_1 \otimes v_1) + \alpha_2^2(v_2 \otimes v_2) + \alpha_3^2(v_3 \otimes v_3)$. Mais on peux aussi regrouper ces termes pour obtenir des tenseurs bâton, plateau et boule. Si les valeurs propres sont rangées dans l'ordre décroissant: $\alpha_1 > \alpha_2 > \alpha_3$, on peux écrire:

$$\overline{T(\mathbf{n})} = (\alpha_1^2 - \alpha_2^2 - \alpha_3^2)(v_1 \otimes v_1) + (\alpha_2^2 - \alpha_3^2)(v_1 \otimes v_1 + v_2 \otimes v_2) + \alpha_3^2(v_1 \otimes v_1 + v_2 \otimes v_2 + v_3 \otimes v_3)$$

On obtient ainsi une combinaison d'un tenseur bâton (qui représente la normale la plus probable et la confiance par rapport aux autres possibilités), d'un tenseur plateau

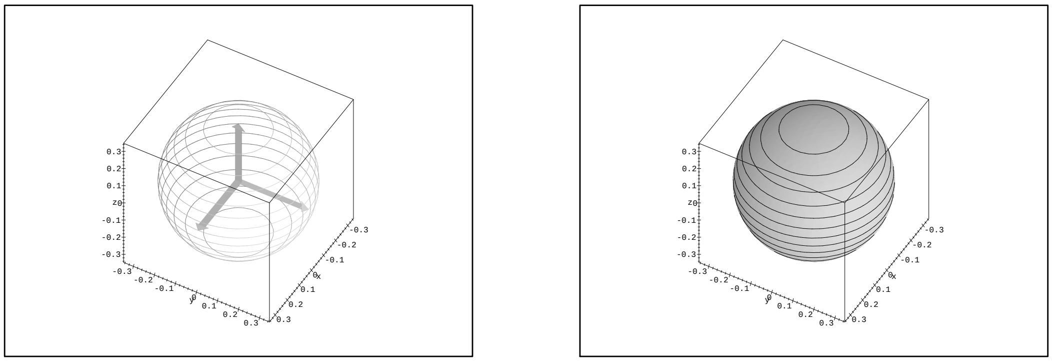


FIG. 7.7 – *Le tenseur boule*

(qui représente le plan dans lequel la normale est la plus probable, et la confiance par rapport aux autres), et d'un tenseur boule (qui ne représente pas grand chose, si ce n'est l'absence d'information sur l'orientation).

7.3.3.6 Tenseurs conditionnels

Nous rencontrerons par la suite des normales aléatoires dont la loi de probabilités est conditionnée par d'autres évènements aléatoires. De telles normales seront donc données par des lois de probabilités conditionnelles de la forme $P(n/...)$ et nous noterons $T(n/...)$ le tenseur conditionnel associé. Nous allons ici voir comment sont reliés ces tenseurs.

Le premier cas est celui d'une normale aléatoire pouvant provenir de plusieurs processus aléatoires a_i différents, représentés par les lois $P(n/a = a_i) = P(n_i)$ de tenseurs $\overline{T(\mathbf{n}_i)}$. La provenance est exprimée au travers d'une loi a-priori sur le processus aléatoire d'origine $P(a = a_i) = \pi_i$. On a alors par le théorème des probabilités totales :

$$P(n) = \sum_i P(n/a = a_i)P(a = a_i) = \sum P(n_i)\pi_i$$

le passage au tenseur étant linéaire par rapport à la loi de probabilité, on a immédiatement :

$$\overline{T(\mathbf{n})} = \sum \overline{T(\mathbf{n}/a = a_i)}P(a_i) = \sum \pi_i \overline{T(\mathbf{n}_i)} \tag{7.13}$$

on peut alors faire complètement abstraction des loi de probabilités et travailler directement sur les tenseurs.

Un deuxième cas est celui où la loi sur la normale dépend d'une deuxième normale, dont la loi $P(n_0)$ est partiellement connue au travers du tenseur $\overline{T(n_0)}$. La dépendance est exprimée par la loi de probabilité conditionnelle $P(n/n_0)$ et le tenseur conditionnel

$\overline{T(n/n_0)}$. on a alors :

$$P(n) = \int P(n/n_0)P(n_0)dn_0$$

on peut alors écrire, par linéarité :

$$\overline{T(n)} = \int \overline{T(n/n_0)}P(n_0)dn_0 \tag{7.14}$$

on pourrait alors croire que cette égalité implique, pour $T(n/n_0)$ donné, une relation linéaire entre les tenseurs $\overline{T(n)}$ et $\overline{T(n_0)}$[22]. Il n'en est rien, comme le montre l'exemple 2D de la figure 7.8, où deux normales aléatoires ayant le même tenseur $\overline{T(n_0)}$ définissent, via l'équation 7.14, des tenseurs $\overline{T(n)}$ très différents.

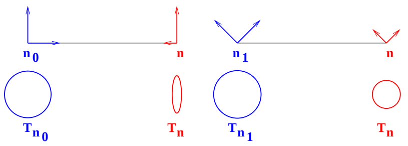


FIG. 7.8 – *Contre-exemple à la linéarité des votes de Médioni : prenons deux normales aléatoires* $\mathbf{n}_0$ *et* $\mathbf{n}_1$ *n'ayant que deux possibilités équiprobables, verticale et horizontale pour* $\mathbf{n}_0$*, à 45 degrés pour* $\mathbf{n}_1$*. Elles ont alors le même tenseur isotrope* $\overline{T(n_0)} = \overline{T(n_1)} = Id/2$*. Prenons maintenant un vote calqué sur le vote bâton de Médioni : la normale* n_0 *verticale donne une normale* n *verticale très probable (avec une faible covariance), la normale horizontale donne une normale horizontale très improbable (avec une forte covariance), les deux normales à 45 degrés donnent des normales symétriques équiprobables. On obtient alors pour* $\mathbf{n}$ *un tenseur indiquant une normale très probablement verticale en considérant* $\mathbf{n}_0$*, et isotrope en considérant* $\mathbf{n}_1$*.*

7.3.4 Utilisation de la distance tensorielle

Nous allons maintenant voir comment retrouver, à l'aide de notre tenseur, l'espérance et la covariance approchée au sens de la distance tensorielle. Cette distance est définie, pour deux normales non orientées $\mathtt{n}$ et $\mathtt{m}$, par :

$$d_t(\mathtt{m},\mathtt{n})^2 = \|\mathtt{n} - \mathtt{m}(\mathtt{n}|\mathtt{m})\|^2 = \|\Pi_{\mathtt{m}\perp}(\mathtt{n})\|^2 = (1 - (\mathtt{n}|\mathtt{m})^2) = 1 - \mathtt{n}\overline{\otimes}T(\mathtt{m})\overline{\otimes}\mathtt{n}$$

22. C'est ce que fait Médioni lorsque il parle de linéarité de la fonction de vote [Médioni *et al.*, 2000, page 46] : il exprime le tenseur $\overline{T(n_0)}$ sous sa décomposition bâton-plateau-boule $\overline{T(n_0)} = (\lambda_1 - \lambda_2)S + (\lambda_2 - \lambda_3)P + \lambda_3 B$ et en déduit le tenseur $\overline{T(n)}$ en combinant linéairement les tenseurs de vote bâton-plateau-boule : $\overline{T(n)} = (\lambda_1 - \lambda_2)V_S + (\lambda_2 - \lambda_3)V_P + \lambda_3 V_B$

où $\|\|$ est la norme quadratique usuelle de $\mathbb{R}^{\mathfrak{D}}$ et $(.|.)$ est son produit scalaire usuel, et où $\Pi_{\mathtt{m}\perp}$ est le projection orthogonale le long de $\mathtt{m}$. Elle vérifie aussi :

$$d_t(\mathtt{m},\mathtt{n}) = \sin((\widehat{\mathtt{m},\mathtt{n}})) = d(\mathtt{m},\mathtt{n}) + O(d(\mathtt{m},\mathtt{n})^3)$$

On a donc :

$$\Psi_t(d) = \sin(d) \qquad \psi_t(d) = \frac{\sin(d)}{d} \qquad \psi_t(\overrightarrow{mn}) = \Pi_{\mathtt{m}\perp}(\mathtt{n})$$

7.3.4.1 Approximation tensorielle de l'espérance

En utilisant l'égalité $d_t(\mathtt{m},\mathtt{n})^2 = 1 - \mathtt{n}\overline{\otimes}T(\mathtt{m})\overline{\otimes}\mathtt{n} = 1 - \mathtt{m}\overline{\otimes}T(\mathtt{n})\overline{\otimes}\mathtt{m}$, on peut écrire : $\sigma^2_{t\,\mathbf{n}}(\mathtt{m}) = 1 - E_{\mathbf{n}}\left[\mathtt{m}\overline{\otimes}T(\mathtt{n})\overline{\otimes}\mathtt{m}\right] = 1 - \mathtt{m}\overline{\otimes}E_{\mathbf{n}}\left[T(\mathtt{n})\right]\overline{\otimes}\mathtt{m}$. On retrouve donc le tenseur d'une variable aléatoire tel que défini dans la section 7.3.3, et nous recherchons les normales $\mathtt{m}$ qui maximisent : $\mathtt{m}\overline{\otimes}\overline{T(\mathbf{n})}\overline{\otimes}\mathtt{m}$. Pour les trouver, il faut diagonaliser $\overline{T(\mathbf{n})}$: $\overline{T(\mathbf{n})} = \sum_i(\alpha_i\mathtt{n}_i)\otimes(\alpha_i\mathtt{n}_i)$. On a alors $\mathtt{m}\overline{\otimes}\overline{T(\mathbf{n})}\overline{\otimes}\mathtt{m} = \sum_i \alpha_i(\mathtt{n}_i|\mathtt{m})^2 \leq \max_i(\alpha_i).\sum_i(\mathtt{n}_i|\mathtt{m})^2 = \max_i(\alpha_i)$, avec égalité si et seulement $\mathtt{m}$ appartient au sous-espace propre associé à la plus grande valeur propre. On a donc la proposition suivante :

Proposition 7.5 *Espérance de Fréchet au sens de la distance tensorielle*
L'approximation tensorielle de l'espérance au sens de Fréchet d'une normale non-orientée aléatoire est constituée de l'ensemble des vecteurs unitaires du sous-espace propre de $\overline{T(\mathbf{n})}$ associé à sa plus grande valeur propre.

Elle est de plus capable de traiter correctement le cas des distributions de normales du type plateau et boule (section 7.3.3.5).

Enfin, on doit se rappeler que cette espérance n'est pas orientée. On peut l'orienter en utilisant le même sens que la moyenne vectorielle. Pour cela, il suffit de calculer l'espérance du produit scalaire entre la normale trouvée $\overline{n}$ et la normale aléatoire. Si elle est négative, on inverse la normale $\overline{n}$.

7.3.4.2 Approximation tensorielle de la covariance

La covariance par rapport à une primitive fixe $\mathtt{m}$ peut s'écrire $\Sigma_{t\,\mathbf{nn}}(\mathtt{m}) = E_{\mathbf{n}}\left[\psi_t(\overrightarrow{mn}).\psi_t(\overrightarrow{mn})^t\right] = E_{\mathbf{n}}\left[\Pi_{\mathtt{m}\perp}(\mathtt{n}).\Pi_{\mathtt{m}\perp}(\mathtt{n})^t\right]$, soit, en utilisant la matrice de la projection

$$\Sigma_{t\,\mathbf{nn}}(\mathtt{m}) = \Pi_{\mathtt{m}\perp}.E_{\mathbf{n}}\left[\mathtt{n}.\mathtt{n}^t\right].\Pi^t_{\mathtt{m}\perp} = \Pi_{\mathtt{m}\perp}.\overline{T(\mathbf{n})}.\Pi^t_{\mathtt{m}\perp}$$

Le tenseur d'une normale aléatoire contient donc aussi une approximation de la covariance par rapport à n'importe quelle primitive fixe $\mathtt{m}$.

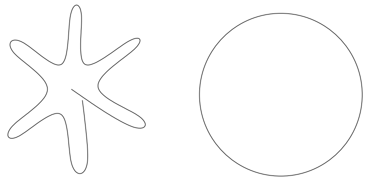
Proposition 7.6 *Covariance au sens de la distance tensorielle*

L'approximation tensorielle de la covariance d'une normale non-orientée aléatoire est obtenue en restreignant $\overline{T(\mathbf{n})}$ *à l'hyperplan orthogonal à la moyenne.*

7.3.4.3 Synthèse

Le tenseur utilisé par Lee et Médioni est permet donc une approximation extrêmement efficace de l'espérance et de la covariance d'une primitive aléatoire. Il suffit en effet de les diagonaliser, et la plus grande valeur propre et ses vecteurs propres unitaires associé indique l'espérance, et les autres valeurs propres et vecteurs propres indiquent la covariance.

Elle trouve cependant ses limites dans certains cas extrêmes de lois multimodales, comme dans l'exemple suivant :


Loi multimodale   Espérance approchée

7.3.4.4 Approximation tensorielle de la distance de Mahalanobis

Le tenseur d'une normale aléatoire contient une approximation de la covariance qui permet de calculer la distance de Mahalanobis. Nous allons donc ici voir comment exprimer directement et simplement une approximation de la distance de Mahalanobis en fonction du tenseur.

La vraie distance de Mahalanobis entre une primitive n et une primitive aléatoire centrée en $\mathbf{o}$ et de covariance $\Sigma_{\mathbf{oo}}$ est donnée par :

$$\mu^2(\mathbf{o},n) = \overrightarrow{on}\,\overline{\otimes}\,\Sigma_{\mathbf{oo}}^{-1}\,\overline{\otimes}\,\overrightarrow{on}$$

où $\overrightarrow{on}$ et $\Sigma_{\mathbf{oo}}$ sont calculés dans la carte exponentielle. Il semble alors naturel de remplacer ces deux éléments par leur approximation tensorielle. On a alors :

$$\mu_t^2(\mathbf{o},n) = \Psi_t(\overrightarrow{on})\,\overline{\otimes}\,\Sigma_{t\,\mathbf{oo}}^{-1}\,\overline{\otimes}\,\Psi_t(\overrightarrow{on})$$

Or $\Psi_t(\overrightarrow{on})$ et $\Sigma_{t\,\mathbf{oo}}^{-1}$ sont respectivement la projection de n et de $\overline{T(\mathbf{o})}^{-1}$ dans le plan perpendiculaire à $\mathbf{o}$. On peut alors se contenter de ne projeter qu'un seul de ces deux éléments pour obtenir le même résultat :

$$\mu_t^2(\mathbf{o},n) = n\,\overline{\otimes}\,\Sigma_{t\,\mathbf{oo}}^{-1}\,\overline{\otimes}\,n = \Psi_t(\overrightarrow{on})\,\overline{\otimes}\,\overline{T(\mathbf{o})}^{-1}\,\overline{\otimes}\,\Psi_t(\overrightarrow{on})$$

On utilisera en général la première formule, car on doit de toute façon diagonaliser le tenseur pour connaître la normale moyenne et savoir ainsi dans quelle direction effectuer la projection. Il est alors aisé de calculer la covariance en annulant la première valeur propre.

Le principal intérêt de cette approximation est qu'elle conserve sa principale propriété : calculons l'espérance de la distance de Mahalanobis approchée entre une primitive aléatoire et son espérance (lorsque celle ci est unique) :

$$\begin{aligned}
E\left[\mu_t^2(\mathbf{o},n)\right] &= \int_{\mathcal{M}} \Psi_t(\overrightarrow{on})^t.\Sigma_{t\,\mathbf{oo}}^{-1}.\Psi_t(\overrightarrow{on}).p_{\mathbf{o}}(n).d\mathcal{M}(n) \\
&= \int_{\mathcal{M}} \mathrm{Tr}(\Sigma_{t\,\mathbf{oo}}^{-1}.\Psi_t(\overrightarrow{on}).\Psi_t(\overrightarrow{on})^t).p_{\mathbf{o}}(n).d\mathcal{M}(n) \\
&= \mathrm{Tr}\left(\Sigma_{t\,\mathbf{oo}}^{-1}.\int_{\mathcal{M}}^{-1} \Psi_t(\overrightarrow{on}).\Psi_t(\overrightarrow{on})^t.p_{\mathbf{o}}(n).d\mathcal{M}(n)\right) \\
&= \mathrm{Tr}(\Sigma_{t\,\mathbf{oo}}^{-1}.\Sigma_{t\,\mathbf{oo}}) = \mathrm{Tr}(Id) \\
&= \mathcal{D}
\end{aligned}$$

l'espérance de la distance de Mahalanobis approchée entre une primitive aléatoire et son espérance est donc toujours égale à la dimension de l'espace.

Enfin, on peut utiliser directement cette distance de Mahalanobis pour calculer la densité de probabilité approchée d'une loi gaussienne :

$$p_{t\,\mathbf{o}}(n) = \exp\left(\frac{\mu_t^2(\mathbf{o},n)}{2}\right)/k$$

7.3.5 Comparaison des statistiques exactes et approchées

Nous allons ici comparer les précisions des différentes méthodes d'estimation de la moyenne d'un ensemble de normales. Nous travaillerons avec les normales non-orientées.

Nous allons effectuer une série d'expériences sur le modèle suivant : on tire une première normale aléatoire uniforme, puis n normales suivant une loi gaussienne centrée autour de la première et d'écart-type σ. On estime l'espérance suivant les trois méthodes (l'espérance rigoureuse : min de la distance angulaire par descente de gradient, les deux espérances approchées : min de la distance vectorielle par moyenne classique+normalisation, min de la distance tensorielle par calcul du tenseur+extraction du premier vecteur propre). On examine alors l'écart entre le véritable centre de la distribution et les 3 espérances ainsi calculés. On mesure aussi l'écart entre les deux espérances approchées et l'espérance rigoureuse.

Pour chaque paramètres, on réalise ainsi plusieurs expériences (10000), et on calcule la RMS (figure 7.9) et le maximum de chacun des écarts (figure 7.10).

On constate alors les faits suivants :

- Jusqu'à 20 degrés, les 3 méthodes d'estimation ont des précisions comparables : les écarts absolus (entre l'espérance estimée et la vraie valeur centrale de notre loi)

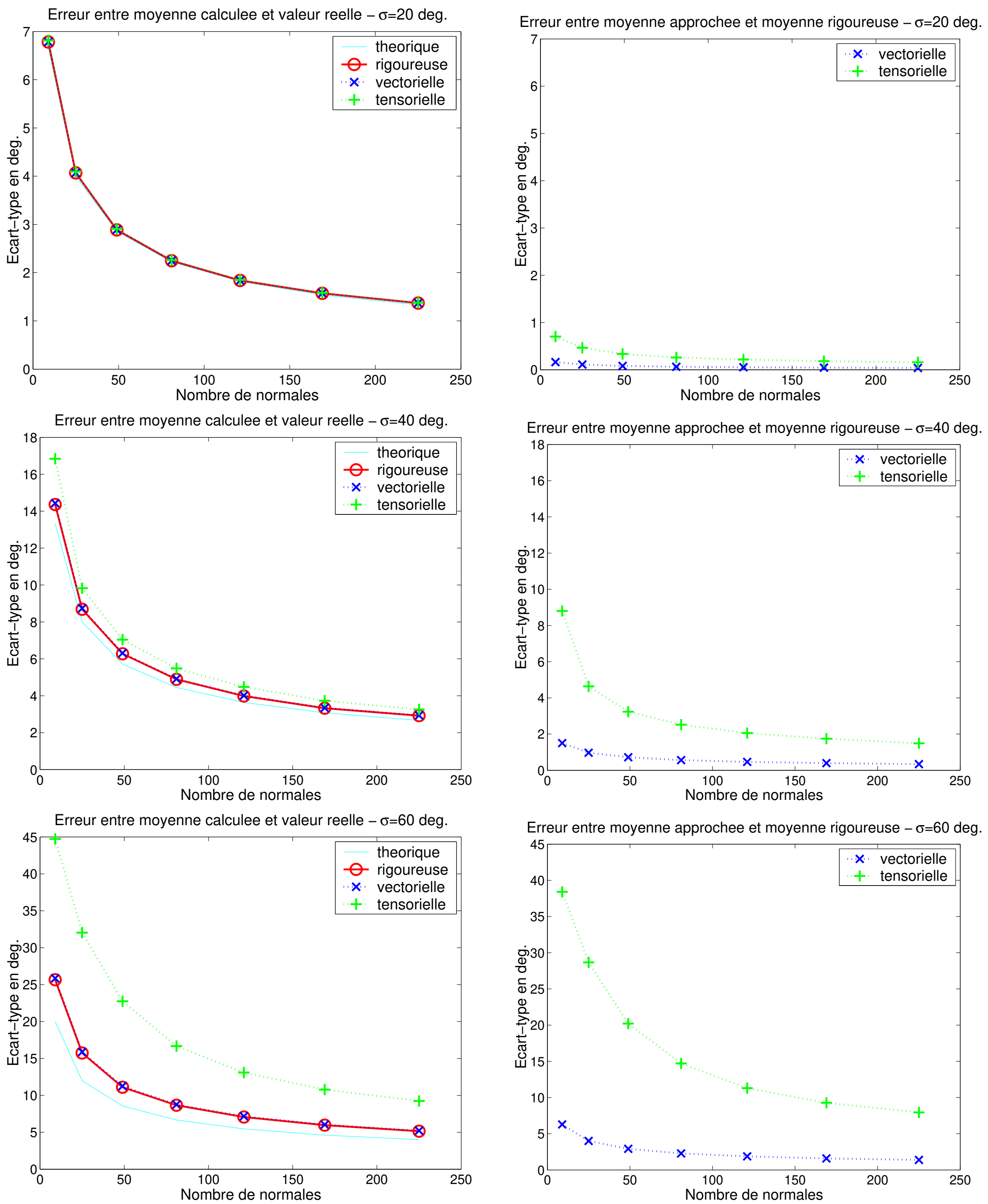


FIG. 7.9 – *Écart-type de l'erreur sur l'estimation de la moyenne d'une normale aléatoire 3D par la méthode rigoureuse et les méthodes approchées. A gauche : erreur des méthodes par rapport à la véritable valeur moyenne de la normale aléatoire. A droite : erreur des méthodes approchées par rapport au résultat de la méthode rigoureuse.*

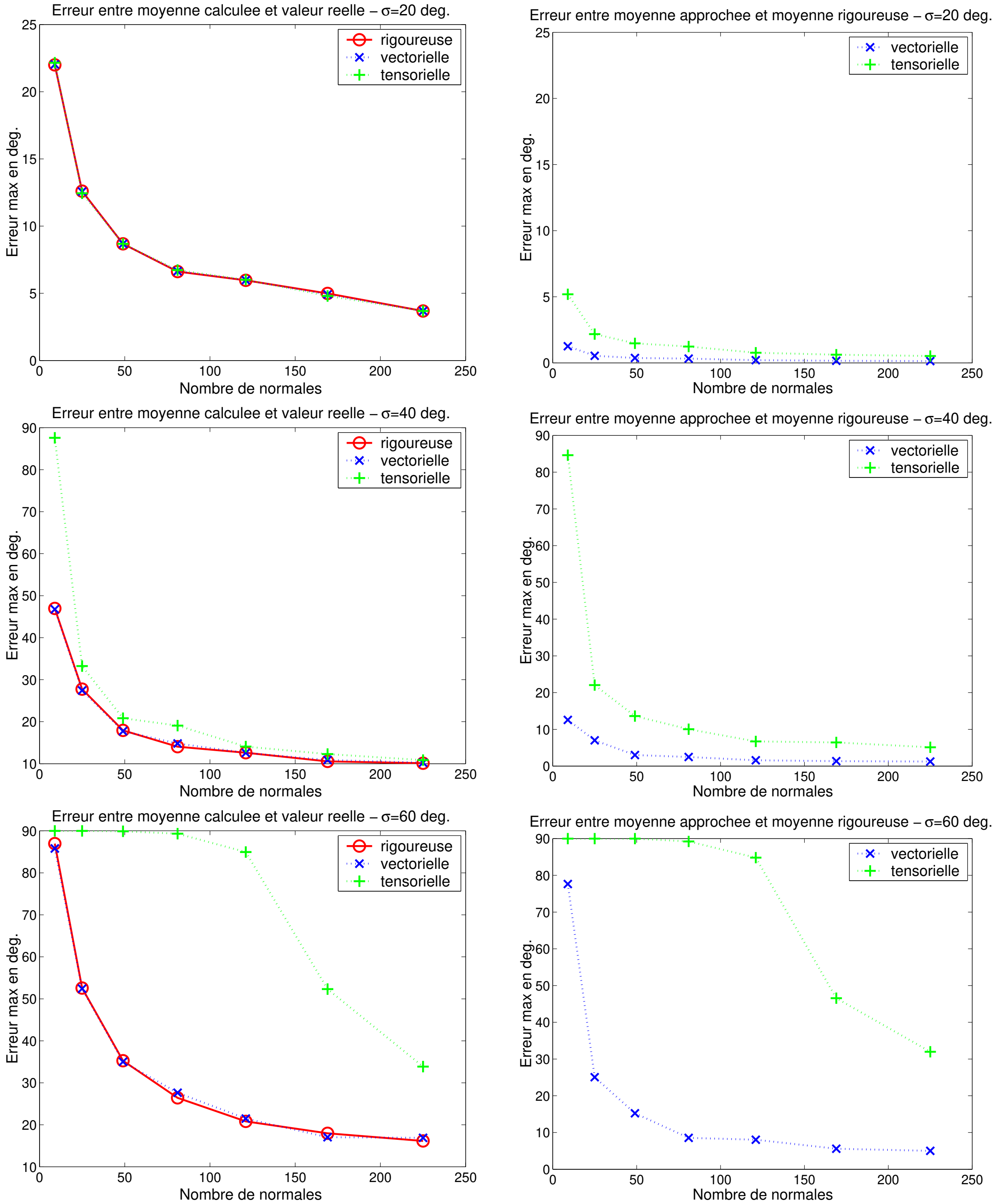


FIG. 7.10 – *Erreur maximale sur l'estimation de la moyenne d'une normale 3D aléatoire par la méthode rigoureuse et les méthodes approchées.*

sont analogues. Ceci explique par le fait que les écarts relatifs (entre les différentes méthodes d'estimations) restent négligeables par rapport aux écarts absolus.

- Au delà, l'espérance vectorielle garde un écart absolu comparable à celui de l'espérance rigoureuse, même si l'écart relatif entre les deux estimés augmente. On constate même dans certains cas avec forte variance que l'espérance vectorielle est légèrement plus précise. Cela provient probablement du fait que la distance vectorielle est une M-Estimée de la distance angulaire, et est par conséquent plus robuste [Rousseeuw and Leroy, 1987].
- En revanche, l'espérance tensorielle devient inutilisable. A 40 degrés, et pour un faible nombre de normales, les écarts maximaux augmentent, même si les variances des écarts restent comparables. La méthode est donc encore bonne en moyenne, mais peu robuste. Au delà de 60 degré, les écarts deviennent trop importants.

On ne pourra donc utiliser les tenseurs que pour des lois assez bien définies (écart-type inférieur à 40 degrés). Au delà, il faudra revenir à des techniques plus précises : l'espérance vectorielle, et même si besoin les techniques rigoureuses.

7.3.6 Statistiques approchées sur les rotations aléatoires

Des méthodes de statistiques approchées sur les rotations ont déjà été développées et largement étudiées. L'objet de cette section n'est donc pas de les représenter en détail, mais simplement de les replacer dans notre cadre mathématique et de faire le parallèle avec les méthodes utilisées pour les normales.

On distingue 3 méthodes couramment utilisées pour calculer une rotation moyenne :

- Le calcul de la vraie moyenne, au sens de la distance intrinsèque, introduite par [Pennec, 1996]. Cette méthode est itérative comme dans le cas des normales (voir section 7.2.7).
- Le calcul direct par la méthode de la SVD [Arun *et al.*, 1987; Umeyama, 1991].
- Le calcul direct par la méthode des quaternions unitaires [Horn, 1987].

Ces deux dernières méthodes ont été d'abord introduite dans le cadre du recalage rigide puis adaptées pour le calcul de la rotation moyenne. Les trois méthodes ont été comparées dans le cadre du recalage [Eggert *et al.*, 1997] et dans le cadre du calcul de la rotation moyenne [Gramkow, 2001]. Cette dernière comparaison permet de comprendre que les deux dernières méthodes suivent exactement la même démarche que la méthode vectorielle pour les normales (voir 7.3.2) :

- La méthode des quaternions unitaires calcule dans un premier temps la moyenne vectorielle des quaternions unitaires qui représentent les rotations. On obtient alors un quaternion $\overline{q}$ qui n'est plus forcément unitaire. On retrouve alors la rotation moyenne en prenant le quaternion unitaire $\overline{Q}$ le plus proche de $\overline{q}$ au sens de la distance euclidienne entre quaternions (il suffit pour cela de normaliser $\overline{q}$).

- La méthode de la SVD calcule dans un premier temps la moyenne des matrices qui représente les rotations. On obtient alors une matrice $\overline{M}$ qui n'est plus forcément orthogonale. On retrouve alors la rotation moyenne en prenant la matrice orthogonale $\overline{R}$ la plus proche de $\overline{M}$ au sens de la distance euclidienne entre matrices.

L'article montre aussi que les deux méthodes directes minimisent la variance totale au sens des distances suivantes :

$$d_{quat}^2 = 8(1 - \cos(\theta/2)) = (4\sin(\theta/4))^2$$

$$d_{SVD}^2 = 2(1 - \cos(\theta)) = (2\sin(\theta/2))^2$$

où θ est l'angle entre les deux rotations, qui est la distance intrinsèque.

Finalement, ces deux méthodes sont donc des méthodes de statistiques approchées pour le calcul des rotations, avec :

$$d_{quat} = \Psi_{quat}(d) = 4\sin(d/4)$$

$$d_{SVD} = \Psi_{SVD}(d) = 2\sin(d/2)$$

ces fonctions sont sensiblement les mêmes que celles que nous avons utilisées pour le calcul approché des normales.

Pour finir notre parallèle, on peut comparer les résultats expérimentaux des méthodes sur les normales et des méthodes sur les rotations. L'article indique que les résultats commencent à significativement se détériorer pour un écart-type de 40 degrés, et ne va pas au delà. C'est aussi le cas pour nos méthodes.

7.4 Calcul des normales pour un nuage de points non structuré

Nous allons examiner une première application qui nous permettra de nous familiariser avec le vote de tenseur que nous étudierons dans le chapitre suivant. Il s'agit d'estimer les normales lorsqu'on dispose d'un nuage de points non structurés et non orientés, mais qui provient d'une surface en 3D (ou d'un contour en 2D ou d'une hypersurface en nD).

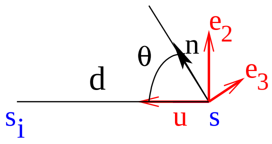
Pour cela, on utilise la remarque suivante : si deux points sont sensés appartenir à la même surface, la normale à cette surface a plus de chances d'être perpendiculaire au segment qui relie ces deux points. Aussi, pour chaque point de notre nuage, il suffit de synthétiser l'information donnée par les autres points du nuage (les votes) pour en déduire la normale la plus probable. Cette synthèse des votes est effectuée en utilisant les tenseurs précédemment définis.

Les calculs seront indépendants pour chaque point du nuage. Nous allons donc détailler les différentes étapes pour un point de la scène donné, les interpréter, et évaluer la qualité du résultat.

7.4.1 Vote d'un point en un autre point

Fixons nous un point du nuage pour lequel nous allons rechercher la normale la plus probable. Nous le noterons s. Prenons maintenant un des autres points du nuage, noté s_i.

Nous allons essayer de modéliser la loi de probabilité de la normale n_s au point s, sachant que le point s_i appartient aussi à la surface. Cette loi est donc notée $p(n_s/s,s_i)$. Nous allons commencer par utiliser les symétries du problème : il est invariant par rotation (d'angle quelconque) autour du segment ss_i. Notre loi de probabilité ne dépend donc que de la distance d entre les deux points et de l'angle θ entre la normale n_s et le segment ss_i - elle est de plus symétrique par rapport à θ : $p(n_s/s,s_i) = p(|\theta|,d)$.



On peux alors synthétiser cette loi dans un tenseur (rappelons que le point, et donc d est fixe) : on intègre sur l'ensemble des normales (i.e. sur θ). On travaille pour cela dans le repère orthogonal formé par le vecteur ss_i normalisé noté u et une base de l'espace perpendiculaire notée $(e_2,e_3,...,e_{\mathfrak{D}})$. On obtient alors, par symétrie (les directions e_2, ..., $e_{\mathfrak{D}}$ sont équivalentes), un tenseur nécessairement de la forme suivante :

$$\overline{T(\mathbf{n}_s/s,s_i)} = \alpha^2(u \otimes u) + \sum_j \beta^2(e_j \otimes e_j) = (\alpha^2 - \beta^2)(u \otimes u) + \beta^2(u \otimes u + \sum_j e_j \otimes e_j)$$

avec

$$\alpha^2 = E\left[\cos^2\theta\right] \qquad \beta^2 = E\left[\sin^2\theta\right]/(\mathfrak{D}-1)$$

On a donc une combinaison d'un tenseur boule et d'un tenseur bâton. Reste à déterminer α et β en fonction de d, ou plus simplement $(\alpha^2 - \beta^2)$ dans la mesure où le tenseur boule n'apporte aucune information sur l'orientation.

Nous le ferons pour l'instant de façon assez heuristique, les modèles de vote étant détaillés dans le chapitre suivant (où on dérive en particulier le modèle boule, section 8.6.1.3, que l'on utilise justement dans le cas où l'on ne connaît pas les normales aux points). Intuitivement, on voudrait favoriser les surfaces de forme plus lisses, donc les normales perpendiculaires au segment ss_i. On doit donc avoir une probabilité croissante et maximale pour $\theta = \pi/2$, et symétrique autour de ce mode, qui est donc aussi sa moyenne. En conséquence, $\alpha^2 < \beta^2$.

Ensuite, on peut penser que l'incertitude (i.e. l'écart-type) sur cette moyenne dépend de la distance entre les deux points : s'ils sont proches, la surface n'a pas le temps de se tordre ou se courber, l'écart-type est donc faible ($\alpha^2 \to 0$ et $\beta^2 \to 1/(\mathfrak{D}-1)$). S'ils sont éloignés, la surface a le temps de se courber, et l'écart-type sur l'orientation de la normale est plus élevé ($\alpha^2 \to 1/\mathfrak{D}$ et $\beta^2 \to 1/\mathfrak{D}$).

Modélisation gaussienne Nous allons utiliser une modélisation gaussienne sans la justifier : disons qu'on a une loi gaussienne sur la normale n_s, centrée autour de la perpendiculaire au segment, et qui possède une dispersion proportionnelle à la distance entre les points[23] (a^2 représente la dispersion angulaire sur la normale pour une distance d'un mm) :

$$p(\theta,d) = \exp(-\frac{(\pi/2-|\theta|)^2}{2da^2})/k$$

En 2D, on trouve alors (I représente le nombre imaginaire : $\sqrt{-1}$) :

$$\begin{bmatrix} \alpha^2 \\ \beta^2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} - \\ + \end{bmatrix}\frac{1}{2}\exp(-2da^2)\frac{erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi+4Ida^2))+erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi-4Ida^2))}{2.erf(\frac{\sqrt{2}}{4a\sqrt{d}}\pi)}$$

En 3D, on trouve alors :

$$\begin{bmatrix} \alpha^2 \\ \beta^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4}-\frac{1}{4}* \\ \frac{3}{8}+\frac{1}{8}* \end{bmatrix}\exp(-4da^2)\frac{erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi+6Ida^2))+erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi-6Ida^2))}{erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi+2Ida^2))+erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi-2Ida^2))}$$

7.4.2 Intégration des votes en un point

On veut maintenant intégrer les votes des différents points de la scène. On va leur donner une probabilité à priori $p(s_i) = \pi_i$, qui sera en général uniforme, mais pourra aussi permettre de prendre en compte une confiance a-priori, si bien que l'on peut écrire la probabilité empirique :

$$p(n_s/S) = \sum_i p(n_s/s,s_i)p(s_i) = \sum_i \pi_i p(n_s/s,s_i)$$

Le passage au tenseur étant linéaire, on trouve alors :

$$\overline{T(\mathbf{n}_s/S)} = \sum_i \pi_i \overline{T(\mathbf{n}_s/s,s_i)} = \sum_i \pi_i(\alpha_i^2-\beta_i^2)(u_i \otimes u_i) + Cte * Id$$

23. Ceci correspond à un modèle stochastique, que nous présenterons dans le chapitre suivant, section 8.5.2 : on suppose que l'on peut suivre, sur la surface, des courbes à vitesse constante, et que la direction sur ces courbes suit un mouvement brownien. La dispersion sur la direction de la tangente est alors proportionnelle à la distance au premier point de la courbe, ici s.

7.4.3 Extraction de la normale

Le tenseur que l'on a calculé ci-dessus représente véritablement la loi de probabilité sur la normale en s. On extrait donc la normale la plus probable en calculant le vecteur propre associé à la plus grande valeur propre de ce tenseur. On comprend pourquoi on peut se contenter de calculer le tenseur à une composante boule près (représentée ici par $Cte * Id$), et une constante multiplicative positive près : en effet, multiplier le tenseur par un scalaire ou lui ajouter un tenseur boule ne fait que modifier les valeurs propres, sans modifier ni leur ordre ni leurs vecteurs propres associés.

7.4.4 Pertinence de la normale

Lorsqu'on le souhaite, on peut aussi obtenir une information sur la pertinence de la normale ainsi extraite. En effet le tenseur, lorsqu'il est entièrement calculé (i.e. lorsqu'on n'élimine pas la partie boule et lorsqu'on veille à bien le normaliser) puis diagonalisé (on connaît alors ses valeurs propres dans l'ordre décroissant : $\lambda_1 > ... > \lambda_{\mathfrak{D}}$), contient une approximation de la covariance sur la normale, donnée par les valeurs propres secondaires $(\lambda_2,...,\lambda_{\mathfrak{D}})$. On peut alors :

- étudier directement cette covariance, ou simplement la variance totale σ^2 (la trace de la covariance ou plus simplement la somme des valeurs propres secondaires du tenseur : $\lambda_2 + ... + \lambda_{\mathfrak{D}}$).
- étudier le complémentaire de la variance totale $1 - \sigma^2$, qui sera d'autant plus élevé que la variance et donc le doute sur la direction de la normale sera faible. Cet indice est en fait très simple à calculer : $1 - \sigma^2 = 1 - \lambda_2 - ... - \lambda_{\mathfrak{D}} = \lambda_1$. Il s'agit donc de la plus grande valeur propre, qui indique la différence de pertinence entre la normale la plus probable et les autres normales possibles.
- étudier une variante de ce complémentaire : on peut par exemple penser à la différence entre les deux premières valeurs propres $\lambda_1 - \lambda_2$, qui correspond à l'amplitude de la composante bâton du tenseur et indique la différence de pertinence entre la normale la plus probable et la "deuxième" plus probable.
- étudier ce que Médioni appelle la saillance qui est l'indice précédent avant normalisation du tenseur. Cela n'a pas vraiment de sens mathématique, mais permet de prendre en compte le nombre de votes reçus en un point, et de pouvoir accentuer par la suite les zones où beaucoup d'informations étaient disponible.

7.4.5 Implémentation pratique

Pour l'implémentation pratique, nous utiliserons, comme fonction de vote, une version simplifiée de la modélisation gaussienne.

Vote simplifié en 2D L'équation ci-dessus nous donne en 2D :

$$\alpha^2 - \beta^2 = -\exp(-2da^2)\frac{erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi + 4Ida^2)) + erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi - 4Ida^2))}{2.erf(\frac{\sqrt{2}}{4a\sqrt{d}}\pi)}$$

Nous obtenons, à un terme correctif près, une belle gaussienne. Or ce terme correctif est proche de 1 pour de faibles valeurs de da^2, ce qui nous intéresse.

Nous utiliserons donc l'approximation suivante :

$$\alpha^2 - \beta^2 = -\exp(-2da^2)$$

Vote simplifié en 3D L'équation ci-dessus nous donne, en 3D :

$$\alpha^2 - \beta^2 = -(\frac{1}{8} + \frac{3}{8}\exp(-4da^2)\frac{erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi + 6Ida^2)) + erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi - 6Ida^2))}{erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi + 2Ida^2)) + erf(\frac{\sqrt{2}}{4a\sqrt{d}}(\pi - 2Ida^2))})$$

Comme dans le cas 2D, le terme correctif tend vers 1 en 0. Une approximation grossière est donc :

$$\alpha^2 - \beta^2 = -\frac{1}{2}\exp(-4da^2)$$

Algorithme & variante Pour des raisons d'efficacité algorithmique, nous n'utiliserons pas, pour un point de la scène donnée, tous les autres points, mais uniquement les points suffisamment proche, que nous rechercherons avec un kD-Tree, comme dans le recalage (voir 3.2.3). Les points plus éloignés auront de toute façon des contributions très faibles, que l'on pourra négliger.

L'algorithme est donc basé sur deux paramètres, la dispersion de l'angle par unité de distance a^2 et la distance limite de recherche d_{max}. Son pseudo code est le suivant :

Boucle : pour chaque point s_i :

- Recherche des points proches à l'aide d'un kD-Tree : recherche des s_j tel que $j \neq i$ et $|s_i s_j| < d_{max}$.
- Calcul du tenseur : $T = \sum_j -\exp(-4a^2\,|s_i s_j|).\frac{s_i s_j}{|s_i s_j|} \otimes \frac{s_i s_j}{|s_i s_j|}$.
- Recherche des valeurs et vecteurs propres de T
- Détermination de la normale : vecteur propre associé à la plus grande valeur propre

Notons que l'algorithme ici présenté concerne le calcul des normales aux hypersurfaces (courbes en 2D, surface en 3D). Il peut être aisément adapté pour le calcul des tangentes aux courbes (en dimension quelconque), en prenant le vecteur associé à la plus faible valeur propre.

7.4.6 Comparaison avec la technique Médioni

Lorsqu'il dispose de points non-orientés en entrée, Médioni utilise une première phase de vote pour estimer leurs orientations. Pour cela, il calcule pour chaque point de son nuage un tenseur défini comme la somme des votes des autres points du nuage. L'intégration des votes est donc identique à la nôtre.

Il ne garde ensuite que la composante bâton du tenseur obtenu, et s'en sert comme entrée dans les phases suivantes de son moteur de reconstruction. Cette composante bâton correspond au plus grand vecteur propre, si bien qu'elle représente implicitement la normale que nous avons extrait explicitement. Elle contient cependant une information supplémentaire, sa norme, qui correspond à la différence entre la première et la deuxième valeur propre. Nous avons déjà présenté cette valeur (section 7.4.4), appelée saillance, et expliqué quel pouvait être son intérêt pour des traitements ultérieurs.

La seule différence majeure entre notre technique et celle de Médioni est la fonction de vote élémentaire utilisée. Chez Médioni, celle-ci n'est pas donnée explicitement, mais obtenue par intégration numérique de la fonction de vote élémentaire d'un tenseur bâton. Nous l'avons essayé dans nos expériences, et avons obtenu des résultats légèrement moins bon qu'avec notre fonction de vote. Ceci semble confirmer l'on peut faire varier la fonction de vote sans différence notables dans les résultats [Lee, 2002].

7.4.7 Comparaison avec d'autres techniques

Il existe de nombreuses autres méthodes permettant d'interpoler les normales pour un nuage de points non-structurés. Mais qu'elles fassent appel aux voisins naturels [Amenta and Bern, 1999] ou à des approximations locales par des modèles de surface, ces méthodes sont d'essence géométrique et non probabiliste, et se révèlent très sensible au bruit, en particulier à la présence de points aberrants. On peut cependant faire un parallèle intéressant entre nos travaux et ceux de [Cohen-Steiner and Morvan, 2002] qui a dérivé, dans le cadre de la géométrie algorithmique, une technique permettant de calculer les informations d'ordre 1 et 2 (normale, mais aussi courbures) sur une surface triangulée. Cette technique, qui fait pourtant appel à des concepts complètement différents, s'implémente elle-aussi avec un vote de tenseur.

7.4.8 Résultats

Nous présentons ici les résultats de l'algorithme sur des données synthétiques et sur des surfaces segmentées dans les scanners.

Dans toutes ces expériences, nous avons fait varier une distance caractéristique $CDist$. Le paramètre a^2 a été fixé à une dispersion de $1rad^2$ par distance caractéristique : $a^2 = 1/CDist$ et nous avons déterminé la distance de recherche maximale à partir d'une distance de Mahalanobis maximale μ^2_{max} :

$$4a^2d < 2.\mu^2_{max} \Longrightarrow d_{max} = \mu^2_{max}/2a^2 = \frac{\mu^2_{max}}{2}CDist$$

Nous avons alors étudié la distance angulaire non-orientée entre les normales trouvées par notre méthode et les vraies normales.

7.4.8.1 Données synthétiques

Nous avons ici utilisé des nuages de points échantillonnées sur une forme géométrique simple et connaissions donc parfaitement les normales.

La forme utilisée est un ellipsoïde dont les demi-axes font 100, 50 et 30 mm. L'échantillonnage est composé de 1000 points répartis au hasard (fig. 7.11 gauche), que nous avons détérioré en le bruitant de deux façons :

- En rajoutant un bruit gaussien isotrope sur la position des points (fig. 7.11 centre).
- En rajoutant des points aberrants, répartis uniformément dans la boîte englobante de l'ellipsoïde (fig. 7.11 droite).

FIG. 7.11 – *Échantillonnage irrégulier d'une ellipse de demi-axes 100,50 et 30 mm. Gauche : avec les normales de référence. Centre : avec un bruit de position (écart-type 1 mm). Droite : avec des points aberrants (1 point aberrant pour 1 point du nuage original)*

Nous avons alors tracé la RMS de la distance entre les normales calculées et les normales de référence en fonction de la distance caractéristique (fig. 7.12). On voit sur les résultats que la principale force de la méthode est sa résistance aux points aberrants (les résultats sont encore significativement bons avec 8 fois plus de points aberrants que de points de l'ellipse) et au bruit de position (jusqu'à 2mm, mais un écart-type de 8mm sur une ellipse de 100mm, c'est énorme).

On peut maintenant se poser la question de savoir comment régler le paramètre de l'algorithme (a^2 ou $CDist$ suivant l'approche). Nous le ferons pour l'instant de manière purement expérimentale, en prenant le paramètre qui minimise l'erreur commise (25 dans les expériences précédentes). Nous reviendrons sur ce point dans le chapitre suivant (section 8.7).

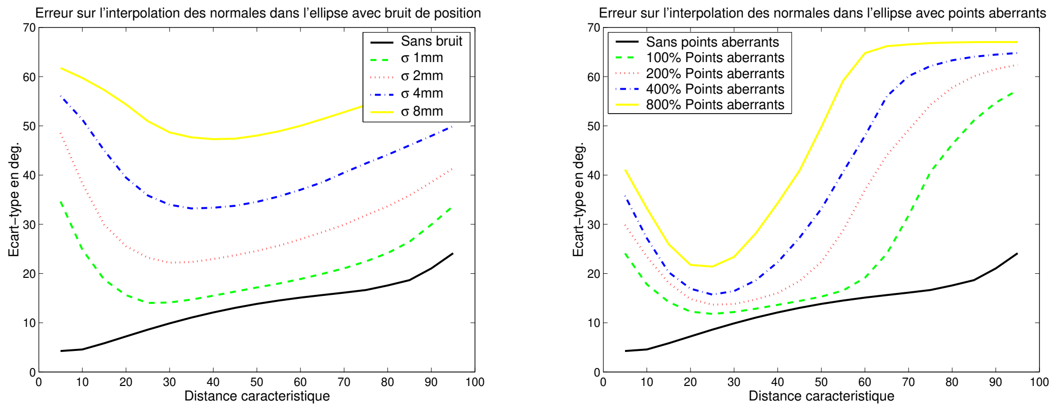


FIG. 7.12 – *RMS de l'erreur commise sur l'estimation des normales pour l'ellipse, en fonction de la distance caractéristique. Gauche : avec un bruit de position (de 1 à 8 mm). Droite : avec des points aberrants (de 100% à 800%).*

7.4.8.2 Données réelles

Nous avons ici testé le calcul des normales sur les surfaces des dents extraites à partir des scanners. On dispose en-effet dans ces surfaces des normales calculées à l'aide du gradient de l'intensité dans l'image scanner. Elles ne sont pas exactes, mais on a pu estimer leur précision en étudiant la RMS de l'angle (section 6.4.2) : l'écart-type est de l'ordre de 10 degrés.

Les résultats sont présentés sur deux cas figure 7.13. Ils indiquent que les résultats sont assez précis.

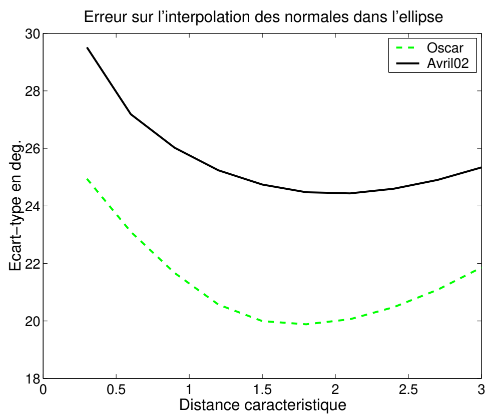


FIG. 7.13 – *RMS de l'erreur commise sur l'estimation des normales pour la surface des dents extraites d'un scanner.*

7.5 Conclusion

Nous avons ici montré commencé par résumer la théorie rigoureuse de [Pennec, 1996] sur les primitives aléatoires et présenté en particulier différents algorithmes pour les normales aléatoires.

Nous avons ensuite vu comment définir, à partir de distances approchées, des méthodes approchées d'estimation de l'espérance et la covariance d'une primitive aléatoire. Nous avons présenté plusieurs exemples de ces méthodes pour les normales et les rotations aléatoires et avons montré leurs qualités (simplicité de calcul, robustesse et précision) et leurs limites (mauvais comportement pour de grandes covariances ou des lois multimodales, incapacité à gérer des lois à espérance multiple).

Nous nous sommes ensuite focalisé sur un cas particulier de ces techniques : le tenseur d'une normale aléatoire, et avons montré comment il permet de synthétiser de façon linéaire l'information sur l'espérance et la covariance d'une normale aléatoire non-orientée.

Ce tenseur a permis, moyennant quelques modifications mineures, de donner une explication théorique cohérente à la technique de vote de tenseurs introduite de façon heuristique par Médioni et al. pour interpoler les tangentes ou les normales dans un nuage de points non-orientés issu d'une courbe ou d'une surface.

Il reste maintenant à expliquer la suite des techniques de vote de tenseurs développées par Médioni pour l'estimation des courbures des surfaces et la reconstruction de surfaces. Une telle explication pourrait en-effet permettre de corriger ses algorithmes (en indiquant par exemple quand et comment utiliser des statistiques rigoureuses lorsque le tenseur s'avère inapproprié), ouvrir la voie à une théorie des surfaces probabilistes et permettre le développement de nouvelles applications basées sur le vote de tenseur (pour la fusion et le recalage de surfaces notamment).

Chapitre 8

Surfaces échantillonnées aléatoires

Sommaire

8.1 Introduction

8.1.1 Problématique

Nous venons de nous intéresser aux normales aléatoires, et avons en particulier justifié l'utilisation des tenseurs pour calculer leurs moyennes et leurs covariances. Nous pouvons maintenant nous intéresser aux points orientés aléatoires, et plus particulièrement à ceux qui proviennent d'une courbe ou d'une surface échantillonnée. En effet, qu'ils s'agisse d'un point muni de sa tangente pour une courbe ou d'un point muni de sa normale pour une surface, il donne une information particulière que nous n'avons pas encore vraiment prise en compte : il indique l'espace tangent de la surface. Et ceci peut aider à prévoir la position d'un autre point échantillonné sur la même surface.

Nous allons donc dans ce chapitre chercher des *modèles statistiques* qui, au travers d'une loi de probabilité $p(s_i|m_j)$, indiquent où peut potentiellement se trouver un point s_i lorsqu'on connaît un point m_j de la surface, en tenant compte des trois phénomènes aléatoires suivants : la *forme de la surface*, la façon dont est réalisé *l'échantillonnage*, le *bruit de mesure*. Nous voudrons par contre nous abstraire de la position de la surface et chercherons donc une approche *invariante par transformation rigide* (i.e. qui donne les mêmes résultats lorsqu'on change de repère).

Les modèles ainsi dérivés pourront par exemple être utilisés pour adapter le terme $p(s_i|m_j,T) = p(T \star s_i|m_j)$ sur lequel repose les algorithmes ICP et ICP/EM (voir par exemple l'eq. 4.1). Mais ils dépassent de loin le simple cadre du recalage : leur mise au point fait appel à des notions provenant de domaines très différents, et ils sont susceptibles d'être utilisés dans de nombreuses autres applications sur les surfaces échantillonnées. Ils permettraient en-effet, par le biais d'algorithmes bayésiens, de prendre intrinsèquement le bruit en compte, et donc d'obtenir des résultats plus robustes que les approches déterministes lorsque les données sont bruitées.

8.1.2 État de l'art

D'un point de vue théorique, il n'existe malheureusement pas à l'heure actuelle de modélisation probabiliste complète pour les courbes ou les surfaces. Les travaux qui s'en rapprochent le plus sont les travaux sur les fonctions aléatoires de la forme $y = f(x)$ [Rice and Silverman, 1991]. Mais ils font jouer un rôle très différent au paramètre x et à la valeur y et ne sont donc pas invariants par rotation.

Il existe cependant, d'un point de vue plus applicatif, les travaux basés sur la notion de saillance (que l'on peut présenter comme la “vraisemblance” d'un élément de courbe ou de surface), introduite par [Sha'ashua and Ullman, 1988] et qui a connu de nombreuses variantes [Williams and Thornber, 2000]. Ces méthodes présentent de nombreuses qualités :

- Elles concernent des applications très différentes : reconstruction de surface [Médioni *et al.*, 2000], complétion de courbes [Williams and Jacobs, 1997], lissage [August, 2001] et segmentation [Williams and Thornber, 2000] d'images 2D.
- Elles sont invariantes par transformation rigide.
- Elles présentent souvent de bonnes propriétés lorsqu'on change l'échelle.
- Elles donnent des résultats pertinents en présence de bruits, mêmes importants.

La diversité des méthodes et la qualité des résultats obtenus en présence de bruit nous ont amené à nous intéresser à cette notion de saillance. Nous nous sommes en particulier focalisés sur deux sous-ensembles de ces méthodes.

Les méthodes de l'équipe de Médioni [Médioni *et al.*, 2000] reposant sur le vote de tenseurs en 3D : il s'agit des seuls travaux en 3D, axés principalement sur la reconstruction de courbes et surfaces. Ces techniques ont donné de remarquables résultats en présence d'un nombre élevé de points aberrants. C'est en cherchant à comprendre quel modèle théorique pouvait se cacher derrière les heuristiques utilisées que nous avons commencé nos recherches sur la saillance.

Nous nous sommes aussi intéressé aux méthodes 2D reposant sur le "Stochastic Completion Field", SCF en abrégé (voir section 8.5.2) : leur particularité est en-effet de donner une définition de la saillance véritablement basée sur un modèle probabiliste des courbes.

8.1.3 Approche et organisation du chapitre

Nous allons ici prolonger ces différents travaux sur la saillance. On s'attachera surtout à dégager les idées principales qui pourraient, à l'avenir, permettre la mise en place d'une théorie unifiée et plus générale. Pour ne pas surcharger ce chapitre, nous utiliserons les modèles les plus simples (en général basés sur des gaussiennes), et, dans le but de faciliter leur compréhension et leur utilisation, nous essayerons d'obtenir des formules simples, quitte à faire quelques approximations.

Notre démarche suivra les grandes idées présentées au début de la deuxième partie de cette thèse (voir page 155) :

- Nous verrons dans un premier temps (section 8.2) que nous travaillerons avec des nuages de *points semi-orientés non structurés* (pour les courbes, les points et la direction de la tangente de la courbe en ce point) ou de surface (un point et la direction de la normale à la surface en ce point). Nous verrons alors comment assurer l'invariance par transformation rigide, et comment représenter la probabilité d'un point semi-orienté connaissant un autre point semi-orienté de la courbe ou la surface.
- Nous présenterons ensuite un premier exemple de modèle (section 8.3), qui montre comment on peut calculer la probabilité qu'une primitive appartienne à la surface,

lorsqu'on a mesuré une (et une seule) autre primitive appartenant à la même surface. On utilisera pour cela trois sous-modèles, qui décrivent *l'échantillonnage*, *la forme de la surface* et le *bruit de mesure*.

- Nous détaillerons ensuite le rôle et les différentes variantes de ces trois sous-modèles : *échantillonnage* (section 8.4), *forme de la surface* (section 8.5) et *bruit de mesure* (section 8.6). Nous obtiendrons ainsi un modèle théorique général.
- Nous comparerons et ajusterons ce modèle général aux lois empiriques observées sur les données réelles (section 8.7).
- Nous verrons, dans les grandes lignes, comment utiliser ces modèles pour adapter les applications qui nous ont inspiré, en leur donnant une interprétation purement statistique (section 8.8). Ceci sera en particulier l'occasion de voir comment calculer et utiliser la probabilité d'une primitive lorsqu'on connaît non pas une mais *plusieurs* primitives déjà mesurées sur la surface.
- Nous ferons enfin le lien avec la première partie de cette thèse, en incorporant ces modèles dans nos algorithmes de recalage (section 8.9).

8.2 Représentation des courbes et surfaces échantillonnées

Nous souhaitons donc travailler sur une courbe ou une surface échantillonnée, i.e. connue via un certain nombre de primitives représentant des mesures bruitées (voire aberrantes) d'éléments de cette courbe ou surface. Il faut donc avant tout se demander sur quel type de représentation nous allons nous baser par la suite. Nous introduirons cette section en expliquant le choix des points semi-orientés (i.e. muni d'un vecteur unitaire non-orienté) et non structurés comme primitive de travail (section 8.2.1). Nous adapterons alors certains résultats du chapitre précédent pour les points semi-orientés : nous étudierons leur géométrie vis-à-vis de l'ensemble des transformations rigides (section 8.2.2), et verrons comment représenter une loi de probabilité pour un point semi-orienté aléatoire par un champ de tenseur (section 8.2.3).

8.2.1 Type de primitive utilisée

Nous allons commencer par nous interroger sur les différentes représentations possibles des surfaces échantillonnées. Elles peuvent se distinguer par le type de primitive et le type de structuration utilisés.

La structuration indique comment on relie les primitives entre elles pour former des morceaux de surface. Il peut s'agir par exemple d'une triangulation. Mais celle-ci peut être complètement faussée (et donc très trompeuse) dès qu'elle inclut des points aberrants. De plus, certains nuages que nous utilisons (les pointés per-opératoires) ne sont absolument pas structurés. Nous ne chercherons donc pas ici à utiliser la structuration lorsqu'elle celle-ci est présente.

Il ne nous reste donc que l'information donnée par les primitives qui ont été mesurées sur la courbe ou la surface. Cette information peut-être de plusieurs ordres différents :

- A l'ordre 0 : on ne considère alors que la position du point.
- A l'ordre 1 : on considère, en plus de la position du point, l'espace tangent au point, et donc la direction de la tangente à la courbe ou de la normale à la surface.
- A l'ordre 2 : on considère alors, en plus de la position et de l'orientation, les variations d'orientation, i.e. la courbure de la courbe ou les courbures de la surface.

Nous souhaitons aller au delà de l'ordre 0. En-effet, en l'absence de structuration, la position d'un unique point ne donne aucune information sur la forme locale de la surface (le point indique juste que la surface passe par là). Il faut donc regarder la position relative de plusieurs points pour obtenir une telle information. C'est par exemple le principe des travaux basés sur les voisins naturels [Amenta and Bern, 1999; Boissonnat and Cazals, 2000] qui permet en fait de structurer les points préalablement à toute étude. Mais ces travaux sont fondamentalement non probabilistes (ils ne prennent pas en compte un éventuel bruit) et en pratique très sensibles aux points aberrants.

Aussi souhaitons-nous prendre au moins en compte l'information à l'ordre 1. En-effet, un seul point semi-orienté permet alors de connaître l'espace tangent et donne donc une première information sur la forme locale de la surface[24]. Les travaux qui l'utilisent, notamment ceux basés sur la notion de saillance, ont donné des résultats déjà plus pertinents en présence de bruit, et se rapprochent donc de nos préoccupations. Rappelons que, si les normales ne sont pas disponibles dans les données, nous disposons déjà de techniques permettant de les interpoler (section 7.4 et 8.8.3).

Quant aux informations d'ordre 2, elles permettent de véritablement connaître la forme locale de la surface, et permettraient donc d'affiner nos modèles. Mais nous n'en disposons pas sur nos nuages. Il existe bien des techniques permettant de les interpoler (nous reviendrons sur ce point à la section 8.8.3), mais nous avons préféré nous limiter ici à l'ordre 1. L'utilisation de l'ordre 2 ou d'ordres supérieurs restera pour nous une perspective.

8.2.2 Invariance par transformation rigide

Nous souhaitons, avec nos points orientés, former une théorie invariante par transformation rigide. En pratique, nous utiliserons la condition d'invariance sur une fonction sca-

24. Notre utilisation du terme forme pour désigner l'emplacement de la surface est un peu abusive. On utilise en-effet généralement ce terme pour désigner les invariants de la surface, et donc en particulier les informations d'ordre 2, mais pas les informations d'ordre 1 (l'espace tangent tourne lorsqu'on applique une rotation, et n'est donc pas invariant).

laire de deux points orientés $g((m,n_m),(s,n_s))$ (par exemple la distance $d((m,n_m),(s,n_s))$ ou la probabilité conditionnelle $p((s,n_s)|(m,n_m))$), qui devra donc suivre la propriété suivante :

$$\forall T \qquad g((m,n_m),(s,n_s)) = g(T \star (m,n_m),T \star (s,n_s))$$

On va assurer cette propriété en l'utilisant pour réduire au maximum le domaine de définition de g : on commence par appliquer la translation $t = -m$ pour ramener m en l'origine. On peut alors se contenter de définir g pour $m = 0$. On calculera ensuite g pour un point m quelconque avec $g((m,n_m),(s,n_s)) = g((0,n_m),(s - m,n_s))$. On peut de même ramener la normale n_m sur la normale origine n_0, en appliquant une rotation adaptée[25] (voir figure 8.1 gauche).

En pratique, utiliser une transformation rigide pour ramener (m,n_m) sur l'origine revient à travailler dans le repère d'origine m et dont l'un des axes est indiqué par n_m. Nous travaillerons par la suite dans ce repère, avec les axes suivants :

- Pour les courbes, notre convention est d'aligner la tangente t_m sur l'axe x. C'est donc y (ou $y' = \sqrt{y^2 + z^2}$ en 3D) qui donne la norme de la composante de ms perpendiculaire au plan tangent, et x qui donne la norme de la composante dans l'espace tangent.
- Pour les surfaces, notre convention est d'aligner la normale n_m sur l'axe y. C'est donc y qui donne la norme de la composante de ms perpendiculaire au plan tangent, et $x' = \sqrt{x^2 + z^2}$ qui donne la norme de la composante dans l'espace tangent.

Pour les surfaces, on peut encore ramener le point $s = (x,y,z)$ dans le plan xy, en appliquant la rotation d'axe y (qui laisse donc n_m inchangé) et d'angle $-atan2(z,x)$. Pour les courbes en 3D, on ramènera de même s dans le plan xy, en appliquant la rotation d'axe x (qui laisse donc t_m inchangé) et d'angle $-atan2(z,y)$ (voir figure 8.1 droite).

Dans le cas semi-orienté, on peut enfin se ramener au demi-plan $x \geq 0$ en appliquant la symétrie autour de l'origine en 2D et autour de n_0 en 3D.

En conclusion, on peut se ramener à l'étude de g pour (m,n_m) situé en l'origine et s situé dans le plan xy. Ensuite, pour une paire de points (m,n_m) et (s,n_s) quelconques, on cherche la transformation rigide T^{-1} qui ramène (m,n_m) sur l'origine et s dans le plan xy, et on peut calculer la fonction g pour cette paire de point :

$$g((m,n_m),(s,n_s)) = g((0,n_0),T^{-1} \star (s,n_s))$$

L'invariance par transformation rigide est ainsi assurée par construction.

Il faudra parfois considérer les autres symétries du problème, comme les symétries par rapport à la droite n_0 en 2D et par rapport aux plans contenant n_0 en 3D, qui pourront

25. Comme $R^{-1}_{n_0 n_m}$, l'inverse de la fonction placement canonique pour les normales, d'axe perpendiculaire à n_0 et n_m et d'angle $(\widehat{n_0,n_m})$, voir section 7.2

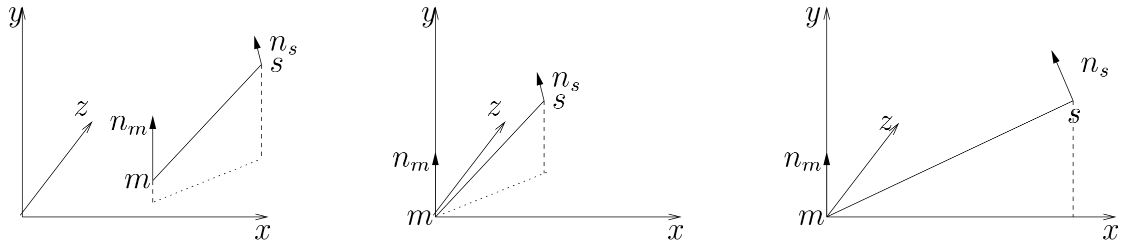


FIG. 8.1 – *Construction de la transformation rigide permettant de ramener (m,n_m) à l'origine et s dans la plan xy :*
On applique d'abord la translation permettant de ramener m à l'origine et la rotation permettant d'aligner n_m sur l'axe y (passage de la figure de gauche à la figure du milieu). Puis on applique la rotation autour de n_m permettant de ramener s dans le plan xy (passage de la figure du milieu à la figure de droite).

permettre de restreindre encore le domaine de définition de g à $y \geq 0$. Ces symétries pourront aussi permettre de trouver des conditions nécessaires supplémentaires sur g.

Remarque pour l'implémentation

En fin de compte, on a appliqué la transformation :

$$T^{-1} = (R_{n_0.atan2(z,y)},0)^{-1} \circ (R_{n_0 n_m},m)^{-1} \quad \text{avec}\, T = (R_{n_0 n_m},m) \circ (R_{n_0.atan2(z,y)})$$

et on obtient le point $(s',n_{s'})$:

$$s' = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} ((s-m)|n_m) \\ \sqrt{(s-m)^2 - ((s-m)|n_m)^2} \\ 0 \end{pmatrix}$$

$$n_{s'} = \begin{pmatrix} \cos\theta \\ \sin\theta.\cos\alpha \\ \sin\theta.\sin\alpha \end{pmatrix} \quad \theta = \left(\widehat{n_m,n_s}\right) \quad \alpha = \left(\widehat{\Pi_{\perp n_m}(s-m),\Pi_{\perp n_m} n_s}\right)$$

et on peut finalement calculer g :

$$g((m,n_m),(s,n_s)) = g(T \star (0,n_0),T \star (s',n_{s'})) = g((0,n_0),(s',n_{s'}))$$

Pour finir, on peut calculer les dérivées de g, qui seront par exemple nécessaires pour implémenter une descente de gradient. On montre rapidement que :

$$\frac{\partial g((m,n_m),(s,n_s))}{\partial(s,n_s)} = R\frac{\partial g((0,n_0),(s',n_{s'}))}{\partial(s',n_{s'})}$$

$$H_{(s,n_s)}g((m,n_m),(s,n_s)) = R.Hg(0,n_0),(s',n_{s'})).R^t$$

où $R = R_{n_0 n_m} \circ R_{n_0.atan2(z,y)}$ est la partie rotation de la fonction T. Enfin, en 3D, on travaille avec $z=0$, ce qui impose au niveau des dérivés, en considérant toutes les symétries du problème:

$$\frac{\partial g}{\partial z} = 0 \qquad Hg = \begin{pmatrix} \frac{\partial^2 g}{\partial x^2} & \frac{\partial^2 g}{\partial x \partial y} & 0 \\ \frac{\partial^2 g}{\partial x \partial y} & \frac{\partial^2 g}{\partial x^2} & 0 \\ 0 & 0 & \frac{\partial g}{\partial y}/y \end{pmatrix}$$

On pourra ainsi se contenter de calculer les dérivés par rapport à x et y.

8.2.3 Point semi-orienté aléatoire et champ de tenseurs

Nous allons manipuler dans ce chapitre des points semi-orientés aléatoires. Un tel point sera alors représenté par une loi de probabilité $p(s,n_s)$ sur sa position et son orientation. Pour l'interpréter, nous la diviserons souvent en :

– la loi de probabilité marginale sur la seule position du point, donnée en intégrant, pour un point donné, sur toutes les orientations possibles :

$$p(s) = \int_{n_s} p(s,n_s).dn_s$$

– la loi de probabilité sur l'orientation, connaissant la position du point :

$$p(n_s|s) = \frac{p(s,n_s)}{p(s)}$$

La loi pourra alors être retrouvée par :

$$p(s,n_s) = p(n_s|s).p(s)$$

Remarque pour l'implémentation

En pratique, pour stocker une loi de probabilité sur un point semi-orienté aléatoire, il faudrait discrétiser l'espace des points semi-orientés, qui est de dimension $2\mathfrak{D} - 1$ (3 en 2D et 5 en 3D). Cette forte dimension nous obligerait à limiter considérablement la résolution de cette discrétisation pour tenir dans la mémoire des ordinateurs. Or nous avons vu au chapitre précédent que nous pouvons synthétiser dans un seul tenseur symétrique une loi sur une normale aléatoire (ce tenseur permet alors de retrouver la moyenne et la covariance de la normale) ainsi qu'une information supplémentaire (le tenseur est de trace unitaire, on peut donc le multiplier par un scalaire). Dans notre cas, on peut ainsi stocker, pour un point s, la loi de probabilité sur l'orientation $p(n_s|s)$ connaissant le point (qui donne un tenseur $T(n_s|s)$ de trace unitaire) et la probabilité sur la seule position du point $p(s)$ (en multipliant ensuite le tenseur $T(n_s|s)$ par $p(s)$). C'est ce que nous allons faire ici en chaque point de l'espace, ce qui nous permettra de représenter une loi sur l'ensemble des points semi-orientés en dimension $\mathfrak{D}$ par un champ de tenseur de même dimension. Cette astuce est bien-sûr directement inspiré des champs de vote de Médioni.

Soit donc une loi de probabilité $p(s,n_s)$ sur l'ensemble des points semi-orientés. Pour chaque point s, on prend le tenseur associé à la loi sur l'orientation connaissant la position :

$$T(n_s|s) = \int p(n|s).(n.n^t).dn$$

Que l'on multiplie ensuite par la probabilité du point s toutes orientations confondues, pour obtenir le champ de tenseur qui représente $p(s,n)$:

$$T(s) = p(s).T(n_s|s)$$

Remarquons que, comme $p(s).p(n_s|s) = p(s,n_s)$, on peut directement calculer le tenseur $T(s)$ à partir de la loi $p(s,n_s)$:

$$T(s) = \int p(s,n).(n.n^t).dn$$

On peut, à partir de ce tenseur, retrouver la probabilité du point s toutes normales confondues, en extrayant la trace :

$$Tr(T(s)) = \int p(s,n).dn = p(s)$$

et retrouver le tenseur associé à la probabilité de l'orientation connaissant la position, en divisant le tenseur par sa trace :

$$T(n_s|s) = \int \frac{p(s,n_s)}{p(s)}.(n.n^t).dn = \frac{T(s)}{p(s)} = \frac{T(s)}{Tr(T(s))}$$

En résumé, pour un point de l'espace s, la trace du tenseur $T(s)$ représente la probabilité du point, et la forme du tenseur représente la probabilité de la normale en ce point.
Enfin, il faut adapter les résultats de la section précédente : lorsqu'on calculera la probabilité conditionnelle $p((s,n_s)|(m,n_m))$, on utilisera $p((s,n_s)|(m,n_m)) = p(T \star (s',n_{s'})|T \star (0,n_0)) = p((s',n_{s'})|(0,n_0))$. On a alors, au niveau des tenseurs :

$$T(s) = T(T \star s') = R.T(s').R^t$$

où R est la partie rotation de la transformation T.

8.3 Premier exemple : modélisation d'une surface plane

Nous allons maintenant modéliser l'information donnée par la connaissance de la position et l'orientation (m,n_m) d'un point de la surface. Nous le ferons en calculant la loi de probabilité $p((s,n_s)|(m,n_m))$ sur la position et l'orientation potentielle d'une seconde primitive (s,n_s) mesurée sur la même surface. Nous appellerons cette loi *probabilité d'une primitive connaissant une autre primitive.*

Le but est ici de montrer comment on obtient cette loi de probabilité on mélangeant les 3 phénomènes aléatoires (échantillonnage, forme et bruit de mesure). Nous le ferons dans cette section en prenant pour chacun d'eux l'exemple le plus simple (échantillonnage simple, forme affine et bruit de mesure gaussien). Des exemples plus complexes seront ensuite présentés dans les sections suivantes.

Précisons que nous calculerons toutes les lois de probabilités à une constante multiplicative près, pour alléger les notations. Cette constante pourra être retrouvée, lorsqu'elle sera nécessaire (par exemple lorsqu'on utilisera plusieurs modèles différents dans la même application), en normalisant la loi de probabilité.

8.3.1 Prise en compte de la forme et de l'échantillonnage

Nous connaissons donc une première primitive (m,n_m) mesurée sur la surface, et souhaitons en déduire une loi de probabilité $p((s,n_s)|(m,n_m))$ pour une deuxième primitive mesuré sur la même surface. Supposons dans un premier temps qu'il n'y a pas de bruit de mesure. Deux phénomènes aléatoires conditionnent alors le placement de (s,n_s) :

- *La façon dont est échantillonnée la surface* : elle influence *l'éloignement* entre le point s et le point m. Cet éloignement, noté u, sera pour l'instant égal à la distance euclidienne[26]. Avec un échantillonnage uniforme (tel que celui des pointés per-opératoires), aucune distance n'est privilégiée. En revanche, avec un échantillonnage régulier (tel que celui du Picza, où les points sont répartis sur une grille), des distances privilégiées peuvent apparaître (en fonction du pas de la grille).

– *La forme de la surface* : elle influence la *dispersion* de (s,n_s) autour du plan tangent donné par (m,n_m), en fonction de l'éloignement entre le point (s,n_s) et le point (m,n_m). Pour une surface plane, les points (s,n_s) devront rester sur le plan tangent. Pour une surface courbe, il pourront se disperser de plus en plus autour du plan tangent lorsqu'on s'éloigne de (m,n_m).

Il nous faut modéliser ces deux phénomènes. Nous le ferons toujours par le biais de deux lois de probabilités :

– Pour l'échantillonnage : il s'agira d'une loi de probabilité $p_{\mathbf{u}}(u)$ sur l'éloignement entre les deux primitives.
– Pour la forme de la surface : il s'agira d'une loi de probabilité $p((s,n_s)|(m,n_m),u)$ qui indique la répartition des points (s,n_s) en fonction de (m,n_m) pour un éloignement u connu.

Ces deux éléments sont mélangés via le théorème des probabilités totales, pour tenir compte de tous les éloignements possibles :

$$p\left((s,t_s)|(m,n_m)\right) = \int p\left((s,t_s)|(m,n_m),u\right).p_{\mathbf{u}}(u).du$$

On obtient ainsi la première version (sans bruit de mesure) de la loi d'une primitive connaissant une autre primitive. Nous allons maintenant l'illustrer avec un modèle d'échantillonnage quelconque et un modèle de forme affine. Nous verrons ensuite comment rajouter le bruit de mesure.

8.3.2 Exemple de l'approximation affine

Le modèle de forme le plus simple est le sous espace affine (droite en 2D, droite ou plan en 3D). Par modèle, il faut entendre que nous allons supposer que notre courbe ou surface est un sous espace affine (que notre courbe est une droite, que notre surface est un plan) et qu'elle passe par le point (m,n_m) que nous connaissons. Bien-sûr, il ne faut pas s'attendre avec un modèle si simple à de bons résultats lorsqu'on travaillera avec des courbes ou surfaces véritablement courbées. Il pourra par contre suffire sur des surfaces

26. On verra plus loin que l'éloignement u correspond en-fait à la distance sur la surface, i.e. la longueur de la géodésique entre les points s et m. Dans le cas affine, la géodésique suit une ligne droite, et on retrouve donc la distance euclidienne. Ensuite, plus la surface est courbe et plus leur longueur et donc l'éloignement augmente. Mais nous garderons alors souvent la distance euclidienne comme approximation.

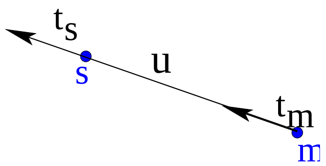
relativement planes (comme l'os de la mâchoire) voire complètement planes (comme les pièces usinées qui sont souvent utilisées pour les études métrologiques).

Pour le modèle d'échantillonnage, on peut d'ores et déjà considérer une loi $p_{\mathbf{u}}$ quelconque[27].

8.3.2.1 Droite déterministe

Supposons donc que nous sommes en présence d'une droite passant par m et dont la direction t_m est connue.

Éloignement connu On va d'abord envisager le cas le plus simple, celui où l'éloignement u est connu. L'éloignement correspond ici à la distance (signée) entre les deux points m et s :



On sait qu'on est parti de m dans la direction t_m, et qu'on n'a pas changé de cap (puisqu'on est sur une droite). On arrive donc au point s, donné par $ms = u.t_m$, et l'orientation en ce point reste inchangée. On a donc :

$$(s,t_s) = (m + u.t_m,t_m)$$

qui se ré-exprime par la loi de probabilité suivante :

$$p((s,t_s)|(m,t_m),u) = \delta_u(\|ms\|).\delta_{u.t_m}(ms).\delta_{t_m}(t_s)$$

Cette loi ne laisse en-fait que deux possibilités équiprobables, $(s,t_s) = (m + u.t_m,t_m)$ et le symétrique $(s,t_s) = (m - u.t_m,t_m)$.

Remarque pour l'implémentation

Cette loi est représentée par le champ de tenseur suivant :

$$T(s|(m,n_m),u) = \delta_u(\|ms\|).\delta_{u.t_m}(ms).t_m.t_m^t$$

qui indique à la fois que la direction est imposée $t_s = t_m$, et que la point est imposé par $ms = u.t_m$.

27. Le choix de $p_{\mathbf{u}}$, discuté en détail dans la section 8.4, dépend surtout de l'application. Pour le recalage, on utilisera en général un modèle gaussien, qui permet de défavoriser les points plus éloignés, pour lesquels l'approximation plane est moins bonne.

Éloignement quelconque On effectue maintenant le mélange entre la loi sur l'échantillonnage $p_{\mathbf{u}}(u)$ et la loi sur la forme $p((s,t_s)|(m,n_m),u)$:

$$p((s,t_s)|(m,t_m)) = \int p((s,t_s)|(m,t_m),u).p_{\mathbf{u}}(u)du = p_{\mathbf{u}}(\|ms\|).\delta_{\|ms\|t_m}(ms).\delta_{t_m}(t_s) \quad (8.1)$$

On obtient ainsi la loi sur la primitive connaissant une primitive, qui montre que les primitives (s,t_s) sont réparties sur la droite donnée par (m,t_m) avec un éloignement qui suit la loi indiquée indiquée par $p_{\mathbf{u}}$.

8.3.2.2 Plan déterministe

Supposons maintenant que nous sommes en présence d'un plan (ou plus généralement d'un hyperplan) donné par un premier point orienté (m,n_m) connu.

Éloignement connu Comme dans le cas précédent, la normale reste inchangée. On a donc déjà $n_s = n_m$, et le point s doit vérifier $(ms|n_m) = 0$ pour être situé dans le plan (m,n_m). Lorsque nous connaissons l'éloignement $u = \|ms\|$, il devient aisé de savoir où peut se trouver le point s. En 3D, les points s potentiels se répartissent sur un cercle (de centre m, de rayon u et situé dans le plan perpendiculaire à n_m). De plus, la répartition, en terme de probabilité, doit être uniforme sur ce cercle, pour respecter l'invariance par transformation rigide (ici les rotations autour de l'axe (m,n_m), d'angles quelconques).

On obtient donc :

$$p\left((s,n_s)|(m,n_m),u\right) = \delta_u\left(\|ms\|\right).\delta_0\left((n_m|ms)\right).\delta_{n_m}(n_s)$$

Éloignement quelconque On peut maintenant tenir compte des différents éloignements possibles :

$$p((s,t_s)|(m,t_m)) = \int p((s,t_s)|(m,t_m),u).p_{\mathbf{u}}(u)du = p_{\mathbf{u}}(\|ms\|).\delta_0\left((n_m|ms)\right).\delta_{n_m}(n_s) \quad (8.2)$$

Remarque sur le calcul précédent

Le calcul qui précède cache en-fait une petite complication. En-effet, la constante de normalisation pour la loi à éloignement connu n'est pas fixe, mais varie en fonction de la circonférence du cercle (et donc en fonction de de l'éloignement $u = \|ms\|$). Il faudrait donc écrire :

$$p'\left((s,n_s)|(m,n_m),u\right) = \frac{\delta_u\left(\|ms\|\right).\delta_0\left((n_m|ms)\right).\delta_{n_m}(n_s)}{\|ms\|}$$

Si on mélangeait maintenant avec une loi uniforme sur l'éloignement, on obtiendrait :

$$\frac{\delta_0\left((n_m|ms)\right).\delta_{n_m}(n_s)}{\|ms\|}$$

Cette densité de probabilité décroît lorsqu'on s'éloigne du point connu. Mais ce n'est pas là ce qu'on attendait avec un échantillonnage uniforme.
Pour résoudre ce problème, il faut modifier aussi la loi de probabilité sur l'éloignement. Le nombre de point situés à un éloignement u est, sur le plan, proportionnel à u. On a donc u fois plus de chances d'échantillonner un point à cet éloignement, et la loi sur u devient : $p'_{\mathbf{u}}(u) = u.p_{\mathbf{u}}(u)$. Lorsqu'on mélange les deux lois corrigées, la division par $\|ms\|$ et la multiplication par u se compensent, et on obtient bien :

$$p((s,t_s)|(m,t_m)) = \int p'((s,t_s)|(m,t_m),u).p'_{\mathbf{u}}(u)du = p_{\mathbf{u}}(\|ms\|).\delta_0\left((n_m|ms)\right).\delta_{n_m}(n_s)$$

8.3.3 Ajout d'un bruit de mesure sur la position du point - points semi-anisotropes

Nous avons supposé jusqu'à maintenant que le point (m,n_m) était parfaitement connu. Mais en pratique, les points que nous mesurons sur une surface sont corrompus par un bruit, dû à la méthode d'acquisition, que ce soit l'outil qui a servi à mesurer les points (par exemple les nuages per-opératoires) ou l'algorithme qui a servi à segmenter la surface (par exemple à partir du scanner). Nous allons maintenant prendre en compte le bruit sur la position du point m. Nous apporterons ainsi une nouvelle justification aux bruits semi-anisotropes utilisés pour le recalage dans la section 5.3.

Supposons donc que l'emplacement du point m est corrompu par un bruit additif e_m qui suit une loi de probabilité $p_{\mathbf{m}}(e_m)$ [28]. On applique alors tout simplement le théorème des probabilités totales :

$$p\left((s,n_s)|(m,n_m)\right) = \int_m p((s,n_s)|(m+e_m,n_m)).p_{\mathbf{m}}(e_m).de_m$$

Mais on peut toujours utiliser la translation de vecteur $-e_m$ pour ramener le point bruité en m, et se ramener ainsi à la loi connaissant (m,n_m) :

$$p\left((s,n_s)|(m,n_m)\right) = \int_m p((s-e_m,n_s)|(m,n_m)).p_{\mathbf{m}}(e_m).de_m = p\left((s,n_s)|(m,n_m)\right) * p_{\mathbf{m}}$$

On aura bien-sûr reconnu le produit de convolution.

Ce résultat est intéressant car il donne une base théorique aux bruits semi-anisotropes présentés à la section 5.3. En-effet, nous composons là :

- Un bruit conséquence de l'échantillonnage, modélisé par la probabilité sur l'éloignement $p_{\mathbf{u}}$, et de la forme plane de la surface. Ce bruit est tangent à la courbe ou situé dans le plan tangent à la surface, et dans ce dernier cas, il doit être isotrope dans le plan (pour respecter l'invariance par transformation rigide). Avec un modèle

28. Nous utiliserons en général un modèle gaussien isotrope, car il est bien adapté à la plupart des outils utilisés pour mesurer un point sur une surface.

gaussien pour $p_{\mathbf{u}}$[29], cela se traduira par une loi gaussienne sur s avec une covariance de la forme $\Sigma = \sigma_u^2.(t_m.t_m^t)$ pour une courbe et $\Sigma = \sigma_u^2.(Id - n_m.n_m^t)$ pour un plan.

– Un bruit de mesure, modélisé par p_m. On utilisera en général un bruit blanc additif gaussien et isotrope de variance σ_m^2.

Dans le cas simple (gaussien isotrope), on obtient donc finalement pour s un bruit gaussien de variance $\sigma_n^2 = \sigma_m^2$ dans l'axe de la normale et $\sigma_t^2 = \sigma_m^2 + \sigma_u^2$ dans le plan tangent. On justifie donc ainsi la définition de nos points semi-anisotropes.

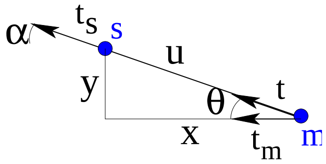
8.3.4 Ajout d'un bruit de mesure sur l'orientation

Nous allons maintenant revenir à un point m parfaitement connu pour examiner le cas où c'est l'orientation qui est mal connue. Nous conclurons en montrant comment tenir à la fois compte d'un bruit sur l'orientation et la position.

8.3.4.1 Prise en compte d'un bruit sur l'orientation - cas d'une droite

On peut aborder le problème de la même façon que pour le bruit sur la position. On obtient alors une convolution du modèle déterministe par rapport à la loi du bruit sur l'orientation. Mais les calculs sont assez techniques, à cause de l'aspect non-linéaire des orientations. Nous allons donc présenter une autre approche, strictement équivalente mais plus directe, en supposant directement que l'orientation au point connu suit une loi centrée autour de la valeur mesurée et représentative du bruit (dans le cas d'un bruit gaussien additif et centré, il s'agira donc d'une loi gaussienne centrée), et en re-dérivant notre modèle en prenant cette loi en compte.

Supposons donc que nous sommes en présence d'une droite passant par m et dont la direction $\mathbf{t}$ suit une loi probabiliste centrée autour de t_m. On raisonne d'abord à éloignement u connu. Nous travaillerons dans un premier temps en 2D et en coordonnées polaires : les coordonnées du point semi-orienté (s,t_s) seront les coordonnées polaires (r,θ,α) autour de (m,t_m) :



La loi de probabilité sur l'orientation effective de la droite (égale à t_s) sera donnée au travers d'une densité de probabilité notée p_θ sur l'angle θ entre t_s et l'orientation au point

29. Rappelons que le choix de $p_{\mathbf{u}}$, discuté en détail dans la section 8.4, dépend surtout de l'application. Pour le recalage, on utilise en général un modèle gaussien, qui permet de défavoriser les points plus éloignés, pour lesquels l'approximation plane est moins bonne.

connu t_m. La probabilité du point est alors :

$$p(r,\theta,\alpha|u) = \delta_u(r).p_\theta(\theta).\delta_\theta(\alpha)$$

que nous pouvons ré-écrire en coordonnées cartésiennes[30] :

$$p(x,y,\alpha|u) = \delta_u(r).\frac{p_\theta(\theta)}{r}.\delta_\theta(\alpha) \simeq \delta_u(r).\frac{p_\theta(y/r)}{r}.\delta_\theta(\alpha)$$

Soit, en utilisant directement (s,t_s) et la loi de probabilité sur t_m donnée par la densité de probabilité $p_{\mathbf{t}_m}$, et en généralisant en dimension $\mathfrak{D}$ quelconque[31] :

$$p((s,t_s)|(m,t_m),u) = \delta_u(\|ms\|).\frac{p_{\mathbf{t}_m}(t_s)}{\|ms\|^{\mathfrak{D}-1}}.\delta_{ms/\|ms\|}(t_s)$$

On prend ensuite en compte la loi a-priori sur les éloignements :

$$p(x,y,\alpha) = p_u(r).\frac{p_\theta(\theta)}{r^{\mathfrak{D}-1}}.\delta_\theta(\alpha) \simeq p_u(r).\frac{p_\theta(y/r)}{r^{\mathfrak{D}-1}}.\delta_\theta(\alpha) \tag{8.3}$$

$$p\left((s,t_s)|(m,t_m)\right) = p_{\mathbf{u}}(\|ms\|).\frac{p_{\mathbf{t}_m}(t_s)}{\|ms\|^{\mathfrak{D}-1}}.\delta_{ms/\|ms\|}(t_s)$$

8.3.4.2 Prise en compte d'un bruit sur l'orientation - cas d'un plan

Supposons maintenant que nous sommes en présence d'un plan (ou plus généralement d'un hyperplan) passant par m et dont la normale $\mathbf{n}$ suit une loi de densité $p_{\mathbf{n}}$.

On obtient, pour un éloignement connu (le détail des calculs est donné dans l'annexe F.1) :

$$p\left((s,n_s)|(m,n_m),u\right) = \delta_u(\|ms\|).\frac{\delta_0((n_s|ms))}{\|ms\|}.p_{\mathbf{n}_m}(n_s)$$

On prend ensuite en compte la loi a-priori sur les éloignements :

$$p\left((s,n_s)|(m,n_m)\right) = p_{\mathbf{u}}(\|ms\|).\frac{\delta_0((n_s|ms))}{\|ms\|}.p_{\mathbf{n}_m}(n_s) \tag{8.4}$$

8.3.4.3 Prise en compte des deux bruits

On peut enfin prendre en compte à la fois un bruit sur l'orientation et la position, lorsqu'ils sont indépendants. Il suffit en-effet de prendre en compte le bruit sur l'orientation

30. On a $r = \sqrt{x^2+y^2}$ et $\theta = \arcsin(y/r)$ (qu'on peut approximer par y/r), et le jacobien du passage entre les deux systèmes de coordonnées est le coefficient d'échelle $1/r$.

31. La seule différence en dimension supérieure à 2 est alors dans le jacobien, i.e. le coefficient d'échelle qui est de la forme $r^{\mathfrak{D}-1}$.

(cf cette section), puis de lui appliquer la convolution avec le bruit sur la position (section précédente). Nous y reviendrons dans la section consacrée aux bruits de mesure, préférant ici tirer les conclusions de notre premier exemple.

8.3.5 Discussion

Nous avons montré, sur un exemple simple, la démarche que nous suivrons systématiquement pour construire nos modèles :

- Modéliser l'échantillonnage, par le biais d'une loi $p_{\mathbf{u}}(u)$ sur l'éloignement.
- Modéliser la forme de la courbe ou la surface, par la biais d'une loi $p((s,n_s)|(m,n_m),u)$ sur la position d'une primitive, connaissant une autre primitive et l'éloignement qui les sépare.
- Mélanger ces deux lois par le biais du théorème des probabilités totales :

$$p((s,t_s)|(m,t_m)) = \int p((s,t_s)|(m,t_m),u).p_{\mathbf{u}}(u)du$$

- Modéliser le bruit de mesure en orientation par une loi $p_{\mathbf{n}_m}(n)$ centrée autour de l'orientation du point d'entrée n_m, et adapter le modèle précédent.
- Modéliser le bruit de mesure en position par une loi additive $p_{\mathbf{m}}(e_m)$ et la convoluer au modèle précédent :

$$p((s,t_s)|(m,t_m)) = p((s,t_s)|(m,t_m)) * p_{\mathbf{m}}$$

Cette démarche est celle que nous suivrons systématiquement par la suite. Il faudra cependant se rappeler qu'elle fait implicitement l'hypothèse que les bruits de mesures sont indépendants de l'échantillonnage et la forme. Ceci ne sera pas toujours vrai (nous illustrons ce problème sur l'exemple de l'échantillonnage du Picza, section 8.4.4), mais l'adaptation de notre démarche à ces cas problématiques ne restera pour nous qu'une perspective.

Pour finir sur notre premier exemple, rappelons la formule finale :

$$p\left((s,t_s)|(m,t_m)\right) = \left(p_{\mathbf{u}}(\|ms\|).\frac{\delta_{\|ms\|.t_m}(ms)}{\|ms\|^{\mathfrak{D}-1}}.p_{\mathbf{t}_m}(t_s)\right) * p_{\mathbf{m}}$$

pour une droite en dimension $\mathfrak{D}$, et :

$$p\left((s,n_s)|(m,n_m)\right) = \left(p_{\mathbf{u}}(\|ms\|).\frac{\delta_0((n_s|ms))}{\|ms\|}.p_{\mathbf{n}_m}(n_s)\right) * p_{\mathbf{m}}$$

pour un plan en 3D. Celle-ci montre comment on mélange la loi sur l'échantillonnage $p_{\mathbf{u}}$, la loi sur la forme $\delta_{\|ms\|.t_m}(ms)$ ou $\delta_0((n_s|ms))$, et les lois sur le bruit de mesure $p_{\mathbf{m}}$ et

$p_{\mathbf{n}_m}$.

8.4 Loi sur l'éloignement - Modèle d'échantillonnage

Nous allons maintenant détailler, une par une, les trois composantes de notre modèle : l'échantillonnage, la forme et le bruit de mesure. La loi sur l'échantillonnage joue un rôle un peu spécial, car elle dépend beaucoup du contexte.

Ainsi, lorsqu'on cherche à identifier le modèle d'échantillonnage (i.e. la loi sur l'éloignement entre les points) sur une surface, on pourra trouver :

- Une loi uniforme, comme avec les pointés per-opératoires.
- Une loi beaucoup plus complexe, lorsque l'échantillonnage suit un schéma régulier (comme avec les scanners ou les Picza).

En revanche, lorsqu'on veut utiliser nos modèles dans un algorithme, on peut utiliser une loi qui ne corresponde pas nécessairement à une façon d'échantillonner la surface (comme les précédentes), mais réponde à un besoin particulier de notre application :

- Une loi gaussienne pour le recalage : l'utilisation d'une telle loi dans le recalage permet de réduire l'importance accordée aux points plus éloignés.
- Une loi exponentielle pour la complétion de courbes : cette loi se justifie mieux avec les modèles de forme stochastiques.

Nous allons ici détailler ces différentes lois.

Remarque

Comme nous l'avions déjà fait en dérivant le modèle plan (section 8.3.2.2), nous masquerons ici le problème de la définition de la loi d'échantillonnage pour les surfaces.

Rappelons que le nombre de point situés à un éloignement u est, sur une surface, proportionnel à u. On a donc u fois plus de chances d'échantillonner un point à cet éloignement, et la loi sur u devient : $p'_{\mathbf{u}}(u) = u.p_{\mathbf{u}}(u)$. Inversement, la constante de normalisation pour la loi à éloignement connu n'est pas fixe, mais varie en fonction de la circonférence du cercle (et donc en fonction de de l'éloignement $u = \|ms\|$). Il faudrait donc écrire : $p'\left((s,n_s)|(m,n_m),u\right) = \frac{\delta_u(\|ms\|).\delta_0((n_m|ms)).\delta_{n_m}(n_s)}{\|ms\|}$. Lorsqu'on mélange les deux lois corrigées, la division par $\|ms\|$ et la multiplication par u se compensent, et on obtient bien :

$$p((s,t_s)|(m,t_m)) = \int p'((s,t_s)|(m,t_m),u).p'_{\mathbf{u}}(u)du = \int p((s,t_s)|(m,t_m),u).p_{\mathbf{u}}(u)du$$

En pratique, ce raccourci permet d'utiliser la même loi $p_{\mathbf{u}}$ pour les courbes et les surfaces, et de lui donner une signification plus naturelle. Ainsi, dans le cas d'un échantillonnage *uniforme* (voir ci-dessous) sur une surface, on retrouve un $p_{\mathbf{u}}$ *uniforme*, bien que la véritable probabilité $p'_{\mathbf{u}}(u) = u.p_{\mathbf{u}}(u)$ d'échantillonner un point à un éloignement u augmente proportionnellement à u.

8.4.1 Loi uniforme

La loi uniforme permet de modéliser le cas où aucun éloignement entre deux points n'est privilégié.

On peut rencontrer cette loi sur de véritables surfaces, lorsque l'échantillonnage est non-corrélé : l'endroit où l'on va échantillonner un nouveau point ne dépend pas des points déjà échantillonnés. Cette démarche peut par exemple être suivie pour prendre les pointés per-opératoires sur les dents.

On peut aussi retrouver une loi uniforme lorsqu'on cherche à modéliser le position des points d'un nuage de points S mesuré sur une surface en fonction de la position des points d'un second nuage M mesuré sur la même surface (par exemple lorsqu'on cherche à les recaler). Pour peu qu'il n'y ait pas de corrélations entre la façon d'échantillonner les deux nuages, aucune distance particulière ne pourra être privilégiée entre un point s de S et un point m de M, et on n'aura donc bien une répartition uniforme des éloignements.

Pour les applications, nous voudrions ici suggérer une perspective, bien que nous ne sachions pas si elle est véritablement applicable. Elle concerne le ré-échantillonnage de surface. L'idée est que lorsque la densité de l'échantillonnage du nuage d'entrée est hétérogène, on pourrait essayer d'imposer en sortie une loi d'échantillonnage uniforme, pour obtenir une densité de points homogène. C'est d'ailleurs un peu ce que nous avons fait dans la décimation par les sphères lorsque nous avons imposé un poids uniforme unitaire aux points provenant de chaque sphère (voir section 4.4.3).

8.4.2 Loi gaussienne

La loi gaussienne se rencontre rarement sur des surfaces réelles. On peut tout de même donner l'exemple où l'on dispose d'un modèle de la surface et on demande à un opérateur de venir re-mesurer sur la surface, avec une pointe de mesure, certains points indiqués sur le modèle. L'opérateur ne placera jamais la pointe exactement à l'endroit voulu, et mesurera donc un point de la surface placé à côté du point du modèle. L'éloignement entre le point du modèle et le point effectivement mesuré suit alors une loi qui dépend de l'opérateur. Dans la plupart des cas, une loi gaussienne semble bien indiquée $p_{\mathbf{u}}(u) = \exp\left(-u^2/2\sigma_u^2\right)$.

La loi gaussienne pourra en revanche être souvent utilisée dans les applications. Elle permet alors de défavoriser les points éloignés, pour lesquelles notre modèle est moins bon, d'une part parce que la surface ne se comporte pas forcément comme le prévoit le modèle de forme, d'autre part parce que le nombre de points aberrants augmente. Pour choisir σ_u, il faudra donc faire un compromis entre la distance à laquelle le modèle de forme devient incorrect (dans le cas d'un modèle affine, la distance pour laquelle les points commencent à s'éloigner du plan tangent, à cause de la courbure de la surface), la distance à laquelle commencent à apparaître des points aberrants, et l'intérêt de l'application (qui fonctionne souvent d'autant mieux que le nombre de points utilisé- et donc σ_u - est élevé).

8.4.3 Loi exponentielle

La loi exponentielle est souvent utilisée dans le cas des courbes, lorsqu'on utilise un modèle de forme stochastique (voir [Williams and Jacobs, 1997] ou la section 8.5.2 pour plus de détails). En deux mots, on modélise dans ce cas nos courbes comme les trajectoires de particules qui suivent des marches aléatoires orientées. On veut alors pouvoir favoriser les trajectoires les plus courtes (i.e. les plus directes) au détriment des trajectoires plus longues. L'idée est d'introduire un phénomène de désintégration pour les particules. La trajectoire s'arrête alors lorsque la particule se désintègre, et sa longueur (égale au temps de vol de la particule, car les trajectoires sont parcourues à vitesse constante unitaire) est donc une variable aléatoire qui suit une loi de probabilité $p_{\mathbf{l}}(l)$ décroissante. Lorsqu'on utilise un taux de désintégration constant, on obtient une loi exponentielle sur la longueur des courbes $p_{\mathbf{l}}(l) = \exp(-l/l_0)$[32].

Sur une trajectoire de longueur l, l'éloignement u des points au point de départ, lui-aussi égal au temps de vol, ne peut donc dépasser l. Autrement-dit, on ne pourra avoir des points d'éloignements u que pour les courbes de longueur supérieure à u. On a donc :

$$p_{\mathbf{u}}(u) = \int_{l>u} p_{\mathbf{l}}(l) dl$$

dans le cas exponentiel, on obtient :

$$p_{\mathbf{u}}(u) = \exp(-u/l_0)$$

Concrètement, si cette loi est plus satisfaisante d'un point de vue théorique pour les modèles stochastiques de la section 8.5.2 (elle est, comme eux, sans mémoire), elle peut en pratique être utilisée dans les applications comme la loi gaussienne, pour défavoriser les points les plus éloignés (on règle alors l_0 de la même façon que σ_u dans la section précédente). La principale différence est une décroissance moins rapide.

8.4.4 Échantillonnage régulier

Un dernier exemple est celui d'une loi liée à un échantillonnage régulier de la surface. Sur la segmentation du scanner, le "Marching Cube" produit des points placés sur les arrêtes d'un maillage cubique. Sur une image 2D 1/2 comme le Picza, les points sont situés sur les verticales des sommets d'un maillage carré (voir figure 8.2).

32. On parle souvent de la demi-vie $t_{1/2}$ des particules. Celle-ci correspond au temps pour lequel le nombre de particules est divisée par deux :

$$\exp(-t/t_0) = 1/2 \quad \Leftrightarrow \quad t_{1/2} = t_0.log(2)$$

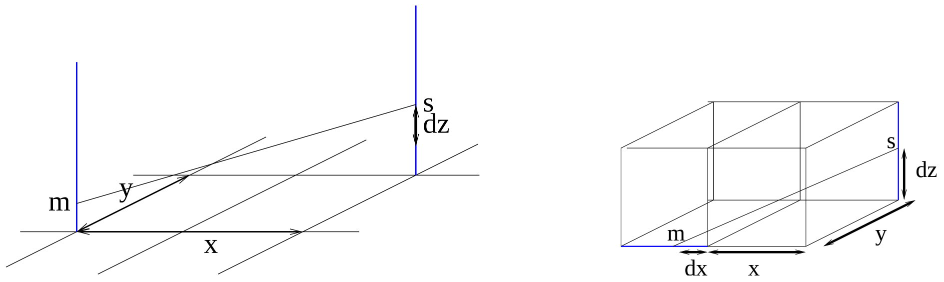


FIG. 8.2 – *Répartition des points sur une image 2D1/2 (à gauche) et sur une segmentation par Marching Cube (à droite).*

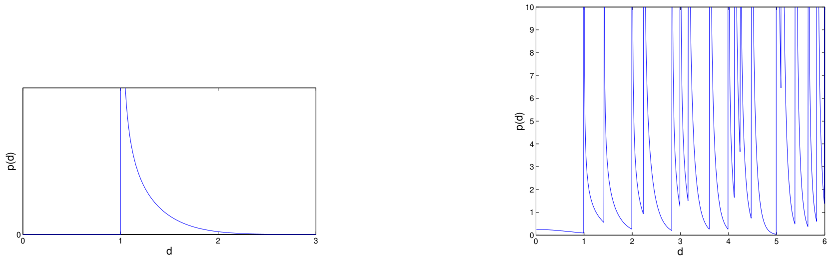


FIG. 8.3 – *Probabilité théorique de la distance entre deux points d'une image 2D1/2, avec z gaussien de variance 1. A gauche : pour $x = 1$ et $y = 0$. A droite : pour (x,y) entiers*

Cas d'une image 2D1/2 Nous étudions ici les cas du Picza (à gauche sur la figure).

Les coordonnées x et y sont forcément des multiples du pas de la grille, que nous supposerons ici unitaire. Par contre la dénivelée z prend des valeurs continues. Pour chaque couple (x,y), la distance suit alors une loi de probabilité que l'on peut facilement relier à la loi de probabilité sur z : on a $p_{\mathbf{d}}(d) = p_{\mathbf{z}}(z)/\left|\frac{\partial d}{\partial z}\right|$ et donc

$$p_{\mathbf{d}}(\sqrt{x^2+y^2+z^2}) = \frac{\sqrt{x^2+y^2+z^2}}{z}.p_{\mathbf{z}}(z) \qquad p_{\mathbf{d}}(d) = \frac{d}{\sqrt{d^2-x^2+y^2}}.p_{\mathbf{z}}(\sqrt{d^2-x^2+y^2})$$

La figure 8.3 à gauche montre cette probabilité avec un modèle gaussien : on a une probabilité nulle pour $d < \sqrt{x^2+y^2}$, un brusque pic pour $d = \sqrt{x^2+y^2}$, puis une probabilité décroissante.

Si nous prenons maintenant en compte tous les couples (x,y) possibles, on voit apparaître des pics caractéristiques correspondant à la répartition des $\sqrt{x^2+y^2}$ possibles avec x et y entiers (figure 8.3 à droite).

Nous n'aurons pas besoin de pousser notre étude plus loin, mais il existe certainement déjà des travaux sur ce sujet, notamment dans le domaine de la géométrie algorithmique ou concernant le spectre des atomes.

Cas du Marching-Cube Avec le Marching-Cube, la situation est à peine plus complexe. Il y a toujours deux coordonnées discrètes par points, mais ce ne sont plus forcément les mêmes (comme dans l'exemple de la figure 8.2 à droite). On montre facilement que l'on retrouve une structure avec des pics (certes moins marqués), dont la répartition suit l'ensemble des $\sqrt{x^2+y^2+z^2}$ possibles avec x, y et z entiers.

Remarquons que ces exemples vont à l'encontre de notre démarche générale présentée à la section 8.3.5. Le bruit de mesure est en-effet corrélé avec l'échantillonnage (il n'intervient que dans certaines directions). Il faudrait donc revoir dans ce cas précis notre façon de prendre en compte le bruit de mesure.

8.5 Modèle de forme

Nous allons maintenant revenir sur le modèle de forme, i.e. le terme $p((s,n_s)|(m,n_m),u)$ qui indique la loi sur une primitive en fonction d'une autre primitive et connaissant l'éloignement qui les sépare. L'approximation plane est certainement l'aspect le plus discutable de notre premier exemple (section 8.3), car elle entraîne deux conséquences gênantes :

- L'orientation reste inchangée lorsqu'on s'éloigne du point connu.
- Les points ne peuvent se disperser qu'à vitesse constante (et encore, c'est une conséquence du bruit de mesure sur l'orientation) autour du plan tangent lorsqu'on s'éloigne du point connu.

Ce comportement n'est certainement pas représentatif des courbes et surfaces, pour lesquelles on devrait avoir une variation des normales et une accélération de la dispersion des points autour du plan tangent lorsqu'on s'éloigne du point connu. L'intensité de ce phénomène devrait dépendre de la courbure.

Nous commencerons par présenter le modèle sphérique, où l'on suppose que les courbes et surfaces suivent des cercles et des sphères. Puis nous verrons un modèle plus général, où l'orientation de la tangente ou la normale peut varier aléatoirement à tout moment.

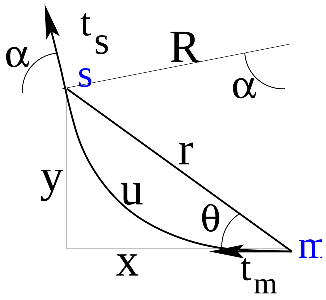
8.5.1 Modèles sphériques - "Diabolo Field"

Le modèle de probabilité utilisé par Médioni (le "Diabolo Field") s'appuie sur une modélisation des courbes et surfaces par des hypersphères (cercle en 2D, sphère en 3D). Il considère en-effet qu'il s'agit des surfaces les plus probables, et ne veut considérer que celles-ci. Nous allons donc reprendre cette idée et dériver ainsi le modèle sphérique. Nous verrons alors qu'il n'apporte qu'une seule correction majeure au "Diabolo Field" de Médioni.

8.5.1.1 Cas 2D : Cercle passant par un point orienté connu avec bruit sur la courbure

Supposons donc que nous sommes en présence d'un cercle passant par m, de tangente t_m connue, et dont la courbure suit une loi de densité $p_{\mathbf{C}}$.

Éloignement connu Comme dans le cas de la droite, on va d'abord s'intéresser à la loi pour un point à un éloignement u_0 donné (l'éloignement correspond maintenant à l'abscisse curviligne, i.e. la longueur de l'arc de cercle parcouru). La figure suivante présente les différentes notations utilisées, et les rapports entre les angles :



On peut donner la probabilité de s, en le paramétrant directement par l'éloignement u (connu), la courbure C (dont on connaît la loi $p_{\mathbf{C}}$) et l'angle α de la tangente en s :

$$p(u,C,\alpha|u_0) = \delta_{u_0}(u).p_{\mathbf{C}}(C).\delta_{2\theta}(\alpha)$$

En passant en coordonnées cartésiennes, on obtient l'approximation suivante (le détail des calculs est donné dans l'annexe F.2, on a en particulier utilisé $u = \frac{\theta.r}{\sin\theta} \simeq r$ et $C = 2\theta/r = 2\frac{\arcsin(y/r)}{r} \simeq 2y/r^2$, approximations valables lorsque θ reste faible) :

$$p(x,y,\alpha|u) = \delta_u\left(r\right).\frac{p_{\mathbf{C}}\left(2\theta/r\right)}{r^2}.\delta_{2\theta}(\alpha) \simeq \delta_u\left(r\right).\frac{p_{\mathbf{C}}\left(2y/r^2\right)}{r^2}.\delta_{2\theta}(\alpha)$$

Éloignement quelconque On utilise le théorème des probabilités totales, pour obtenir :

$$p(x,y,\alpha) = p_{\mathbf{u}}(r).\frac{p_{\mathbf{C}}(2\theta/r)}{r^2}.\delta_{2\theta}(\alpha) \simeq p_{\mathbf{u}}(r).\frac{p_{\mathbf{C}}(2y/r^2)}{r^2}.\delta_{2\theta}(\alpha) \tag{8.5}$$

8.5.1.2 Sphère en 3D

On peut directement généraliser le résultat précédent : les cercles deviennent des sphères de courbure C tangentes au plan défini par m et n_m (la sphère de courbure nulle est le plan lui-même). Dans un plan contenant (m,n_m), le problème se ramène aux cercles en 2D. On ré-utilise donc directement les résultats de la section précédente, en remplaçant y par $y' = \sqrt{y^2+z^2}$.

La probabilité du point est donnée par :

$$p(x,y,z) = p_{\mathbf{u}}(r).\frac{p_{\mathbf{C}}(2.\sqrt{y^2+z^2}/r^2)}{r^2} \tag{8.6}$$

La normale est située dans le plan contenant n_m (i.e. l'axe y), m et s (et donc l'axe $a = (\cos\varphi)x + (\sin\varphi)z$, où $\varphi = atan2(z,x)$), et forme un angle $\alpha = 2\theta$ avec l'axe a. La normale est donc donnée par :

$$n_s = (\cos\varphi.\cos 2\theta, \sin 2\theta, \sin\varphi.\cos 2\theta) \text{ avec } \varphi = atan2(z,x) \text{ et } \theta = atan2(\sqrt{x^2+z^2},y)$$

8.5.1.3 Diabolo Field

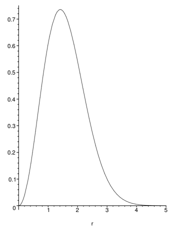
Le Diabolo Field est le “champ de vote de l'origine” utilisé par Médioni. Il utilise la même normale (déterministe) que nous, et la probabilité de son point (la trace de son tenseur) est donnée par la formule :

$$\exp(-(\alpha r^2 + cC^2)) = \exp(-\alpha r^2).\exp(-cC^2)$$

qui correspond à $p_{\mathbf{u}}(r).p_{\mathbf{C}}(C)$ avec un modèle gaussien pour $p_{\mathbf{u}}$ et $p_{\mathbf{C}}$. Or nous venons de trouver la formule $p_{\mathbf{u}}(r).\frac{p_{\mathbf{C}}(C)}{r^2}$ pour le modèle sphérique. Le Diabolo Field correspond donc au modèle sphérique à ceci prés qu'il ne tient pas compte de la “normalisation” en $1/r^2$. Précisons tout de même que Médioni impose une probabilité nulle pour un angle θ supérieur à 45 degrés.

Remarque

On pourrait présenter les choses autrement en disant que le modèle de Médioni correspond à une loi sur l'échantillonnage égale à $p_{\mathbf{u}}(r) = r^2.\exp(-\alpha r^2)$. Voici son tracé pour $\alpha = 1/2$ (i.e. $\sigma_u = 1$) :



Avec une telle loi, on défavorise toujours les points éloignés, mais on accorde une importance maximale aux points situés à une distance $d = \sqrt{2.\sigma_u^2}$, et on défavorise les points plus proches. Cette dernière propriété nous paraît infondée et dangereuse.

8.5.1.4 Conclusion

On retrouve là la même formule que pour le modèle affine avec bruit sur l'orientation, à ceci près qu'il apparaît une constante $A = 2$ comme exposant pour la distance r et pour l'égalité entre les angles des normales ($\alpha = 2\theta$). Celle-ci a pour effet d'accélérer la dispersion des normales et des points autour du plan tangent lorsqu'on s'éloigne de du point connu.

Mais ce modèle reste encore trop déterministe pour représenter une courbe ou une surface. En-effet, le seul phénomène aléatoire est le point de départ et le paramètre (orientation ou courbure) de la courbe ou de la surface, et, une fois ceux-ci connus, la forme de la courbe lorsqu'on s'éloigne du point de départ est parfaitement déterminée. L'orientation en un point reste toujours déterminée. En outre, comme la courbure ne varie pas, l'accélération de la dispersion est très importante.

Il faudrait donc trouver maintenant un modèle intermédiaire, où le comportement de la courbe puisse varier, passant aléatoirement entre des zones plates et des zones plus ou moins courbées.

8.5.2 Modèles stochastiques

Dans une véritable courbe, l'orientation et la courbure varient. C'est précisément ce que les modèles stochastiques de courbes permettent de modéliser (et en particulier le "Stochastic Completion Field" qui est le premier à avoir été introduit et qui reste aujourd'hui le plus utilisé et le mieux étudié). Nous allons donc les présenter ici. Le lecteur pourra se reporter à [August, 2001, Chap. 2] pour une présentation plus mathématique.

8.5.2.1 Introduction

Le "Stochastic Completion Field" a été introduit par [Mumford, 1994] pour la complétion des courbes occultées (nous évoquerons cette application dans la section 8.8.2). Il n'a été baptisé de ce nom qu'un peu plus tard, lorsque Thornber et Williams ont étudié son calcul par plusieurs méthodes [Williams and Jacobs, 1997; Thornber and Williams, 1996]. Le SCF a alors remplacé la définition heuristique de la saillance utilisée par [Sha'ashua and Ullman, 1988] pour segmenter les contours dans les images 2D (section 8.8.1). Il a enfin permis d'obtenir de remarquables résultats pour le lissage d'images 2D de contours [August, 2002] (section 8.8.1).

Avant de rentrer dans les détails, on peut préciser que les travaux sur le SCF se veulent proches du fonctionnement réel du cerveau : il a été démontré expérimentalement que la première couche du cortex visuel (V1) se charge de repérer les orientations (le cerveau travaille ensuite comme nous sur un ensemble de points semi-orientés), et que la seconde couche (V2) travaille sur des paires de points semi-orientés pour compléter les occultations [Zweck and Williams, 2000]. Il semble même qu'elle soit effectivement basée sur une "implémentation par réseau de neurone" du SCF[33] [Field *et al.*, 1993; Williams and Jacobs, 1997].

8.5.2.2 Marches aléatoires orientées

Nous voulons donc que l'orientation puisse varier aléatoirement. Dans le contexte des courbes, cela fait penser aux trajectoires de particules qui suivent des marches aléatoires. Nous allons donc présenter ici le mouvement brownien, avant de nous focaliser sur les marches aléatoires orientées.

Le mouvement brownien est le premier exemple de marche aléatoire. Il est utilisé pour modéliser la trajectoire des molécules dans un gaz, et donne une justification physique à la température (qui est une mesure de l'agitation, i.e. la vitesse moyenne des particules), et aux phénomènes de diffusions (de la chaleur ou de molécules).

Nous reprenons la définition de Wiener [Wiener, 1976] : si la particule est à une position s_u à un instant u, alors son mouvement brownien la mènera à l'instant $u' = u + du$ à un point aléatoire dont la loi suit une gaussienne centrée en u et de variance proportionnelle à du, *sans mémoire*, i.e. indépendante de tout ce qui a pu se passer avant[34]. Des exemples de mouvement browniens sont donnés fig. 8.4 à gauche.

On constate que si la trajectoire est continue, la direction suivie par le point peut elle changer à tout moment suivant une loi uniforme (puisque la loi sur s_{u+du} est isotrope autour de s_u). La tangente n'est donc pas vraiment définie (en tout cas pas continue).

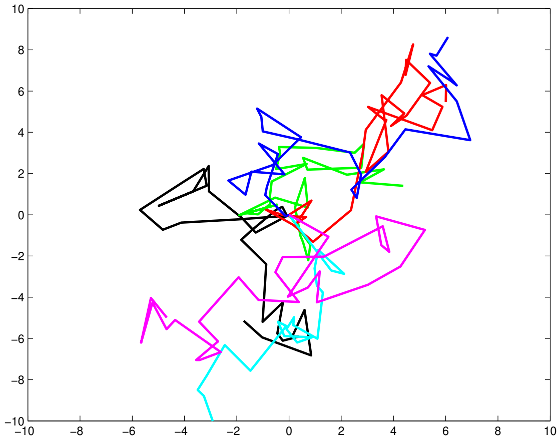
Les trajectoires qui vont nous servir à modéliser les courbes sont assez proches du mouvement brownien, à ceci près que c'est la direction t_{s_u} qui vérifie maintenant la propriété du mouvement brownien (et est donc continue) : $t_{s_{u+du}}$ suit une loi gaussienne centrée autour de t_{s_u}, *sans mémoire*, et de variance proportionnelle à l'intervalle de temps du :

$$t_{s_{u+du}} \sim G(t_{s_u}, \sigma_b^2 du)$$

Ces trajectoires sont parcourues à vitesse constante (et égale à 1 pour simplifier). On peut donc retrouver la trajectoire de la particule avec l'équation différentielle suivante

33. Les spécialistes ont pour cela présenté à différents volontaires plusieurs dessins analogues à notre exemple, et leur ont demandé de compléter les courbes. Pour chaque dessin, l'ensemble des réponses a fourni un étalon, auquel on peut comparer les résultats des algorithmes.

34. Pour être consistante, cette définition ne doit pas dépendre du pas de temps choisi. Ici, le point $s_{u''}$ à l'instant $u'' > u' > u$ suit une gaussienne de variance $\sigma_b^2(u'' - u')$ autour de $s_{u'}$ qui suit une gaussienne de variance $\sigma_b^2(u' - u)$ autour du point s_u. Comme le mouvement est sans mémoire, ces deux variables aléatoires gaussiennes sont indépendantes, et leur somme donne donc une troisième variable aléatoire de loi gaussienne de variance égale à la somme des deux premières : $s_{u''}$ suit une gaussienne de variance $\sigma_b^2(u'' - u') + \sigma_b^2(u' - u) = \sigma_b^2(u'' - u)$, ce qui est compatible avec notre définition. Cela n'aurait pas marché avec une loi uniforme (la somme de deux lois uniformes n'est pas uniforme) ni d'ailleurs avec toute autre loi.



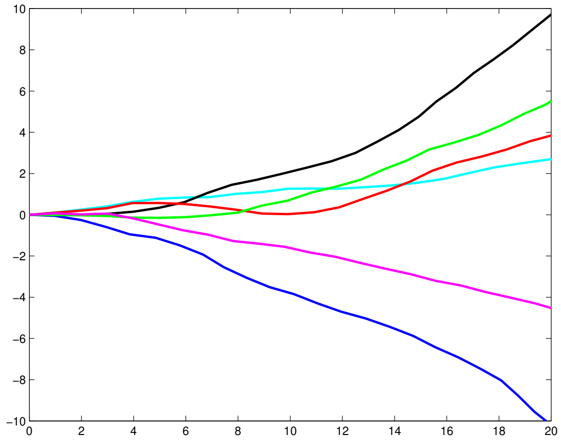


FIG. 8.4 – *Exemples de mouvements browniens (à gauche) et de marches aléatoires orientées (à droite)*

(qui est la définition de la tangente $t_s = (\cos\theta, \sin\theta)$) :

$$\frac{ds_u}{du} = vitesse.t_{s_u} = t_{s_u} \quad \Longleftrightarrow \quad \begin{cases} \frac{dx}{du} = \cos\theta_u \\ \frac{dy}{du} = \sin\theta_u \end{cases}$$

qu'on intègre à partir du point s_u :

$$s_{u'} = s_u + \int_u^{u'} t_{s_u}.du$$

On le voit fig. 8.4, les trajectoires ne sont plus complètement désordonnées, car les variations de l'orientation sont limités par la variance σ^2 de la gaussienne. Lorsque cette variance tend vers 0 les déviations disparaissent, et on retrouve une ligne droite déterministe. Inversement lorsque cette variance tend vers $+\infty$, la normale peut varier suivant une loi uniforme à chaque instant, et on retrouve un mouvement complètement désordonné. Les marches aléatoires orientées permettent donc de représenter des processus dont le comportement est intermédiaire entre le modèle affine (voir section 8.3) et le mouvement brownien.

Remarque : Lien avec la courbure

Le paramètre σ_b^2 du mouvement brownien sur l'orientation est homogène à des rad^2/mm (en-fait des rad^2/s, mais la vitesse de parcours est constante et la distance est donc proportionnelle au temps). Autrement dit la variation sur l'orientation est proportionnelle, en écart-type, à $\sqrt{d}$, la racine carré de la distance qui a été parcourue. Difficile dans ces conditions de relier le paramètre σ_b^2 à une courbure, i.e. une variation d'orientation par élément de distance (homogène à des rad/mm). Deux pistes se présentent :

– On peut transposer le raisonnement qui permet de calculer l'énergie cinétique (i.e. la vitesse qua-

dratique) moyenne des particules qui suivent un mouvement brownien. Ici, on obtiendrait ainsi la courbure (i.e. la vitesse de variation de l'orientation) quadratique moyenne des trajectoires.
- On peut redéfinir notre marche aléatoire orientée en remplaçant la vitesse fixe par une vitesse qui correspond à un mouvement brownien. Ainsi, la variation sur l'orientation et la distance parcourue seraient toutes deux en moyenne proportionnelles à $\sqrt{d}$, et leur rapport (i.e. la courbure) serait ainsi bien défini.

8.5.2.3 Passage au modèle de forme

Un modèle de trajectoire stochastique, que ce soit notre marche aléatoire orientée sans mémoire ou tout autre modèle, permet d'écrire la loi sur la position de (s,t_s) en fonction du point de départ (m,t_m) et du temps de parcours u. On obtient donc directement la loi sur la forme $p((s,t_s)|(m,t_m),u)$. Mais maintenant, c'est le temps de parcours u qui sert de définition pour l'éloignement. Comme la trajectoire est parcourue à vitesse unitaire, cet éloignement est donc égal à la longueur de la courbe. Cet éloignement peut donc maintenant être égal (pour les trajectoires rectilignes), mais aussi supérieur (pour les courbes légèrement courbées) voire très supérieur (pour les courbes très courbées) à la distance entre les deux points.

Remarquons que les auteurs associent souvent ce modèle de forme stochastique à une décroissance exponentielle $p_{\mathbf{u}}(u) = e^{-u/\tau}$ pour le modèle sur l'éloignement (nous avons déjà expliqué pourquoi à la section 8.4.3).

8.5.2.4 Calcul

Nous avons défini le SCF mais n'en avons pas donné de formule explicite. Il s'agit en-effet d'un problème complexe, auquel ont été trouvé plusieurs réponses :
- La simulation : on lance plusieurs millions de trajectoires qui suivent le processus stochastique choisi, et on accumule leur trace dans l'espace des points semi-orientés discrétisé.
- La résolution de l'équation de Fokker-Plank[35].

35. Les méthodes présentées ici sont discutées en détail dans [August, 2001, Section 6.1.1]. Nous ne donnons ici que les idées essentielles.

Le produit des modèles de forme et d'échantillonnage $P(s,t_s,u) = p((s,t_s)|(m,t_m),u).p_{\mathbf{u}}(u)$ vérifie, lorsqu'ils sont sans mémoires, une équation différentielle stochastique de Fokker-Plank. Dans le cas du SCF (i.e. de la marche aléatoire orientée sans mémoire avec échantillonnage exponentiel), il s'agit de l'équation suivante :

$$\frac{\partial P}{\partial u} = -\cos\theta\frac{\partial P}{\partial x} - \sin\theta\frac{\partial P}{\partial y} + \frac{\sigma^2}{2}\frac{\partial^2 P}{\partial \theta^2} - \frac{1}{\tau}P$$

On retrouve la loi de probabilité d'une primitive connaissant une autre primitive pour un éloignement u quelconque en intégrant P par rapport à u :

$$p((s,t_s)|(m,t_m)) = \int P(u,s,t_s).du = \int p((s,t_s)|(m,t_m),u).p_{\mathbf{u}}(u).du$$

- L'utilisation d'un réseau de neurone : comme nous l'avons déjà dit, il semble que la zone V2 du cortex visuel reproduise le calcul des modèles stochastiques. Il existe en tout cas une méthode [Williams and Jacobs, 1997] permettant d'effectuer son calcul à partir d'un réseau de neurone très simple (un perceptron avec une première couche de 7 neurones et un seul neurone de sortie).
- Le calcul explicite : celui-ci ne semble pas possible avec la définition standard du SCF. Seul [Thornber and Williams, 1996] a présenté le calcul explicite du produit de deux SCF avec une définition légèrement différente (ce produit est utilisé pour la complétion de courbes, voir 8.8.2).

Comme nous aurons probablement besoin de la différencier, nous avons choisi de trouver une formule explicite approchant le SCF sous sa définition standard. Pour cela, nous avons choisi l'approche expérimentale : nous avons simulé de très nombreuses trajectoires (plusieurs millions), et avons cherché à trouver une forme analytique aussi proche que possible des résultats expérimentaux.

8.5.2.5 Simulation du SCF 2D

Nous avons trouvé dans l'annexe F.3 une forme analytique du SCF qui se révèle être proche de celui que nous avons obtenu par simulation, et que nous utiliserons par la suite. La probabilité du point toutes orientations confondues est approximativement :

$$p(s=(x,y)) \approx p_u(r) \, . \frac{\exp\left(-\frac{y^2}{2*k*\sigma_b^2*r^{1.5}}\right)}{r^{1.5}} \quad \text{avec}\, k = 0.34 = 1/2^{1.5} \tag{8.7}$$

En ce qui concerne la tangente, la plus probable correspond à $\alpha = 1.5 * \theta$ avec pour variance $\sigma_\alpha^2 = \frac{\sigma_b^2 . r}{2}$ (rappelons qu'elle était nulle pour les modèles affines et sphériques).

8.5.2.6 Conclusion

Le Stochastic Completion Field permet donc de modéliser les courbes et d'en déduire un modèle probabiliste sur l'emplacement de leurs point. Deux aspects caractérisent cette description :

- L'aspect sans mémoire : les trajectoires sont sans mémoire. On ne peut donc modéliser des trajectoires présentant des structures répétitives ou symétriques. On ne peut non plus prendre en compte les problèmes d'auto-intersection.

Pour faire effectivement le calcul, on peut donc résoudre numériquement l'équation différentielle en travaillant sur l'espace discrétisé, en partant d'un point parfaitement connu, i.e. de la condition initiale suivante : $P(u=0) = \delta_{(m,t_m)}$. On intègre ensuite P.

Cette approche pose malheureusement des problèmes numériques : appliquée directement, elle donne des résultats qui ne sont pas invariants par transformation rigide. Il faut donc travailler sur une base de fonctions invariantes par transformation rigide (ou de façon équivalente dans l'espace de Fourier), pour obtenir des résultats corrects, au prix de calculs très complexes.

– La loi sur les variations d'orientation : elle conditionne la répartition des courbures.

Ce modèle ne peut pas être calculé explicitement (bien que l'on puisse le faire en utilisant une variante de sa définition), mais nous en avons trouvé une formule analytique approchée par la simulation qui permettra de faire les calculs par la suite.

8.5.3 Modèle général sans bruit

On peut constater une certaine analogie en regardant la forme (figure 8.5) et l'expression des 3 modèles que nous avons présenté en 2D pour les courbes (équations 8.3, 8.5 et 8.7) et des modèles plans et sphériques en 3D (eq. 8.4 et 8.6). En effet, la probabilité du point toutes orientations confondues peut se mettre sous la forme :

$$
\begin{array}{lll}
p(x,y) \simeq p_u(r) . \dfrac{\exp\left(-\frac{\theta^2}{2.\sigma_{forme}^2.r^{2(A-1)}}\right)}{r^A} & \simeq p_u(r) . \dfrac{\exp\left(-\frac{y^2}{2.\sigma_{forme}^2.r^{2A}}\right)}{r^A} & \\
A = 1 \quad \sigma_{forme}^2 = \sigma_n^2 & \text{pour le modèle affine} & \\
A = 2 \quad \sigma_{forme}^2 = \sigma_C^2/2^2 & \text{pour le modèle sphérique} & \\
A = 1.5 \quad \sigma_{forme}^2 = \sigma_b^2/2^{1.5} & \text{pour le modèle stochastique} &
\end{array}
\tag{8.8}
$$

On suppose donc là que l'éloignement u est égal à la distance r (ce qui n'est vrai que pour le modèle affine, seulement approximatif pour les deux autres), et nous prenons le produit de la loi sur l'éloignement $p_u(r)$ et de la loi sur la forme en fonction de l'éloignement $u = r$, qui est de la forme $p(\theta/r^{A-1}) \simeq p(y/r^A)$ avec ici p gaussien, et pour laquelle on doit prendre en compte la constante de normalisation en $1/r^{2A}$.

Les lois pour l'orientation connaissant la position sont toutes centrées sur la normale située dans le plan contenant n_m, m et s, et forme un angle $\alpha = A\theta$ avec l'espace tangent où $\theta \simeq y/r$ est l'angle entre ms et l'espace tangent. Elles sont déterministes pour les modèles affines et sphériques, et isotrope de variance totale $\sigma_\alpha^2 = \frac{\sigma_b^2.r}{2}$ pour le modèle stochastique.

Nous allons maintenant expliquer en détail le comportement de ce modèle général en fonction des différents paramètres. Précisons ici que les approximations faites peuvent être mises en défaut pour de fortes valeurs de θ.

Dispersion des points autour de l'espace tangent La loi de probabilité tend, à proximité du point connu, à imposer une distance à l'espace tangent nulle ($y = 0$).

Ensuite, lorsqu'on s'éloigne du plan tangent, les points peuvent se disperser plus ou moins vite autour de l'espace tangent. Nous pouvons estimer cette dispersion en fonction de l'éloignement, en calculant la variance de y (la composante de ms perpendiculaire à l'espace tangent) en fonction de x (la composante de ms parallèle à l'espace tangent). Le

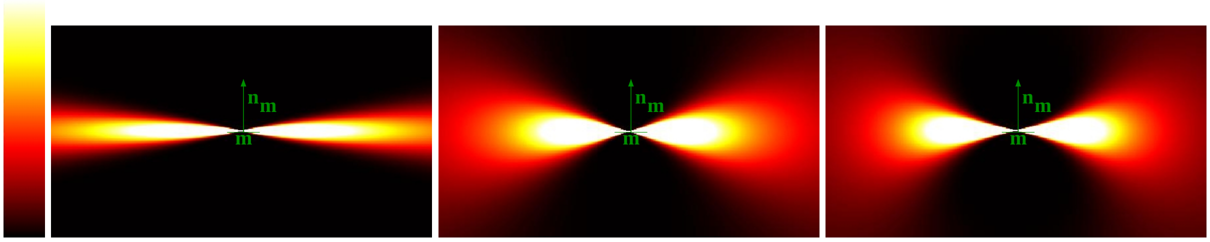


FIG. 8.5 – *Image de la probabilité du point $p(s)$ pour le modèles affine (à gauche), stochastique (au milieu) et sphérique (à droite). On a utilisé une loi uniforme sur l'échantillonnage $p_{\mathbf{u}} = Cte$, et un modèle gaussien.*
L'échelle de couleur est celle que nous utiliserons pour toutes les images de ce type : la couleur varie du noir pour une probabilité nulle, au blanc pour une probabilité égale (ou supérieure) à un seuil fixé arbitrairement pour chaque image (mais rappelons ici que tous les modèles sont calculés à une constante multiplicative près).

calcul approché fourni dans l'annexe F.5 donne :

$$\sigma^2_{y|x} = \sigma^2_{forme}.x^{2A}$$

La dispersion augmente donc en fonction de la vitesse σ^2_{forme}, et de l'ordre d'accélération A. Pour $A = 1$ (i.e. pour les modèles affines), la vitesse est constante, égale à σ^2_{forme}, et le modèle est donc invariant par changement d'échelle. Ensuite, plus A est élevé, plus la vitesse de dispersion s'accélère lorsqu'on s'éloigne du point connu. Le modèle n'est alors plus invariant par changement d'échelle.

Évolution des normales La loi de probabilité tend, à proximité du point connu, à imposer une orientation inchangée ($t_s = t_m$ ou $n_s = n_m$). L'orientation tourne ensuite dans le plan contenant n_m, m et s, A fois plus vite que le vecteur ms. Dans le cas affine ($A = 1$), elle tourne à la même vitesse, et reste donc alignée avec ms.

Dans le cas stochastique, ceci ne vaut que pour l'orientation moyenne. La variance autour de cette orientation moyenne est égale à $\sigma^2_{forme}.r$.

Complétude Les modèles que nous avons présenté ici ne sont pas complets, au sens où on ne peut atteindre, à partir d'une primitive, qu'un ensemble négligeable d'autres primitives (ceci est particulièrement vrai au voisinage du point connu, où on doit rester dans l'espace tangent). Ce problème va être immédiatement résolu, par l'ajout d'un bruit de mesure.

8.5.4 Discussion

Pour finir, nous allons donner ici quelques perspectives.

8.5.4.1 Modèles stochastiques pour les surfaces

Nous avons présenté les modèles stochastiques (défini à partir d'une marche aléatoire sur l'orientation) pour les courbes en 2D et 3D. Il nous manque leur équivalent pour les surfaces. Il semble qu'un tel modèle n'ait jamais été mentionné dans la littérature, et nous n'avons malheureusement pas eu le temps d'en dériver un de façon rigoureuse. Voici cependant quelques pistes pour y arriver.

La première s'appuie sur l'analogie qui existe, pour les courbes, entre les différents modèles. Comme nous venons de le mentionner, ceux-ci ne diffèrent que par le coefficient A, égal à 1 pour les modèles affines, 2 pour les modèles sphériques, et 1.5 (i.e. intermédiaire) pour les modèles stochastiques. Cette analogie reste toujours valable pour les versions surfaciques des modèles affines et sphériques. Il semble alors naturel de définir un SCF surfacique en remplaçant par 1.5 les termes égaux à 1 pour les modèles plans et 2 pour le modèles sphérique. C'est ce que nous avons effectivement fait dans notre implémentation.

La deuxième piste s'appuie sur l'analogie entre la version courbe 2D et la version surfacique 3D des modèles affines et sphériques. Dans les deux cas, le modèle surfacique 3D prend, dans un plan contenant la normale, les mêmes valeurs que le modèle 2D. Ceci n'est pas surprenant dans la mesure où l'image d'un plan (resp. d'une sphère) dans un tel plan et une droite (resp. un cercle) qui varie de la même façon.

Il faut cependant noter que ces deux analogies ne sont pas complètes : si elles sont valables pour la probabilité du point toutes normales confondues et la normale la plus probable en un point, elles ne sont en revanche pas valables pour la covariance de la normale en un point, et on ne peut donc pas caractériser entièrement le SCF pour les surfaces de cette façon là.

Il serait donc préférable de dériver le SCF pour les surfaces à partir d'un véritable modèle. Pour définir définir une surface stochastique, plusieurs approches semblent se dessiner :

- Une approche reposant sur les courbes : notre seconde analogie n'est qu'une illustration du lien qu'il existe entre les surfaces et les courbes que l'on peut tracer sur ces surfaces. Or nous n'avons pas du tout étudié ce lien sur les modèles affines 3D (où l'on vérifie pourtant facilement que le normale la moins probable pour les surfaces est la tangente la plus probable pour les courbes). Cette étude, déjà intéressante en soit, permettrait probablement de renforcer et compléter notre seconde analogie, et dériver ainsi le SCF pour les surfaces, sans pour autant lui donner une véritable définition.
- Une approche directe : les marches orientées sans mémoire que nous avons utilisé pour modéliser les courbes peuvent se voir, si on discrétise le temps u, comme une chaîne de Markov (c'est d'ailleurs la modélisation qu'utilise [August, 2001, Chap. 2]), i.e. un processus aléatoire $s(u)$ fonction de l'éloignement u tel que la valeur de

$s(u+1)$ ne dépend que de la valeur de $s(u)$. Pour les surfaces, il faudrait utiliser un champs de Markov bidimensionnel i.e. un modèle aléatoire de la forme $s(u_1,u_2)$ avec u_1 et u_2 discrets (il s'agit donc d'un maillage de la surface à maille carré) tel que $s(u_1,u_2)$ ne dépend que de ce qui se passe au voisinage immédiat de s (les points $s(u_1 \pm 1,u_2)$ et $s(u_1,u_2 \pm 1)$).

8.5.4.2 Marche aléatoire sur la courbure

Avec le mouvement brownien, il pouvait y avoir des changements brusques de direction et la notion de tangente était ainsi mal définie. Avec les marches aléatoires orientées, c'est la direction qui suit un mouvement brownien. Ainsi, il n'y a plus de changement brusque de direction et la tangente est bien définie. Mais nous n'avons fait que décaler le problème : la vitesse de variation de la direction, i.e. la courbure, peut varier brusquement et est donc mal définie. Ceci n'est pas forcément très représentatif des courbes que l'on peut rencontrer dans la nature, et [August, 2001] a donc introduit le "Curve Indicator Random Field" en 2D, en décalant une nouvelle fois le problème : cette fois-ci, c'est la courbure κ_u qui suit un mouvement brownien. Il faut alors une position, une direction, mais aussi une courbure initiale. Pour retrouver la trajectoire, il faut intégrer deux fois. Dans un premier temps, on intègre la courbure pour connaître l'orientation de la courbe à tout instant, puis on intègre cette orientation pour connaître la position :

$$\theta_{u'} = \theta_u + \int_u^{u'} \kappa_u.du$$

$$s_{u'} = s_u + \int_u^{u'} \begin{pmatrix} \cos\theta_u \\ \sin\theta_u \end{pmatrix} .du$$

Ce nouveau modèle pose deux problèmes :

- Comment fixer la courbure initiale : imposer une courbure nulle n'est pas pertinent, et on utilisera donc une courbure initiale qui suit une loi de probabilité donnée (en général gaussienne). Remarquons que si on travaillait avec des primitives d'ordre 2 (i.e. pour lesquelles les informations sur la courbure sont connues), il n'y aurait alors plus de problème pour fixer la courbure initiale (i.e. celle du point de départ qui est connu). Le CIRF serait alors probablement un modèle particulièrement bien adapté.
- Comment le généraliser en 3D : ce n'est pas évident, car la courbure seule ne permet pas de définir complètement le comportement de la courbe : il faut de plus savoir quel est le plan osculateur. On connaît déjà la tangente, il nous faut donc un angle supplémentaire pour caractériser ce plan. Il faudra probablement faire évoluer cet angle "de torsion" toujours suivant un processus stochastique.

Pour ces raisons, nous n'utiliserons pas le CIRF dans cette thèse. Ce serait pourtant intéressant d'un point de vue théorique, car, à l'image du SCF qui généralise le modèle affine, il généralise le modèle sphérique (c'est sa limite pour des variations nulles sur la courbure).

8.6 Prise en compte d'un bruit de mesure et modélisation finale

Nous avons ici présenté les modèles élémentaires pour un point et une direction parfaitement connus. Mais en pratique, les primitives que nous connaîtrons de nos courbes ou de nos surfaces seront corrompues par du bruit. Nous avons déjà montré comment le prendre en compte avec l'approximation plane (section 8.3.3). Nous reprenons ici les calculs avec le modèle plus général que nous venons d'obtenir. Nous en profitons pour ajouter quelques remarques sur ces bruits de mesure.

Nous expliquons dans un premier temps comment prendre en compte le bruit sur l'orientation (section 8.6.1). Ce dernier cas permet en particulier de dériver les modèles "plateau" et "boule" de Médioni, qui correspondent au cas où la normale suit une distribution uniforme dans un plan ou dans l'espace (i.e. lorsqu'elle n'est pas connue). Enfin, on étudie le cas où les deux bruits sont présents (section 8.6.2).

8.6.1 Prise en compte d'un bruit sur l'orientation

8.6.1.1 Contexte

La prise en compte d'un bruit sur l'orientation peut en fait s'appliquer dans plusieurs contextes.

Le premier est identique à la prise en compte du bruit sur la position des points. Il s'agit tout simplement de prendre en compte le fait que notre orientation t_m ou n_m n'est pas une mesure parfaite de la véritable orientation au point m. On utilise alors en général un bruit gaussien centré sur l'orientation mesurée et d'écart-type correspondant à l'écart-type estimé du bruit sur l'orientation.

Le second correspond aux tenseurs plateau et boule tels que nous les avons présentés dans la section 7.3.3.5. Il s'agit respectivement du cas où l'on sait que l'orientation est située dans un plan donné (lorsqu'on connaît par exemple un élément de courbe situé sur une surface : on sait alors que la normale à la surface est située dans le plan perpendiculaire à l'élément de courbe) et du cas où l'on ne sait rien de l'orientation (lorsqu'on n'en dispose pas sur les points d'entrées). Le premier cas peut être représenté par un bruit sur la normale uniforme dans le plan, le second par un bruit sur la normale complètement uniforme.

8.6.1.2 Cas général

Supposons donc que l'orientation au point m suit une loi de probabilité $p_{\mathbf{n}_m}(n)$, centrée autour de n_m. Le théorème des probabilités totales permet d'écrire :

$$p(s,n_s) = \int p(s,n_s|m,n).p_{\mathbf{n}_m}(n).dn$$

Il n'existe pas d'approche analytique simple pour calculer cette probabilité (contrairement au cas de la position, où nous avions un produit de convolution). On peut bien-sur procéder par intégration numérique (c'est ce que fait Médioni lorsqu'il dérive le modèle plateau et boule à partir du modèle bâton), mais nous préférons ici obtenir une forme analytique, même approchée, pour notre modèle général.

Le calcul en 2D (mais nous utiliserons la même formule en 3D) et les approximations qu'il utilise sont présentés dans l'annexe F.4. On obtient, pour la loi sur la position du point toutes orientations confondues :

$$\begin{aligned} p(x,y) &= \frac{p_u(r)}{\sqrt{\left(\sigma_{forme}^2.r^{2A}+\sigma_{nm}^2.r^A\right)}}.\exp\left(-\frac{\theta^2}{2.\left(\sigma_{forme}^2.r^{2(A-1)}+\sigma_{nm}^2\right)}\right) \\ &\simeq \frac{p_u(r)}{\sqrt{\left(\sigma_{forme}^2.r^{2A}+\sigma_{nm}^2.r^A\right)}}.\exp\left(-\frac{y^2}{2.\left(\sigma_{forme}^2.r^{2A}+\sigma_{nm}^2.r^2\right)}\right) \end{aligned} \tag{8.9}$$

Concernant la loi sur la normale en un point donné, elle est maintenant centrée autour de la normale définie par :

$$\alpha = \frac{\frac{1}{\sigma_{forme}^2.r^{2(A-1)}} + A\frac{1}{\sigma_{nm}^2}}{\frac{1}{\sigma_{forme}^2.r^{2(A-1)}} + \frac{1}{\sigma_{nm}^2}}.\theta$$

et de variance :

$$\frac{(1-A)^2}{\frac{1}{\sigma_{forme}^2.r^{2(A-1)}} + \frac{1}{\sigma_{nm}^2}} + \sigma_b^2.r$$

8.6.1.3 Modèle boule

Nous pouvons maintenant examiner de plus près le cas particulier d'une loi uniforme sur la normale, i.e. le cas où nous avons un point d'entrée sans aucune information sur la normale. Ce cas correspond à une variance infinie $\sigma_{n_m}^2 = +\infty$. On voit alors que la normale suit une loi gaussienne centrée autour de la normale définie par :

$$\alpha = \theta$$

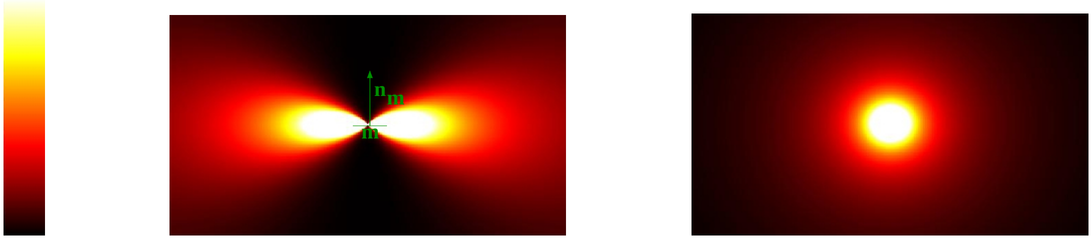


FIG. 8.6 – *Probabilité d'un point $p(s)$ pour le modèle stochastique avec un échantillonnage uniforme et un bruit sur l'orientation de la normale gaussien (à gauche), et uniforme (modèle boule à droite).*

i.e. identique à la normale n_m, ce qui s'imposait par de simples considérations de symétrie (voir 7.4.1). La variance de la gaussienne est donnée par :

$$(1-A)^2.\sigma_{forme}^2.r^{2(A-1)} + \sigma_2^2.r$$

et la probabilité du point est (voir figure 8.6 droite) :

$$p(x,y) = \frac{p_u(r)}{r} \tag{8.10}$$

Ceci peut permettre d'établir une fonction de vote pour l'interpolation des normales (voir 7.4.1).

8.6.2 Modèle final

Pour prendre en compte les deux bruits de mesure à la fois, nous allons faire l'hypothèse qu'ils sont indépendants. Le bien fondé de cette hypothèse dépendra de la méthode d'acquisition[36], mais elle est nécessaire pour effectuer un calcul simple. On peut en-effet alors prendre en compte le bruit sur l'orientation (en utilisant la formule de la section précédente) puis, indépendamment, le bruit sur la position (en convoluant, voir section 8.3.3). Un exemple est donné sur la figure 8.7 droite.

Pour comprendre ce nouveau modèle, nous reprenons l'analyse présentée pour les modèles sans bruit (section 8.5.3).

36. Pour les scanners segmentés par un "Marching Cube", on interpole la position du point et l'orientation de la normale (par le biais du gradient) en utilisant l'intensité des voxels adjacents. Il existe donc une corrélation complexe entre les deux bruits de mesure.

Pour les points per-opératoires, on utilise un capteur de position/orientation optique (pour lequel l'hypothèse est déjà fausse, pour des raisons qui dépassent le cadre de cette thèse), couplé à une pointe de mesure. On utilise l'orientation du capteur et la position relative de la pointe et de son axe pour calculer la position du point et l'orientation de la surface. Les deux bruits sont donc là aussi corrélés de manière complexe.

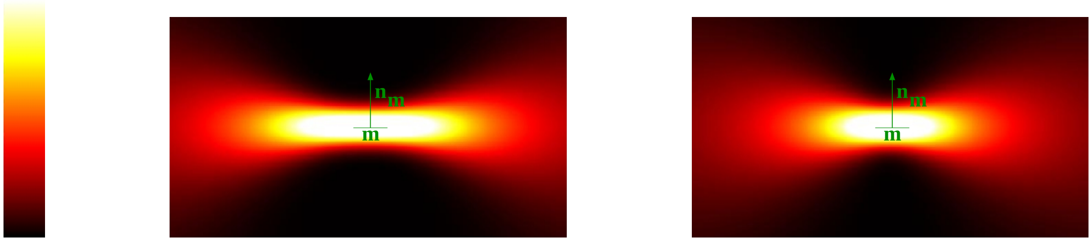


FIG. 8.7 – *Probabilité d'un point pour le modèle stochastique avec un échantillonnage uniforme et un bruit gaussien sur la position du point (à gauche), sur la position du point et l'orientation de la normale (à droite).*

Dispersion des points autour de l'espace tangent La variance de la dispersion autour de l'espace tangent, i.e la variance de y (la composante de ms perpendiculaire à l'espace tangent) en fonction de x (la composante de ms parallèle à l'espace tangent), devient :

$$\sigma_{y|x}^2 = \sigma_{forme}^2 . x^{2A} + \sigma_{n_m}^2 . x^2 + \sigma_m^2$$

On a là trois termes :

- Un terme σ_m^2 d'ordre 0 (i.e. constant) dû au bruit de mesure sur la position.
- Un terme $\sigma_{n_m}^2$ d'ordre 1 (i.e. linéaire) dû au bruit de mesure sur l'orientation.
- Un terme σ_{forme}^2 d'ordre A dû au modèle de forme. Remarquons que dans le cas plan, $A = 1$, et ce dernier terme est en-fait identique au précédent. Il faut alors considérer que c'est la somme $\sigma_{forme}^2 + \sigma_{n_m}^2$ qui donne la variance de la normale n_m.

Répartition des normales L'effet de la convolution par le bruit de mesure en position produit un effet complexe sur la répartition des normales en un point donné. Nous ne chercherons pas de formule explicite, simplement quelques résultats qualitatifs :

- Au voisinage du point connu, le modèle sans bruit tend vers un plan parfaitement connu. La répartition aléatoire est la conséquence du seul bruit de mesure sur l'orientation pour les points du plan. Autrement dit, au voisinage de m (i.e. lorsque la dispersion ne se fait pas encore sortir), la loi sur la normale est quasiment égale à p_{n_m} sur le plan qui est le seul endroit où la probabilité n'est pas nulle. Après lissage, les points autour du plan auront une probabilité non nulle, mais la loi sur la normale suivra toujours p_{n_m}.
- Lorsqu'on est loin du point connu, θ, r et donc la loi de probabilité varient peu sur des zones de la taille de l'échelle du bruit. La lissage gaussien n'a donc pratiquement aucun effet, et la loi que nous avons trouvé en prenant en compte le bruit sur l'orientation convient parfaitement.

Complétude La prise en compte des modèles de bruit a permis de rendre nos modèles plus complets, au sens où on peut maintenant atteindre un nombre non négligeable de primitives à partir d'une primitive donnée.

8.6.3 Conclusion

Nous avons maintenant un modèle théorique complet. Il peut être adapté au cas où certaines normales ne sont pas disponibles :

- Lorsque les normales de la scène n_s ne sont pas disponibles, on s'intéressera à la probabilité du point toutes normales confondues $p(s)$.
- Lorsque les normales du modèle n_m ne sont pas disponibles, on utilisera le modèle boule, i.e. le modèle correspondant à un modèle de bruit de mesure uniforme sur n_m.

Remarque pour l'implémentation

On peut calculer la loi de probabilité en deux temps. Dans un premier temps, on utilise notre formule approchée 8.9, qui prend en compte la loi p_u sur l'échantillonnage, la loi $p(s,n_s|m,n_m,u)$ sur la forme de notre surface, et le bruit de mesure p_{n_m} sur l'orientation. On calcule ainsi un premier champ de tenseur.
Dans un deuxième temps, nous appliquons un lissage gaussien à ce champ de tenseur pour prendre en compte le dernier élément de notre modèle, le bruit de mesure sur les points.

8.7 Évaluation expérimentale des modèles

Nous avons maintenant à disposition un modèle général pour les éléments de courbes et surfaces aléatoires, que nous allons utiliser dans diverses applications. La question va alors se poser de savoir quels paramètres choisir pour un type de courbe ou de surface donné. Nous allons donc ici étudier la loi empirique d'une primitive connaissant une autre primitive, i.e. mesurer directement la probabilité $p((s,n_s)|(m,n_m))$ sur les données dont nous disposons, et la comparer aux modèles théoriques que nous avons dérivés.

L'étude présentée ici porte sur les surfaces. Mais son adaptation aux courbes serait immédiate.

8.7.1 Mesure empirique de $p((s,n_s)|(m,n_m))$

Nous souhaitons donc mesurer la répartition empirique des primitives (s,n_s) par rapport aux primitives (m,n_m). Comme nous nous intéressons à leur position relative, nous allons dans un premier temps appliquer, pour chaque couple de point (s,n_s) et (m,n_m), la transformation rigide (voir section 8.2.2) qui permet de ramener (m,n_m) sur l'origine

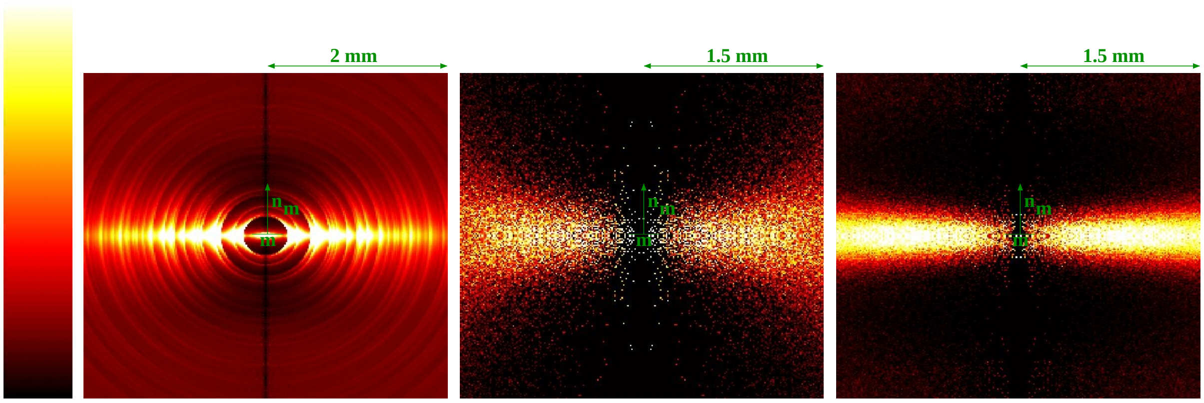


FIG. 8.8 – *Probabilité d'un point $p(s)$ pour différentes lois empiriques :*
A gauche : pour les surfaces segmentées dans les scanners
Au milieu : pour les pointés per-opératoires pris en mode contact sur les dents et recalés sur la surface segmentée du scanner
A droite : idem pour les pointés per-opératoires pris en mode rafale sur l'os
NB: Beaucoup de points de la bande centrale (i.e. pour x faible, mais rappelons que l'on divise la valeur trouvée par x) ont été supprimés pour éviter les instabilités numériques dans les analyses.

et s dans le plan xy. On note alors le nombre de primitives obtenues pour chaque valeur de $(s',n_{s'})$ possible. On obtient là une estimation empirique de la loi $p((s,n_s)|(m,n_m))$.

On peut ainsi mesurer une loi empirique pour :

- Une surface échantillonnée donnée : (s,n_s) et (m,n_m) décrivent alors l'ensemble des points de la surface.
- Un nuage de point $\mathcal{S}$ recalé sur une surface échantillonnée $\mathcal{M}$: (s,n_s) décrit alors l'ensemble des points du nuage $\mathcal{S}$, et (m,n_m) décrit alors l'ensemble des points de la surface $\mathcal{M}$.

Lorsqu'on dispose de plusieurs données du même type, on pourra additionner les lois empiriques obtenues pour chaque donnée. C'est le cas de nos trois exemples, figure 8.8, qui présentent, pour l'ensemble de la base de test (voir section 6.1.5), les lois empiriques de la surface segmentée du scanner (à gauche) et des nuages de points per-opératoires recalés sur les surfaces scanners (au milieu : les nuages de points en mode contact sur les dents, à droite, les nuages de points en mode rafale sur l'os).

Remarque pour l'implémentation

Il faut en fait utiliser les résultats de la section 8.2.3, i.e. utiliser un champ de tenseur pour stocker efficacement la loi $p((s,n_s)|(m,n_m))$. On discrétise donc la partie du plan xy qui nous intéresse, et pour chaque primitive $(s',n_{s'})$ obtenue dans cette partie du plan, on additionne le tenseur $n_{s'}.n_{s'}^t$ au tenseur associé au pixel contenant le point s'. Il faut alors penser que le pixel (x,y) représente en fait un solide de révolution autour de l'axe y (car on a tourné autour de l'axe y pour se ramener dans le plan xy), dont le volume est proportionnel à x. Il faut donc diviser la valeur de chaque pixel par son abscisse x.

8.7.2 Étude générale

La forme des lois empiriques confirme notre décomposition en trois composantes aléatoires indépendantes :

- Le bruit de mesure : celui-ci induit un lissage gaussien, qui est surtout visible dans l'axe de la normale. Précisons que dans le cas du scanner (fig. 8.8 à gauche), il est corrélé, pour des raisons que nous expliquerons, au bruit d'échantillonnage, ce qui explique que la dépendance en r reste forte, et que le lissage ne s'observe qu'en θ.
- La loi sur l'échantillonnage : on distingue sur la loi empirique de la segmentation d'un scanner avec elle même une dépendance très irrégulière en fonction du rayon r i.e. de la distance à l'origine (on constate d'ailleurs que les franges ainsi induites ne sont en-fait pas circulaires mais légèrement elliptiques, comme dans le SCF, cf section F.3). Cette dépendance provient de la structure régulière de l'échantillonnage. En revanche, pour un recalage, la distance à l'origine suit une loi uniforme, car les échantillonnages des deux surfaces sont vraiment indépendants.
- La loi sur la forme de la surface : on retrouve un effet de dispersion autour du plan tangent lorsqu'on s'éloigne du point connu. Il présente a première vue une forme analogue à celle de nos modèles. On remarque en particulier que la dispersion est moins importante pour le recalage sur l'os de la mâchoire (pratiquement plat, fig. 8.8 à droite) que pour le recalage sur les dents (où l'on a de fortes courbures, fig. 8.8 au milieu).

8.7.3 Étude et correction de la loi sur l'échantillonnage

Nous allons dans un premier temps étudier la loi empirique $p_{\mathbf{u}}$ sur l'échantillonnage, i.e. sur la distribution des éloignements et donc des distances entre points. On utilise logiquement l'histogramme des distances entre points, que l'on peut obtenir en intégrant la loi empirique suivant la direction θ et l'orientation de la normale n :

$$p_{\mathbf{u}}(r) = \int p(r,\theta,n).d\theta.dn$$

Pour les échantillonnages indépendants, on s'attend (figure 8.9 droite) à une probabilité uniforme. C'est effectivement le cas, la partie gauche de la courbe (pour $d < 0.2$) étant probablement sous-estimée pour des raisons purement numériques (rappelons que nous avons divisé par x, que nous avons sur-estimé au voisinage de l'origine pour éviter les

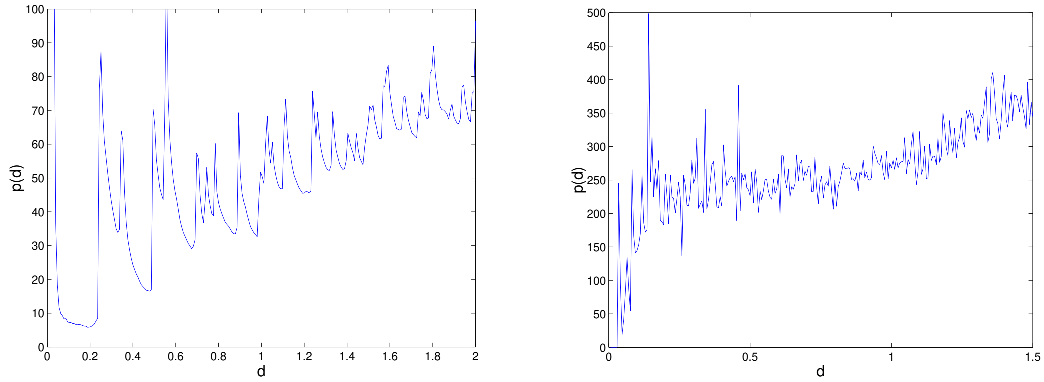


FIG. 8.9 – *Histogramme des distances pour une segmentation scanner à gauche et pour un recalage sur les dents à droite.*

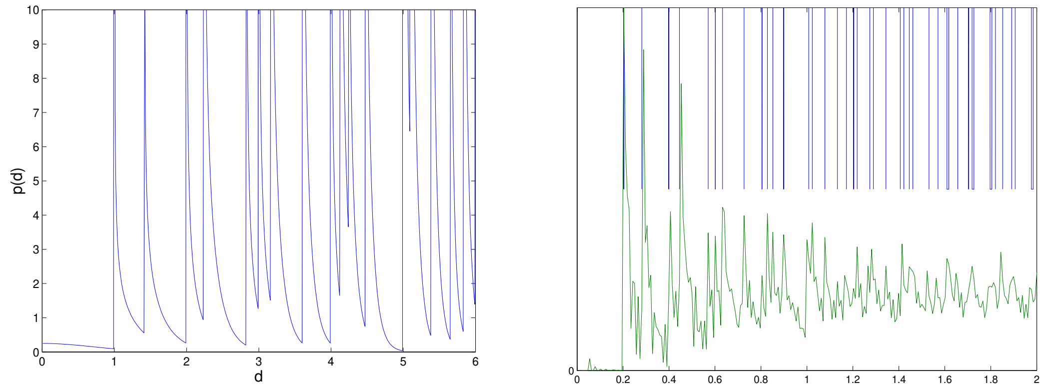


FIG. 8.10 – *Comparaison des distances pour un Picza : histogramme théorique (à gauche) et empirique (à droite, courbe du bas; on a reporté en haut les positions des pics du modèle théorique).*

instabilités), et la partie droite légèrement croissante à cause des points aberrants, dont le nombre augmente avec la distance.

En revanche sur les échantillonnages réguliers tel qu'une segmentation scanner ou un Picza, on voit apparaître des franges caractéristiques (figure 8.9 gauche). On retrouve d'ailleurs une parfaite adéquation entre l'histogramme théorique d'un Picza (voir section 8.4.4) et l'histogramme empirique (voir figure 8.10), et pouvons mesurer le pas d'échantillonnage en ajustant l'échelle horizontale du modèle théorique.

Perspective

Indépendamment de tout modèle, on pourrait corriger notre loi empirique en le divisant par la probabilité empirique de la distance :

$$p'(s,n_s) = p(s,n_s)/p_{\mathbf{u}}(\|s\|)$$

On devrait alors retrouver une loi $p'(s,n_s)$ qui simule celle qui serait obtenu avec un échantillonnage uniforme $p'_{\mathbf{u}}(\|s\|) = Cte$. Nous n'avons pas véritablement testé cette méthode.

8.7.4 Étude du bruit de mesure

Nous allons maintenant évaluer le bruit de mesure. Pour cela, il faut se rappeler que pour toute surface suffisamment bien échantillonnée, l'approximation plane est statistiquement valable (elle n'est pas valable pour les endroits où la surface change brusquement, mais ces cas restent rares, lorsque l'échantillonnage est suffisamment fin) au voisinage du point connu. Il faut donc regarder ce qu'il se passe au voisinage de l'origine[37], où l'approximation plane impose une densité pour les points de la forme $\exp\left(-\frac{y^2}{2.\sigma_m^2}\right)$ et une normale déterministe. La variance empirique de $y = \sqrt{\|ms\|^2 - (n_m|ms)^2}$ par rapport à 0 est alors un estimateur de la variance σ_m^2 du bruit de mesure en position, et la variance empirique des normales autour de la normale origine est un estimateur de la variance du bruit de mesure sur la normale. Nous n'étudierons pas la qualité de ces estimateurs en détail, puisque la section suivante permet d'estimer le bruit de mesure en même temps que le modèle de surface.

8.7.5 Étude du modèle de surface

Il faut maintenant caractériser le modèle qui représente la forme de la surface, i.e. qui contrôle la dispersion des points autour du plan tangent et, pour chaque point, la répartition des normales. La répartition empirique correspond-elle à un de nos modèles théoriques (affine, SCF ou sphérique) ? Nous allons ici répondre à cette question.

Dispersion des points autour du plan tangent Nous allons dans un premier temps nous intéresser à la répartition des points, toutes normales confondues. Nous allons pour cela mesurer la variance de la dispersion des points autour du plan tangent (définie à la section 8.7) et la comparer à la formule théorique :

$$\sigma_{y|x}^2 = \sigma_{forme}^2.x^{2A} + \sigma_{n_m}^2.x^2 + \sigma_m^2$$

On trace donc la valeur empirique de cette variance en fonction de x, et on estime nos paramètres σ_{forme}^2, A, $\sigma_{n_m}^2$ et σ_m^2 qui permettent la meilleure approximation de celle-ci (cf annexe F.5). Ainsi, pour les recalages, on trouve fig. 8.11 des résultats très intéressants :

- Sur l'os (à gauche), peu courbé, on a $A = 1$, ce qui signifie que le modèle plan convient parfaitement. Les deux variances $\sigma_{n_m}^2$ et σ_{forme}^2 jouent alors exactement

37. En pratique on prendra tous les points de l'histogramme correspondant à $x < x_0$, où x_0 est l'abscisse où la dispersion des points autour du plan tangent devient significative.

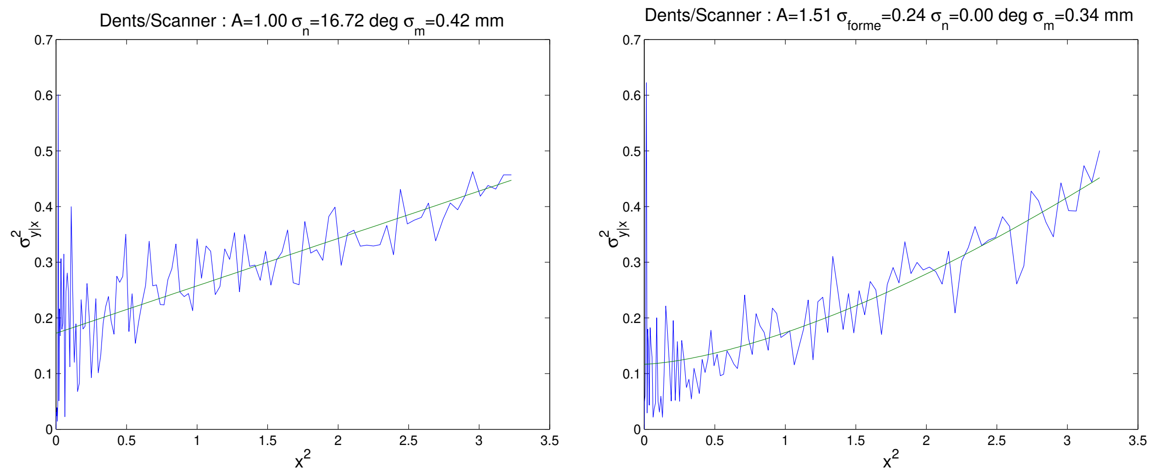


FIG. 8.11 – *Étude de la forme des lois empiriques par le biais de la variance de la dispersion $\sigma^2_{y|x}$: on compare ici la courbe empirique et le modèle le mieux adapté à celle-ci, pour un recalage Os/Scanner (à gauche) et un recalage Dents/Scanner (à droite).*

le même rôle. Il faut donc considérer leur somme, pour obtenir finalement l'écart-type sur l'orientation $\sigma_{n_m} = 16\,deg$, ce qui semble réaliste. L'estimation du bruit de mesure en position $\sigma_m = 0.32\,mm$ est elle-aussi réaliste.

– Sur les dents (à droite), la courbure n'est plus négligeable, et on constate qu'un modèle stochastique convient parfaitement (A est proche de 1.5) avec un paramètre de $\sigma_b^2 = \sigma_{forme}^2/0.34 = 0.8\,rad^2/mm$. Par contre, l'effet de dispersion du modèle de forme masque complètement celui induit par le bruit sur les normales, ce qui explique que l'on trouve un σ_{n_m} quasi-nul. Enfin, l'écart-type du bruit de mesure en position $\sigma_m = 0.26\,mm$ est lui tout à fait réaliste.

Répartition des normales Reste maintenant à étudier, pour chaque point, la répartition des normales. Cette étude est en-fait indépendante de ce qui précède, car elle est indépendante de la probabilité du point.

Par contre il faut tenir compte de la présence d'un bruit de mesure sur le point de la scène. Nous étudierons cette complication en détail dans le recalage (section 8.9), et nous contenterons ici de donner les résultats :

– Pour le bruit de mesure sur la position du point, sa prise en compte dans notre modèle est identique pour le point de la scène et du modèle, si bien que l'on peut ramener la totalité du bruit de mesure sur le seul point du modèle (en prenant la somme des variances des deux bruits de mesure). C'est ce que nous avons implicitement fait jusqu'ici (en particulier, la variance σ_m^2 mesurée correspond en fait à la variance cumulée du bruit sur la position des points du modèle σ_m^2 et de la scène σ_s^2).

- En revanche, pour l'orientation, le bruit sur le modèle intervient dans la formation de la loi de probabilité de la primitive de la scène connaissant le modèle (et a donc une influence sur la répartition des normales en un point donné, mais aussi sur la répartition des points), alors que le bruit de mesure sur la scène vient simplement modifier la répartition des normales en un point s donné. Lorsqu'il est indépendant, il augmente simplement la variance de celles-ci.

Mais ceci reste très théorique. En-effet, sur les champs empiriques, la trace des tenseurs (qui sert à estimer $p(s)$) est déjà très bruitée, et la forme des tenseurs (qui sert à estimer $p(n_s|s)$) l'est encore beaucoup plus. Elle est en pratique inutilisable, sauf si l'on réduit considérablement la résolution de notre champ simulé. On peut alors étudier quantitativement la répartition globale des orientations, mais on ne peut en revanche pas étudier finement son évolution lorsqu'on s'éloigne du point connu.

Ainsi, sur nos exemples, on vérifie que la normale la plus probable correspond à la normale à l'origine (il semble aussi qu'elle s'en écarte lorsqu'on s'éloigne du plan tangent, dans la direction que prévoyait nos modèles théoriques, mais cette analyse reste qualitative), et que l'écart-type sur celle-ci est de l'ordre de $15deg$, $25deg$ pour les recalages pointés sur les dents / scanner[38].

8.7.6 Conclusion

On peut donc estimer assez précisément (sauf pour la répartition des normales) les paramètres du modèle théorique qui s'ajuste bien aux répartitions empiriques des points dans les données.

De manière générale, nous préconisons cette étude empirique qui permettra bien souvent un réglage optimal des paramètres des applications. Ainsi, pour le recalage de points isolés sur les dents, on estime un bruit de mesure d'écart-type $\sigma_m = 0.3mm$, mais la dispersion ne se fait vraiment sentir qu'à partir de $1mm$. Or les résultats expérimentaux du recalage sur les mêmes données (section 6.4.3) se sont avérés optimaux pour $\sigma = 0.3mm$, et ont montré que l'on pouvait augmenter la variance σ_t dans le plan tangent jusqu'à $1mm$ pour robustifier l'algorithme sans que la précision se dégrade notablement. Le réglage des paramètres à l'aide de l'étude empirique des données est donc, sur cet exemple, quasi-optimal.

38. La variance sur les normales ne semble en revanche pas augmenter significativement lorsqu'on s'éloigne du point connu, contrairement à ce que prévoyaient les modèles théoriques. L'explication est probablement que la variance due au modèle reste négligeable face à la variance du bruit de mesure sur les normales de la scène. C'est probablement cette dernière que l'on mesure ici.

8.8 Perspectives pour les applications existantes

Le début de ce chapitre aura permis de justifier et de régler empiriquement les modèles qui permettent de calculer la probabilité d'une primitive connaissant une et une seule primitive. Nous allons maintenant voir, dans les grandes lignes, comment utiliser ces modèles dans des applications variées. Il nous faudra en général être capable de mélanger les informations données par plusieurs primitives connues. Nous parlerons alors de *vote* pour désigner l'information (la probabilité $p((s,n_s)|(m_j.n_{m_j}))$ donnée par chaque primitive $(m_j.n_{m_j})$ et *d'accumulation de votes* pour désigner l'information (la probabilité $p((s,n_s)|(m_1.n_{m_1}),...,(m_j.n_{m_j}))$ obtenue en prenant en compte les différentes primitives connues. Nous verrons donc ici, comment réaliser l'accumulation des votes, et l'utiliser pour écrire une probabilité a-posteriori ou une vraisemblance (adaptée à une application donnée), que l'on va chercher à maximiser.

8.8.1 Filtrage et segmentation d'images

Nous passons ici très rapidement sur deux applications pour le traitement d'images, pour pouvoir ensuite nous focaliser sur les applications concernant les nuages de points.

Le filtrage d'images basé sur la saillance a été présenté par [August, 2001]. Celui-ci dérive, à partir des modèles stochastiques, un nouveau noyau de convolution permettant le lissage anisotrope (et donc le réhaussage) d'images 2D de contours (comme des angiographies). Les résultats qu'il obtient sur des images très bruitées sont remarquables.

L'application qui suit concerne toujours les images 2D de contours, mais il s'agit maintenant de segmenter les contours. La première définition de la saillance avait d'ailleurs été introduite par [Sha'ashua and Ullman, 1988] pour résoudre ce problème. [Williams and Thornber, 2000] a récemment montré comment adapter ces travaux en utilisant les définitions statistiques de la saillance telles celle présentée ici. Le problème de la segmentation apparaît alors comme un problème de maximisation de la vraisemblance. Les résultats expérimentaux ont là aussi prouvé une grande résistance au bruit.

Nos travaux pourraient permettre, dans les deux cas, d'adapter les méthodes aux courbes et surfaces en 3D.

8.8.2 Complétion de courbes et de surfaces

La complétion de courbes en 2D a été un des premiers problèmes auquel le SCF a été appliqué. Nous présentons ici un aperçu des travaux de [Williams and Jacobs, 1997].

Le problème est le suivant : on connaît deux points (m_1,n_{m_1}) et (m_2,n_{m_2}) de la courbe, et on cherche à interpoler la courbe située entre ces deux points. On commence donc par écrire la probabilité qu'un point (s,n_s) appartienne à cette courbe, connaissant (m_1,n_{m_1}) et (m_2,n_{m_2}). On suppose pour cela que la courbe suit un processus stochastique sans mémoire, ce qui permet de justifier que les deux parties (avant et après (s,n_s)) sont indépendantes, et que la probabilité couplée est donc le produit des probabilités individuelles :

$$p((s,n_s)|(m_1,n_{m_1}),(m_2,n_{m_2})) = p((s,n_s)|(m_1,n_{m_1})).p((s,n_s)|(m_2,n_{m_2}))$$

Il faut donc ici multiplier les votes pour les accumuler. On peut alors définir la vraisemblance d'une courbe, et chercher finalement la courbe la plus probable entre les deux points. Dans le cas du SCF, ceci revient à minimiser l'énergie élastique de la courbe [Mumford, 1994], et la meilleure courbe est la spirale d'Euler (i.e. la courbe où la variation de courbure est constante).

Lorsqu'on a maintenant plusieurs points de la courbe, et que l'on connaît leur ordre (i.e. lorsque les points sont structurés), on peut appliquer ce qui précède à chaque paire de points adjacents pour trouver la meilleure courbe passant par ces points. Nous avons donc là une approche bayésienne à l'interpolation de courbes à partir d'un échantillonnage structuré.

Nos travaux permettent de proposer de nouvelles applications à ces techniques :

- L'interpolation de courbes structurées en 3D.
- L'interpolation de surfaces structurées : il s'agirait d'écrire, lorsqu'on dispose d'un nuage de points triangulés, la probabilité d'un point connaissant les 3 sommets du triangle auquel il est rattaché.
- L'approximation de surfaces structurées : les travaux présentés ici ne prennent pas de bruit de mesure en compte. Les courbes interpolées passent donc exactement par les points fournis. L'incorporation d'un bruit de mesure pourrait peut-être assouplir cela, et donner une interpolation approximante.

Remarque pour l'implémentation

Comme nous l'avons expliqué à la section 7.3.3.6, il n'existe pas de formule simple pour calculer le tenseur du produit de deux probabilités - ici $T(s|(m_1,n_{m_1}),(m_2,n_{m_2}))$ - à partir du tenseur de chacune de ces probabilités - ici $T(s|(m_1,n_{m_1}))$ et $T(s|(m_2,n_{m_2}))$. On ne peut donc ici travailler avec les champs de tenseurs, il faut travailler directement avec les lois de probabilités.

8.8.3 Nettoyage des nuages de points et calculs des informations différentielles

Il s'agit ici d'améliorer la technique d'interpolation des normales présentée au chapitre précédent (section 7.4), i.e. formaliser la première phase (l'encodage) du vote de tenseur de Médioni [Médioni *et al.*, 2000].

La technique utilisée ici pour intégrer les votes de plusieurs points connus s'appuie sur une probabilité a-priori pour chaque point connu $p(m_i) = \pi_i$, et doit alors faire la somme pondérée de chaque vote :

$$p((s,n_s)/M) = \sum_i p((s,n_s)/(m_i,n_{m_i})).p((m_i,n_{m_i})) = \sum_i \pi_i.p((s,n_s)/(m_i,n_{m_i}))$$

Ce calcul est fait pour chacun des points d'entrée, et on peut alors extraire la normale la plus probable, mais aussi calculer la pertinence de ces points d'entrée (et nettoyer ainsi le nuage, en éliminant les points aberrants). On peut enfin envisager le calcul d'informations différentielles d'ordre supérieur (courbures comme dans [Tang and Medioni, 1999][39], ...).

Ces travaux avaient été notre point de départ, et il serait maintenant intéressant d'y revenir, en particulier pour voir si nos modèles fonctionnent mieux que ceux utilisés par Médioni (et en particulier remplacer le modèle de la section 7.4.1 par notre modèle boule utilisé lorsque les normales aux points d'entrées ne sont pas connues, voir section 8.6.1.3).

Remarque pour l'implémentation

Comme nous l'avons expliqué à la section 7.4.2, le tenseur d'une loi obtenue en faisant la somme pondérée de plusieurs lois correspond à la somme pondérée des tenseurs de ces différentes lois :

$$p((s,n_s)/M) = \sum_i \pi_i.p((s,n_s)/(m_i,n_{m_i}))$$

devient :

$$T(s/M) = \sum_i \pi_i.T(s/(m_i,n_{m_i}))$$

On pourra donc directement faire les calculs avec les tenseurs.

8.8.4 Reconstruction de surface

Il s'agit ici d'expliquer la deuxième phase du moteur de reconstruction de Médioni (la densification des votes et l'extraction des courbes et surfaces les plus "saillantes"), qui permet la reconstruction de courbes et surfaces à partir d'un nuage de points non structurés. Cette reconstruction est particulièrement intéressante car elle est approximante : la triangulation obtenue ne passera pas forcément par les points fournis, elle peut s'en écarter lorsqu'il semblent bruités ou même les ignorer lorsqu'ils semblent aberrants.

39. Notons que [Cohen-Steiner and Morvan, 2002] a dérivé, dans le cadre de la géométrie algorithmique, une technique permettant de calculer les courbures sur une surface triangulée. Cette technique, qui fait pourtant appel à des concepts complètement différents, s'implémente elle-aussi avec un vote de tenseur.

8.8.4.1 Techniques existantes

La prise en compte des votes se fait comme dans la section précédente, par sommation pondérée par une probabilité a-priori (en général uniforme). On peut ainsi calculer une carte dense de la probabilité $p(s,n_s)$ d'avoir une primitive de la surface en (s,n_s), connaissant les points d'entrée. Reste alors à extraire les courbes et surfaces les plus probables. On peut envisager plusieurs approches.

Lignes de crêtes de $p(s)$ On peut chercher la courbe ou la surface formée des points qui maximisent la probabilité du seul point. On cherche alors les lignes (ou les surfaces) de crêtes de $p(s)$. On obtient ainsi un algorithme simple, mais qui ne tient pas compte de l'information sur la normale en s.

Technique de Médioni On peut chercher la courbe ou la surface pour laquelle la saillance (la probabilité $p(s,n_s)$ ou l'une des variantes discutées dans la section 7.4.4) est maximale dans les directions perpendiculaires à l'espace tangent. C'est ce que fait Médioni : il cherche d'abord, pour chaque point s, l'espace tangent le plus probable (via la tangente ou la normale la plus probable d'après $p(n_s|s)$, i.e. la première valeur propre du tenseur $T(s)$), puis calcule la dérivée de la saillance dans cet espace tangent ainsi choisi. Il extrait ensuite les 0 de cette dérivée qui donnent donc les minima et les maxima de la saillance perpendiculairement à l'espace tangent, qu'il faut enfin trier.

L'avantage de cette technique est qu'elle peut être implémentée efficacement, via un Marching-Cube (pour les surfaces) ou un Marching-Lines (pour les courbes) légèrement modifié.

Le problème est que les espaces tangents de la courbe ou surface ainsi segmentée ne correspondent pas forcément à ceux que l'on a choisi en début d'algorithme : il arrive en-effet qu'il faille faire un compromis, en prenant une surface dont les normales n_s ne sont pas point par point les plus probables ($p(n_s|s)$ n'est pas maximal pour chaque point s) parce qu'elles permettent d'augmenter localement la saillance (i.e. l'intégrale de $p(s,n_s)$ sur un petit bout de surface).

8.8.4.2 Vers une technique de type EM

L'exemple précédent montre qu'il ne faut pas chercher les points les plus probables pour en faire une courbe, mais s'intéresser directement à la courbe ou la surface en tant que telle. C'est ce que nous allons faire ici dans le cas des courbes, en essayant de présenter les grandes lignes d'une approche véritablement justifiée, de type EM.

Remarque sur les notations Nous allons ici inverser les notations sur les points, pour qu'elles soient identiques à celles que nous avons déjà utilisé pour l'EM : les points connus sont les points s_i (qui forment la scène $\mathcal{S}$), les points de la courbe que l'on cherche à retrouver sont les points m_j (qui forment le modèle $\mathcal{M}$).

Représentation de la courbe On va donc chercher ici, d'une façon ou d'une autre, à optimiser un critère qui dépend de la courbe cherchée $\mathcal{C}$. Il faut avant cela choisir une représentation de cette courbe. Nous allons travailler ici avec une représentation discrète de la courbe, i.e. regarder la position des points correspondant à des abscisses curvilignes u_j discrètes :

$$m_j = m(u_j)$$

On définit la tangente (à peu près normalisée, car nous travaillons avec les abscisses curvilignes) :

$$t_j = \frac{m_{j+1} - m_j}{u_{j+1} - u_j}$$

Approche "Maximum A Posteriori" ou "Maximum de Vraisemblance" Maintenant que nous avons une représentation paramétrique de la courbe $\mathcal{C}$, on peut écrire la règle de Bayes avec les données $\mathcal{D}$ (ici l'ensemble des points connus s_i) :

$$p(\mathcal{D}|\mathcal{C})p(\mathcal{C}) = p(\mathcal{C}|\mathcal{D})p(\mathcal{D})$$

Les différents termes sont :

- La probabilité a-priori des données $p(\mathcal{D})$: la répartition a-priori des données est uniforme.
$$p(\mathcal{D}) = Cte$$
- La probabilité a-priori de la courbe $p(\mathcal{C})$: on peut prendre là aussi une répartition uniforme, ou prendre en compte une loi a-priori sur les courbes. Avec un modèle de type marche aléatoire orientée sans mémoire (qui permet de favoriser les courbes plus courtes et moins courbées) :
$$p(\mathcal{C}) = \prod_j p(t_{m_{j+1}}|t_{m_j}) = \prod_i \exp(-\frac{(\widehat{t_{m_{j+1}},t_{m_j}})^2}{2.\sigma_b^2(u_{j+1} - u_j)})$$
- La *vraisemblance*, i.e. la probabilité des données connaissant la courbe $p(\mathcal{D}|\mathcal{C})$: c'est ce que les modèles présentés dans ce chapitre vont permettre de calculer.
- La *probabilité a-posteriori* de la courbe $p(\mathcal{C}|\mathcal{D})$. Elle serait difficile à écrire directement en essayant de mélanger les probabilités de chaque point de la courbe connaissant les données $p((m_i,t_{m_i})|(s_i,t_{s_i}))$, car ces différentes lois ne sont pas indépen-

dantes. En revanche, elle peut-être déduite de la règle de Bayes :

$$p(\mathcal{C}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{C})p(\mathcal{C})}{p(\mathcal{D})} = Cte.p(\mathcal{D}|\mathcal{C})p(\mathcal{C})$$

C'est l'une de ces deux dernières que l'on va chercher à maximiser. Nous présentons ici le cas de la vraisemblance (i.e. l'approche MV). La dernière équation permettra de s'adapter à la maximisation de la probabilité a-posteriori (l'approche MAP) lorsqu'on veut prendre en compte une loi a-priori sur la courbe $p(\mathcal{C})$.

Probabilité des données connaissant la courbe Nous avons renversé le problème, en cherchant à définir la probabilité de nos données en fonction de la courbe, et non la probabilité de la courbe en fonction des données. Or, si les probabilités des différents points de la courbe sont corrélées, on peut en revanche souvent supposer que les données (s_i,t_{s_i}) sont des mesures indépendantes (cela dépend en-fait de la méthode d'acquisition; ce n'est par exemple pas tout à fait vrai pour le scanner ou les pointés en mode rafale). On peut donc prendre le produit de leur probabilité :

$$p(\mathcal{D}|\mathcal{C}) = \prod_i p((s_i,t_{s_i})|\mathcal{C})$$

Reste donc à écrire la probabilité d'obtenir (s_i,t_{s_i}) en cherchant à mesurer un point de la courbe $\mathcal{C}$. Ce point peut alors en-fait être la mesure de n'importe quel point (m_j,t_{m_j}) de la courbe. En notant π_j la probabilité a-priori de que s soit la mesure du point m_j (dans la plupart des cas, π_i sera uniforme), et $p((s_i,t_{s_i})|(m_j,t_{m_j}))$ la probabilité de mesurer le point (s_i,t_{s_i}) sachant que la courbe passe par (m_j,t_{m_j}) (nous prendrons pour celle-ci un le modèle le mieux adapté aux données empiriques) :

$$p((s_i,t_{s_i})|\mathcal{C}) = \sum_j \pi_j.p((s_i,t_{s_i})|(m_j,t_{m_j}))$$

Nous obtenons ainsi la vraisemblance de la courbe :

$$p(\mathcal{D}|\mathcal{C}) = \prod_i \sum_j \pi_j.p((s_i,t_{s_i})|(m_j,t_{m_j}))$$

Optimisation Nous retrouvons là un cas typique où il faut appliquer l'EM : la vraisemblance est de la forme $\prod_j \sum_i \pi_i p((s_i,t_{s_i})|(m_j,t_{m_j}))$ (analogue par exemple à l'eq. 4.5), et cache en-fait une variable cachée (celle qui indique de quel point (m_j,t_{m_j}) de la courbe provient la mesure (s_i,t_{s_i})).

On peut donc directement donner l'algorithme (le lecteur pourra reprendre la section 4.3 pour comprendre comment on y arrive) :

Initialisation Il faut ici une première estimée de la courbe. Il pourra par exemple s'agir de celle donnée par l'une des deux techniques existantes que nous avons présenté.

Étape E Il faut ici estimer la matrice d'appariements, i.e. estimer de quel point (m_j,t_{m_j}) le point (s_i,t_{s_i}) est la mesure. On a :

$$A_{ij} = \frac{\pi_j.p((s_i,t_{s_i})|(m_j,t_{m_j}))}{\sum_k \pi_k.p((s_i,t_{s_i})|(m_k,t_{m_k}))}$$

Étape M Il faut maintenant ré-estimer la courbe, i.e. ré-estimer la position des points (m_j,t_{m_j}) en minimisant l'opposé de l'espérance par rapport à A de la log-vraisemblance :

$$\min_{(m_j,t_{m_j})} \sum_i A_{ij}.\left(-\log p((s_i,t_{s_i})|(m_j,t_{m_j}))\right)$$

Cette optimisation doit se faire sous-contrainte, pour prendre en compte la définition de la représentation paramétrique de la courbe $\mathcal{C}$ (ici, elle impose $\left\|t_{m_j}\right\| = 1$ et $s_{j+1} - s_j = (u_{j+1} - u_j).t_{m_j}$).

Nous avons donc là obtenu un algorithme parfaitement justifié sur le papier. Nous nous ne l'avons malheureusement pas testé, et n'avons donc pas pu évaluer son efficacité.

8.8.4.3 Autres perspectives

L'approche de Médioni était directe et non paramétrique. La méthode basée sur l'EM est itérative et paramétrique, et nécessite une initialisation. D'un point de vue de l'efficacité algorithmique, c'est très décevant. Aussi pensons-nous qu'il faut continuer les recherches sur ces algorithmes. De nombreuses pistes se présentent :

- Aborder le problème de manière variationnelle, par exemple en corrigeant le défaut constaté dans la méthode de Médioni.
- Finir les calculs pour l'EM en espérant trouver une simplification qui permette une résolution directe.
- Chercher une méthode intermédiaire.

Notons enfin que, quel que soit l'algorithme utilisé, l'incorporation du bruit de mesure dans les modèles pourrait permettre de contrôler beaucoup mieux l'aspect approximant de la reconstruction (voir figure 8.12).

8.8.5 Nouvelles applications

A ce stade, on peut envisager de trouver de nouveaux débouchés aux notions de saillances. Pratiquement n'importe quelle application sur des surfaces échantillonnées brui-

FIG. 8.12 – *Exemple sur une courbe des résultats que l'on souhaiterait obtenir avec la reconstruction :*
A gauche : Avec un modèle autorisant un bruit de mesure important, on souhaiterait se rapprocher d'une ligne droite
A droite : Avec un modèle autorisant une courbure importante (mais un bruit de mesure faible), on souhaiterait se rapprocher des points d'entrée

tées pourraient tirer profit de notre modélisation. Citons notamment les statistiques de formes, la reconstruction de surfaces 3D à partir d'une ou plusieurs vues 2D, ... Rappelons aussi que l'équipe de Médioni a déjà appliqué le vote de tenseurs à d'autres sujets comme le suivi de mouvements dans des séquences [Nicolescu and Médioni, 2002], ...

8.8.6 Conclusion

Cette section, très prospective, présente en particulier deux façons d'accumuler les votes :

- En multipliant les votes de points voisins (comme dans la complétion de courbes, section 8.8.2) : cette technique est importante, car elle permet de prendre la structuration en compte lorsqu'elle est disponible.
- En additionnant les votes (comme dans l'interpolation de la normale et des informations différentielles, section 8.8.2) : cette technique sert lorsque la structuration n'est pas disponible.

Elle montre aussi, sur certaines applications, comment dériver des algorithmes bayésiens à partir de ces votes pour répondre au problème posé.

8.9 Recalage de surfaces aléatoires

L'idée originelle de cette deuxième partie de thèse était de trouver une probabilité d'élément de surface pour adapter l'ICP/EM du chapitre 4. Nous espérions ainsi généraliser les techniques utilisant les orientations des normales présentées à la section 5.2 et 5.3 et tirer de ce cadre général une probabilité plus représentative des surfaces, dans l'espoir d'améliorer encore la précision et la robustesse, déjà satisfaisantes, et surtout d'apporter une meilleure réponse aux problèmes de prédiction de la précision (section 5.4), visiblement plus sensible aux hypothèses et aux paramétrages des algorithmes. Nous faisons donc ici la jonction entre nos deux parties.

8.9.1 Algorithme

Rappelons l'algorithme de l'ICP/EM :

Répéter

Décimation : Décimer la scène avec une sphère de rayon $\alpha.\sigma$.

Étape E : Pour chaque point orienté de la scène (s_i,n_{s_i}) décimée avec poids de décimation w_{s_i}
Chercher tous les points du modèle (m_j,n_{m_j}) tel que $\mu^2((s_i,n_{s_i}),(m_j,n_{m_j})) < \mu^2_{max}$ en utilisant un kD-Tree
Calculer les poids $A_{ij} = \frac{p(T\star(s_i,n_{s_i})|(m_j,n_{m_j}))}{\sum_k p(T\star(s_i,n_{s_i})|(m_j,n_{m_j}))}$

Étape M : Ré-estimer T en minimisant $-\sum_{ij} w_{s_i}.A_{ij}.\log p(T\star(s_i,n_{s_i})|(m_j,n_{m_j}))$

Recuit : Ajuster σ^2

Jusqu'à convergence

L'idée pour adapter l'ICP/EM est bien-sûr d'utiliser un de nos modèles en guise de loi de probabilité $p(T\star(s_i,n_{s_i})|(m_j,n_{m_j}))$. Pour savoir quel modèle utiliser, on cherchera le modèle le mieux adapté aux données, comme dans la section 8.7. Nous avions d'ailleurs présenté dans cette section le cas des recalages utilisés dans VirtualScope :

- Pour le recalage Scanner/Pointés sur l'os, le champs affine convenait parfaitement.
- Pour le recalage Scanner/Pointés sur les dents, le champs stochastique semblait mieux adapté.

Remarquons que ces modèles ne prennent en compte que le bruit de mesure sur (m_j,n_{m_j}). Or dans le recalage (et contrairement aux autres applications), le point (s_i,n_{s_i}) est lui aussi une mesure bruitée d'un point de la surface. Il faut donc rajouter à nos modèles de vote ce dernier phénomène aléatoire. Nous le supposerons indépendant (hypothèse en général vérifiée, car les points du modèle et de la scène sont acquis par des méthodes différentes). On montre alors facilement comment le prendre en compte :

- Pour la position du point, nous retrouvons une convolution par la loi du bruit de mesure en position sur le point de la scène. On doit donc convoluer le modèle sans bruit avec la loi du bruit de mesure pour le point du modèle puis avec celle pour le point de la scène. Mais il est équivalent de convoluer les deux lois des deux bruits de mesures (dans le cas de bruits gaussiens, on obtient une gaussienne dont la variance est la somme des variances de chaque bruit), et de convoluer le résultat avec le modèle sans bruit :

$$\left(p(T\star(s_i,n_{s_i})|(m_j,n_{m_j})) * p_{e_{m_j}}\right) * p_{e_{s_i}} = p(T\star(s_i,n_{s_i})|(m_j,n_{m_j})) * \left(p_{e_{m_j}} * p_{e_{s_i}}\right)$$

  Autrement dit, on peut reporter le bruit sur le point de la scène dans le bruit sur le point du modèle.

– Pour l'orientation de la normale, les choses sont différentes : la normale au point du modèle influence la répartition globale des points et normales (et donc $p(s,n_s)$ en tout point). En revanche, le bruit sur l'orientation vient simplement modifier la répartition des normales en s (et donc $p(n_s|s)$). Dans le cas d'un bruit gaussien indépendant de variance $\sigma_{t_s}^2$, il suffit d'augmenter de $\sigma_{t_s}^2$ la variance des normales au point s.

Le modèle peut donc être calculé dans l'ordre suivant : prise en compte de la loi sur la forme, sur l'échantillonnage et sur le bruit de mesure sur n_{m_j} (on utilise l'équation 8.9), puis convolution avec les deux bruits de mesure sur la position, et enfin prise en compte du bruit de mesure sur n_{s_i} (par simple accroissement de $\sigma_{t_s}^2$ de la variance des normales en chaque point s).

Remarques pour l'implémentation

Reste à adapter effectivement l'algorithme. Beaucoup de questions se posent, auxquelles l'on va essayer de répondre dans le désordre :
Distance de recherche et rejet des points aberrants : comment l'adapter ?
Calcul des poids : comment calculer efficacement la probabilité pour deux points donnés ?
Calcul de la transformation : comment minimiser le critère ?
Recuit : quelle échelle est concernée ?
Décimation : la décimation par les sphères est-elle justifiée ? Plus généralement, comment faire la multi-résolution, ou plutôt de quelle façon peut-on décimer la surface en maintenant au mieux les performances du recalage ?
Nous allons ici détailler les trois premières. Les deux dernières, liées aux problèmes d'échelles n'ont pas encore été résolues.

Calcul de la probabilité

Nous allons devoir calculer très souvent (pour chaque paire de points trouvée) la probabilité d'un point orienté de la scène connaissant un point orienté du modèle. Ceci représente à peu près le tiers du temps de calcul dans l'ICP/EM classique. Il faut donc veiller à avoir une méthode de calcul efficace. Nous utiliserons une loi de probabilité $p(T \star (s,n_s)|(m,n_m))$ pré-calculée suivant la méthode présentée à la section 8.6.3. Nous utiliserons comme d'habitude les idées de la section 8.2 pour se limiter à un champs de tenseur dans la plan xy, que nous échantillonnerons sur une grille régulière. Pour un point $T \star (s,n_s)$ de la scène et un point (m_j,n_{m_j}) du modèle, on ramènera (m_j,n_{m_j}) en l'origine et s dans le plan xy, puis on calculera la probabilité $p(s,n_s)$ par interpolation bilinéaire dans le champs pré-calculé.

Distance de recherche et rejet des points aberrants

Pour pouvoir rejeter les points aberrants, nous avions ajouté, dans l'ICP/EM, un pseudo-point m_{out} dans le modèle, avec une probabilité constante p_{out} pour qu'un point s_i soit une mesure de m_{out}, i.e. un point aberrant. Ainsi, pour un point de la scène s_i donné, les points m_j tels que $p(T \star s_i,m_j) < p_{out}$ seront rejetés dans l'ICP (qui ne garde que le plus probable), ou recevront un poids très faible dans le cas de l'EM.

Avec un bruit gaussien, on avait $p(T \star s_i,m_j) = \exp(-\mu^2(T \star s_i,m_j)/2)$ et on pouvait donc ré-écrire l'inégalité précédente : $p(T \star s_i,m_j) < p_{out}$ devient $\mu^2(T \star s_i,m_j) > \mu^2_{out}$. Il suffisait alors d'utiliser une distance de Mahalanobis maximale, pour rejeter les points de probabilité inférieure à p_{out}. Dans le cas isotrope, cette distance de Mahalanobis était directement reliée à la distance usuelle par $\mu^2 = d^2/\sigma^2$. On pouvait donc utiliser la distance de recherche maximale $d_{max} = \sigma.\sqrt{\mu^2_{out}}$.
Nous procéderons ici de même, en fixant une probabilité p_{out}, et en calculant la distance de recherche d_{max} qui convient. Comme nos modèles ne sont plus isotropes, on peut avoir des points s situés à la même distance d de m mais donnant des probabilités très différentes. Pour une distance d donnée, les points situés sur le plan tangent (sous-entendu : et dont l'orientation correspond à celle du plan tangent) seront les plus probables, et les points dans l'axe de la normale seront les moins probables. Le réglage de la distance de recherche doit tenir compte de ces spécificités, et en particulier permettre de trouver tous les points dont la probabilité est supérieure à p_{out}. On utilisera donc la distance d_{max} qui correspond au premier point du plan tangent dont la probabilité est inférieure à p_{out}. Ensuite, pour chaque point s_i, on cherchera dans un premier temps, à l'aide du kD-Tree les points m_j dans la sphère de centre s_i et de rayon d_{max}. Dans un deuxième temps, on calculera la probabilité $p(T \star (s,n_s)|(m,n_m))$, et on ne gardera que les paires de points pour lesquelles cette probabilité est effectivement supérieure à p_{out}.

Calcul de la transformation

Pour le calcul de la transformation, i.e. la minimisation du critère $-\sum_{ij} w_{s_i}.A_{ij}.\log p(T \star (s_i,n_{s_i})|(m_j,n_{m_j}))$, il n'existe pas de solution explicite. Nous utilisons donc une descente de gradient, que l'on peut implémenter en reprenant les sections 3.8.2 (qui explique la descente de gradient pour l'ICP) et 4.5 (qui explique comment l'adapter à l'ICP/EM). Il faut alors remplacer la distance $D(T \star s,m)$ par la log-probabilité $-\log p(T \star s,m)$, et en particulier calculer son gradient et son hessien. Ces derniers sont en-fait pré-calculés en même temps que la probabilité (lors du filtrage gaussien), i.e. pour (m,n_m) à l'origine et s dans le plan xy, et on utilise les formules de la section 8.2 pour effectuer le calcul pour (m,n_m) et (s,n_s) quelconque.

8.9.2 Résultats préliminaires

Nous n'avons pas pu, faute de temps, tester intensivement cette nouvelle mouture de l'ICP/EM. Nous n'avons à vrai dire fait qu'une seule expérience, basée sur une implémentation encore partielle : on n'a testé que le modèle affine, n'avons pas utilisé les normales aux points de la scène n_{s_i}, et n'avons pas utilisé d'approche multi-échelle. Aussi, nous ne pouvions espérer obtenir de bons résultats en terme de robustesse, et nous sommes concentré sur la précision. Nous avons donc cherché automatiquement les paramètres permettant d'atteindre la meilleure précision, exactement comme nous l'avions fait dans la section 6.4.5 (nous avons en particulier utilisé les mêmes données, la même façon de mesurer la précision).

Le tableau 8.1 suivant compare les performances obtenues avec celles de l'ICP/EM

	Modèle gaussien	Modèle affine
Imprécision moyenne	0.26 mm	0.24 mm
Imprécision max.	0.19 mm	0.17 mm
Nb. d'itérations moyen	14	32
Nb d'itérations max.	23	77

TAB. 8.1 – *Performances comparées de l'ICP/EM classique (avec modèle de bruit gaussien) et l'ICP/EM du chapitre 8 (avec modèle de bruit affine) sur la base de test*

classique (i.e. avec modèle de bruit gaussien), lui aussi optimisé pour la précision. On y voit que cette première implémentation, bien qu'incomplète et mal adaptée au recalage sur les dents (il aurait fallu utiliser le modèle stochastique), a déjà apporté un léger gain (de l'ordre de 10%) en terme de précision, et a surtout permis de diminuer fortement le nombre d'itérations. Ceci n'est guère intéressant en terme de temps de calcul (chaque itération demande à peu près deux fois plus de temps, le temps global de recalage diminue donc assez peu), mais montre probablement que nos modèles de surfaces sont mieux adaptées au recalage.

Nous avons enfin voulu étudier la prédiction théorique de l'incertitude pour notre nouvelle variante de l'ICP/EM. Les rares expériences nous ont amené au mêmes conclusion que pour l'ICP/EM classique (section 5.4.1) : la prédiction peut être relativement bonne lorsque les paramètres sont très bien réglés (ce qui est peut-être ici facilité par l'étude des lois empiriques qui permet de régler finement les paramètres du modèle de surface), mais peut encore varier presque du simple au double autour de la bonne valeur.

8.9.3 Discussion

Nous avons donc adapté l'ICP/EM en utilisant les modèles développés dans ce chapitre. Les premiers résultats avec le modèle affine semblent encourageant : les gains en précision et vitesse sont bien visibles, bien que modestes (mais le modèle affine est encore très proche du modèle gaussien, comme nous l'avons signalé à la section 8.3.3). Il faut donc poursuivre nos efforts avec d'autres modèles.

Par contre, il est deux points que nous n'avons pas encore abordé :

– L'utilisation d'une approche multi-échelle associée à une décimation : nous ne savons pas comment intégrer cette nouvelle mouture de l'ICP/EM dans une approche multi-échelle et multi-résolution, car ce sont deux problèmes que nous avons complètement éludés dans ce chapitre, et qui font partie des principales perspectives signalées dans la conclusion de cette thèse.
– Les systèmes de sécurité : on peut très certainement mettre au point un système de sécurité (cf section 6.2.3) par l'étude de la loi empirique entre les deux surfaces

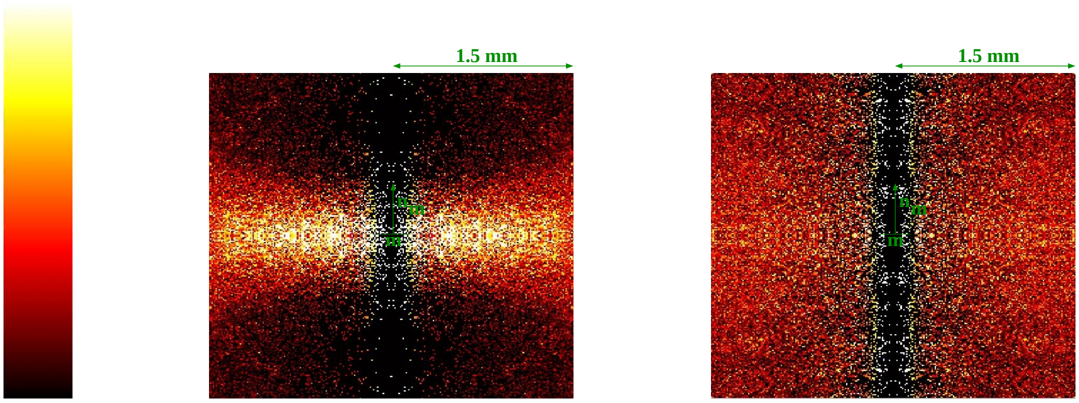


FIG. 8.13 – *Exemple de lois empiriques (cf section 8.7) pour un bon et un mauvais recalage Scanner/Pointés sur les dents.*
A gauche : pour un bon recalage, on retrouve la forme caractéristique de nos modèles de vote.
A droite : pour un mauvais recalage, on retrouve une loi uniforme.

recalées (cf section 8.7), très différente pour un bon et un mauvais recalage, comme le montre l'exemple de la figure 8.13.

8.10 Conclusions

Nous avons, dans ce chapitre, réussi à mettre au point la brique qui manquait à notre ICP/EM : une véritable loi de probabilité pour les éléments de surfaces échantillonnées et bruitées.

Il nous a fallu pour cela regrouper et poursuivre les travaux sur les notions de saillance et les techniques de vote de tenseurs, originellement développés pour des applications différentes de la nôtre (ce qui laisse entrevoir des perspectives nombreuses et importantes, qui seront résumées dans le chapitre suivant).

Nous nous sommes donc appuyés sur les points semi-orientés, et, après avoir montré comment utiliser un champ de tenseur pour représenter et étudier efficacement des lois de probabilité sur ce type de primitives, nous avons montré comment en dériver une à partir de trois composantes :

- Un modèle sur la forme de la surface. Nous avons présenté plusieurs modèles permettant de dériver une loi de probabilité en 2D : un modèle affine, un modèle sphérique et un modèle stochastique. Nous les avons en partie adapté en 3D.
- Un modèle sur l'échantillonnage : nous avons là aussi présenté plusieurs modèles adaptés à des applications variées (échantillonnage régulier, aléatoire uniforme, ...).
- Un modèle sur le bruit de mesure.

Nous avons montré comment mélanger ces 3 composantes, et avons en particulier effectué un calcul approché mais explicite dans les cas les plus simples. La confrontation avec les données empiriques a montré le bien-fondé de notre approche et a permis de régler facilement nos modèles pour une application donnée. Les résultats expérimentaux montrent que le recalage s'en trouve amélioré.

Chapitre 9

Conclusions et perspectives

Cette thèse n'apporte pas d'idées véritablement révolutionnaires. Ceci n'est pas surprenant, dans la mesure où les thèmes abordés ont été abondamment étudiés par la communauté scientifique depuis le début des années 90 : la première partie, consacrée au recalage, repose sur l'ICP introduit par [Zhang, 1992; 1994; Chen and Medioni, 1991; 1992; Besl and McKay, 1992] et la deuxième partie, consacrée à la modélisation des éléments de surfaces échantillonnées bruités, repose sur la notion de saillance telle qu'elle a été présentée par [Mumford, 1994] et [Guy, 1996; Guy and Medioni, 1997].

En revanche, la volonté d'aborder ces travaux en adoptant systématiquement un point de vue statistique[40] a permis de ré-interpréter, sélectionner, regrouper, recouper, formaliser, généraliser et compléter la plupart des idées qui avaient été proposées. Ainsi a-t-on pu proposer une thèse dans laquelle *toutes* les méthodes reposent sur les statistiques usuelles :

- Le problème du recalage est abordé comme un problème de maximisation de la vraisemblance. Cette dernière peut être complètement définie à partir d'un modèle de bruit sur les points. Notre approche permet alors d'adapter l'ICP, l'ICP/EM et la prédiction théorique de leur incertitude à n'importe quel modèle de bruit sur les données. Ceci permet de justifier (ou corriger) et créer de façon remarquablement intuitive des variantes de ces algorithmes qui prennent en compte les spécificités de nos données.
- La modélisation des éléments de courbes et surfaces échantillonnées bruitées repose sur la notion de saillance à laquelle on donne une définition purement probabiliste. Elle permet, dans de nombreuses applications, d'écrire une vraisemblance et d'en

40. J'aimerai ici faire remarquer que, comme j'ai pu le constater à ECCV'02, les techniques statistiques sont en train de rentrer en force dans le monde de la vision par ordinateur. Il y a encore deux ans, la grande majorité des travaux reposaient sur des méthodes "déterministes", i.e. ne prenaient pas explicitement en compte la caractère bruité des données. Maintenant les trois-quarts de travaux s'appuient sur les statistiques, soit en les incorporant aux méthodes existantes, soit en les utilisant pour dériver des méthodes complètement nouvelles. En prenant intrinsèquement en compte du bruit sur les données, elles permettent d'améliorer la précision et surtout la robustesse des algorithmes dans la quasi-totalité des applications où ce bruit est présent.

déduire un algorithme capable de la maximiser.

– Les aspects plus pratiques et expérimentaux utilisent les mêmes outils statistiques. Ceci permet un dialogue fructueux entre les plans pratiques et théoriques. On peut par exemple ajuster les modèles théoriques de bruit sur l'étude empirique des données et savoir ainsi comment régler les algorithmes. On peut aussi confronter l'étude expérimentale et la prédiction théorique de leur précision.

Enfin, nous avons montré, sur l'exemple du recalage, comment mettre en évidence des paramètres d'échelles dans les modèles de bruits. Ceci a permis d'incorporer facilement l'algorithme dans une approche multi-échelle, plus robuste, et nous a donné l'idée de l'associer à une approche multi-résolution, plus rapide.

9.1 Recalage

9.1.1 Interprétation statistique et prédiction théorique de l'incertitude

Nous avons présenté la formalisation statistique du recalage de façon claire en rassemblant et complétant les efforts déjà réalisés dans ce sens. Nous avons montré que l'ICP était tout simplement l'estimateur qui cherchait la transformation et la configuration d'appariements la plus vraisemblable sous hypothèse de bruit gaussien. Nous avons alors calculé la vraisemblance et ainsi généralisé l'algorithme pour une hypothèse de bruit quelconque, puis avons montré comment le modifier à la façon de l'EM pour donner moins d'importance à l'estimation des appariements et rendre ainsi l'algorithme plus robuste.

Le chapitre 5 a prouvé le bien-fondé et la maniabilité de notre approche : nous y avons montré comment adapter aisément nos algorithmes sans perdre leurs bonnes propriétés (une convergence sûre et rapide, une précision quasi-optimale) en modifiant le modèle de bruit. On a ainsi pu estimer un éventuel décalage sur les points, et surtout prendre en compte l'aspect surfacique de nos données, grâce à l'utilisation de la distance entre normale, des modèles semi-anisotropes, puis des modèles plus généraux de la deuxième partie de ce manuscrit.

Enfin, nous avons montré comment adapter la prédiction théorique de la précision de ces algorithmes, en se basant toujours sur le seul modèle de bruit. Nous avons de plus utilisé cette prédiction pour guider le chirurgien lors de l'acquisition de ces nuages de points.

A part cette dernière technique, nous avons validé toutes les étapes de notre approche, et ne voyons guère comment elles pourraient être encore améliorées dans le cadre de VirtualScope. Aussi les perspectives se concentrent-elles dans la possibilité d'essayer des modèles de bruits plus subtils, qui prennent en compte les corrélations ou s'appuient sur les modèles de surfaces échantillonnées bruitées du chapitre 8. On peut même envisager de s'intéresser à d'autres types de recalages auxquels l'ICP a déjà été appliqué, comme le recalage 2D/3D [Nicolau *et al.*, 2003], le recalage non rigide [Feldmar and Ayache, 1996; Rangarajan *et al.*, 1997b], le recalage multiple [Cunnington and Stoddart, 1999] ou le recalage iconique [Feldmar *et al.*, 1997b]. On peut aussi songer à la prise en compte d'autres informations (comme une information a-priori sur la transformation) par le biais d'une approche MAP [Wells, 1997].

9.1.2 Aspects multi-échelles

Nous avons incorporé de façon très intuitive les techniques du multi-échelle à notre approche, ce qui a permis de robustifier le recalage. Mais n'étant pas spécialistes de ces techniques, nous ne l'avons fait que de façon très heuristiques, et une étude un peu plus rigoureuse s'avère maintenant nécessaire pour faire véritablement le lien avec le cadre mathématique du multi-échelle [Lindeberg, 1994]. Ceci permettrait probablement de mieux étudier la robustesse de l'algorithme et de savoir quel stratégie de raffinement progressif de l'échelle choisir.

Nous avons ensuite pensé à associer la multi-résolution au multi-échelle, en décimant nos nuages de points. La technique de décimation utilisée a permis d'accélérer l'algorithme de recalage tout en limitant la dégradation de ses performances. Il nous manque cependant une véritable évaluation mathématique de cette dégradation (que nous conjecturons comme étant optimale car au deuxième ordre, section 4.4.3).

Il nous manque enfin une interprétation vraiment statistique du processus de décimation. Il faudrait probablement pour l'obtenir poser le problème comme celui de la recherche du nuage de points conservant au mieux l'information contenue dans le nuage original. La question se pose alors de savoir quelle définition de l'information utiliser : va-t-on s'appuyer sur une définition générale telle celle de Shanon (et minimiser alors la distance de Kullback-Leibler entre les lois de probabilités représentant les deux nuages), ou une définition de l'information plus spécifique à l'application (l'information utilisée effectivement par l'algorithme de recalage à une échelle donnée, qui ne tient par exemple pas compte de la structuration éventuelle du nuage de points) ?

9.1.3 Performances

De manière générale, notre approche a permis d'améliorer nettement les performances du recalage :

- Précision : l'utilisation de l'ICP/EM a permis une légère amélioration de la précision, principalement due à la résolution des problèmes de répétabilité de l'ICP. Les résultats sont maintenant très proche de la limite de Cramer-Rao (la précision maximale théoriquement atteignable), ce qui nous amène à penser qu'il n'y a plus guère d'améliorations possible.

- Robustesse : l'approche multi-échelle a permis de rendre l'algorithme beaucoup plus robuste. Mais la tolérance sur la transformation initiale, bien que satisfaisante pour notre application, reste très limitée (elle est de l'ordre de $1\,cm$ pour les mâchoires). Il faudrait donc poursuivre les recherches sur la robustesse, soit en réglant mieux la stratégie de raffinement progressif de l'échelle, soit en utilisant des techniques radicalement différentes, telles celles présentées à la section 6.2.2 (recuit-simulé sur la transformation, algorithmes génétiques, ...).
- Temps de calcul : la décimation a permis de diviser le temps de calcul jusqu'à un facteur 10. La possibilité de paralléliser les calculs avec des rendements supérieurs à 90%, même pour un nombre élevé de processeurs, permet de gagner encore un ordre de grandeur, et d'envisager une utilisation temps-réel.

9.1.4 Étude expérimentale

Nous avons mené une étude expérimentale très fine de nos algorithmes, ce qui est encore hélas inhabituel.

Nous avons ainsi mis au point un banc de test entièrement automatisé, permettant l'étude sur une base de plusieurs patients, de la robustesse et de la précision du recalage rigide, même en l'absence de vérité terrain (il a suffi pour cela d'utiliser plusieurs acquisitions des mêmes données pour calculer un étalon bronze ou zinc).

Ce banc de test est conçu pour les transformations rigides, et une première perspective est son adaptation à d'autres types de transformations (e.g. affines, non-paramétriques). Certaines notions (comme l'utilisation de l'erreur de recalage sur cibles) pourront être directement transposées, mais beaucoup (comme la comparaison de deux transformations, utilisée pour l'étude de robustesse, et l'analyse statistique d'une série de transformations, utilisée pour l'étude de précision) pourront poser problèmes.

Dans le cas rigide, l'étude de robustesse semble satisfaisante, mais beaucoup de questions se posent concernant celle de la précision :

- Comment prendre en compte les corrélations entre les différents recalages lorsqu'on dispose de beaucoup de boucles de consistances (i.e. lorsque l'on aura à la fois plusieurs acquisitions de la scène et du modèle).
- Comment estimer la corrélation entre deux algorithmes (ou entre différents paramétrages), pour pouvoir mieux les comparer et comprendre leur fonctionnement, et pour améliorer l'estimation de l'étalon bronze.
- Comment mieux caractériser les performances d'un même algorithme sur une base de test de plusieurs patients ? Comment prendre en compte une base de test hétérogène (contenant des données avec des niveaux de bruits variables, voire des méthodes

d'acquisitions variables), et ainsi mesurer une précision "absolue" (et non relative à une classe de données) des algorithmes?

9.1.5 Application VirtualScope

Nous conclurons sur le recalage avec ce qui a guidé ce travail tant théorique qu'expérimental : son application à VirtualScope.

Pour pouvoir régler l'algorithme pour une application donnée, nous avons discuté en détail la signification et le rôle des différents paramètres. Nous avons fait la distinction entre les paramètres d'échelles, réglables manuellement à partir de l'étude empirique des données[41] , et les paramètres invariants, dont nous avons donné des valeurs types. Nous avons enfin présenté l'ébauche d'une méthode automatique permettant d'ajuster finement tous ces paramètres en optimisant les performances mesurées par le banc de test.

Ainsi, nous avons pu fournir un algorithme de recalage répondant au cahier des charges de VirtualScope : une précision de l'ordre de $0.2mm$, un temps de calcul de quelques secondes, et une robustesse sur-dimensionnée (une tolérance sur la transformation initiale de l'ordre de $8mm$, sachant que la précision de celle-ci est de l'ordre de $2mm$). Enfin, pour garantir une parfaite sécurité, nous lui avons adjoint un système permettant de rejeter les éventuels mauvais recalages.

Il reste à savoir si ces travaux permettront de s'attaquer vraiment facilement à une nouvelle application : les réglages empiriques et les valeurs-type seront-ils corrects et l'optimisation efficace? Faudra-t-il les revoir, ou même changer l'algorithme lui-même? Les nombreuses applications que l'on peut envisager apporteront certainement leur lot de surprises... elles obligeront sûrement à changer un peu de point de vue, comme nous l'avons fait avec notre approche purement statistique. La tâche ne sera pas si facile qu'il n'y paraît, mais elle sera certainement très instructive et fructueuse.

9.2 Modélisation probabiliste des surfaces

La deuxième partie, consacrée à la modélisation statistique d'éléments de surfaces échantillonnées bruitées, essaye de regrouper différents travaux dans un même édifice, qui reste très théorique et incomplet. Certes, nous confrontons souvent nos idées aux données réelles, pour vérifier que notre édifice reste solide et adapté à ces données. Et force est de constater que l'application au recalage, bien que partielle, apporte des résultats prometteurs. Mais, contrairement à nos attentes, nous n'avons que superficiellement abordé les nombreuses autres applications possibles de ces travaux, et nous sommes contenté d'utiliser des briques rudimentaires ou même parfois approximatives pour mieux nous

41. via la variance mesurée par le recalage dans la première partie, via l'ajustement du modèle théorique sur la loi empirique dans la deuxième partie

concentrer sur l'architecture de l'édifice. Aussi les réponses manquantes et les questions nouvelles (notamment celle de la validation de toutes ces techniques) sont innombrables et touchent beaucoup de domaines. Les conclusions et perspectives que nous présentons ici ne sont certainement pas exhaustives.

9.2.1 Statistiques sur les normales

La première brique de notre édifice est formée des outils statistiques que l'on utilise pour les normales aléatoires.

Nous présentons les outils rigoureux puis des outils approchés (qui s'apparentent en-fait à des M-Estimateurs, et le rapprochement serait intéressant à faire) qui s'avèrent robustes et surtout plus pratiques. Nous justifions ainsi certaines méthodes utilisées couramment (notamment dans notre recalage) pour travailler avec des normales ou des rotations. Nous nous focalisons finalement sur l'une d'elle qui, via les tenseurs, permet de manipuler efficacement la moyenne et la covariance d'une normale aléatoire. L'exemple de l'interpolation des normales illustre son potentiel.

Aussi, nous avons choisi cette méthode des tenseurs pour construire le reste de notre édifice. Mais on pourrait très bien la remplacer, par exemple, par les outils rigoureux. Ceci serait même souhaitable dans le cas où peuvent apparaître des lois bi-modales (au croisement de deux surfaces par exemple), car les tenseurs sont incapables de les traiter.

9.2.2 Modèles de surfaces échantillonnées aléatoires

La construction de notre édifice a continué avec l'édification des modèles de surfaces directement inspirée des notions de saillance et de votes de tenseurs. Nous les avons définis comme devant représenter la probabilité de mesurer un point semi-orienté de la courbe ou la surface connaissant un (et un seul) autre point semi-orienté de celle-ci. Nous avons montré comment obtenir un modèle de cette loi de probabilité, à partir de trois éléments :

- Une loi sur la forme de la courbe ou surface, qui contrôle la façon dont les points peuvent se disperser autour du plan tangent lorsqu'ils s'éloignent du point connu.
- Une loi sur l'échantillonnage, qui indique l'éloignement au point connu.
- Une loi sur le bruit de mesure.

Nous avons donné les exemples les plus évidents pour chacun de ces éléments, et calculé approximativement mais explicitement le modèle théorique résultant de leur combinaison la plus simple (sous hypothèse d'indépendance).

Mais là aussi, bien que la confrontation avec les lois empiriques soit satisfaisante, il reste beaucoup de choses à revoir (comme les approximations dans les calculs, la dérivation des modèles en 3D) et de nouvelles lois à essayer.

9.2.3 Applications

Nous avons ensuite réfléchi aux différentes utilisations possibles de nos modèles de surfaces échantillonnées bruitées. Elles concernent potentiellement toutes les applications où interviennent des données bruitées de courbes ou surfaces échantillonnées, que ce soit par le biais d'images 2D ou 3D (réhaussage, segmentation) ou de nuages de points (nettoyage, calcul des normales et autres informations différentielles[42], complétion, reconstruction, rééchantillonnage, décimation). Nous avons montré sur des exemples la probable marche à suivre : ajuster le modèle aux données, comprendre comment intégrer le vote de plusieurs points (i.e. calculer la probabilité d'un point connaissant plusieurs points), en déduire une probabilité ou une vraisemblance adaptée au problème, et construire un algorithme capable de la maximiser. Reste à le faire effectivement. On pourra commencer par revoir, aux vues de notre travail, les méthodes qui nous ont inspiré, puis chercher de nouvelles applications.

Nous en avons trouvé une première, qui fait élégamment la jonction entre les deux parties de cette thèse en utilisant nos modèles au sein de l'ICP/EM. Et elle s'est d'ores et déjà révélée féconde, combinant succès théoriques (elle justifie et généralise les techniques du chapitre 5) et pratiques (les premières expériences montrent qu'elle améliore légèrement la précision et sa prédiction et diminue sensiblement le nombre d'itérations). Mais notre implémentation n'est encore que partielle, et une étude plus poussée s'impose pour comprendre les véritables capacités de notre approche.

9.2.4 Aspects multi-échelles

Dans cette deuxième partie, nous avons parlé d'échelles (les différentes variances qui interviennent dans nos modèles : celle du modèle d'échantillonnages, du modèle de forme, et celles du bruit), mais pas d'approche multi-échelles. Il ne serait pourtant pas surprenant qu'il se passe des choses intéressantes lorsqu'on fait varier certaines de ces échelles, comme nous l'avons fait dans l'ICP/EM.

Il faut en particulier se poser les question suivantes :

- A quelles paramètres appliquer le schéma multi-échelle : à la variance des points, c'est évident, mais quid de la variance des normales et des autres paramètres (paramètre de forme, échelle d'échantillonnage) ?
- Retrouve-t-on des méthodes connues (de géométrie algorithmique par exemple) pour des échelles de bruits de mesure négligeables (comme l'ICP avec l'ICP/EM, voir annexe B) ? Inversement, peut-on incorporer un modèle de bruit dans ces méthodes connues ?

42. qui pourrait permettre de calculer des invariants et les utiliser dans le recalage

– Que se passe-t-il lorsque l'on tend vers une échelle d'échantillonnage nulle, i.e. vers le continu ?
– Comment mettre au point une approche multi-résolution sur les surfaces échantillonnées, qui explique rigoureusement comment lisser ou décimer au mieux une surface en respectant l'information significative à une échelle donnée ?

9.3 Épilogue : validation clinique de VirtualScope

La validation clinique de VirtualScope est en cours. Cette section sera rédigée lorsque les résultats seront connus.

9.4 Le mot de la fin

Annexe A

Estimation non-biaisée des écart-types pour l'ICP et l'ICP/EM

A.1 Utilisation des valeurs des résultats des recalages : Principe

Nous avons déjà expliqué, dans la section 3.2.1, que la valeur du critère sous sa forme la plus simple $C(T,A) = \frac{1}{\sum_{ij} A_{ij}} . \sum_{ij} A_{ij}.d^2(T \star s_i,m_j)$ était, pour un bon recalage, une estimation statistique de la variance réelle des erreurs. Les différentes modification que nous avons depuis apporté doivent cependant être prises en compte.

Nous ferons les calculs pour un bruit homogène isotrope, bien qu'il ne soit pas difficile de généraliser. Nous noterons σ_r^2 la valeur réelle de la variance du bruit, σ_e^2 la valeur utilisée par les algorithmes (i.e. l'échelle) et σ_m^2 l'estimation statistique de la variance *totale* obtenue en sortie de l'algorithme :

$$\sigma_m^2 = \frac{1}{\sum_{ij} A_{ij}} . \sum_{ij} A_{ij}.d^2(T \star s_i,m_j)$$

Nous supposerons dans la suite que le recalage est bon. En effet, s'il ne l'est pas, σ_m^2 augmente. De plus, nous verrons que la fonction qui nous permet d'estimer σ_r^2 en fonction de σ_m^2 pour σ_e^2 donnée est croissante. On aura donc tendance à surestimer σ_r^2 si le recalage est moins bon, ce qui n'est, comme nous le verrons dans les résultats d'expérience, pas gênant.

A.2 Prise en compte des points aberrants

Nous avons introduit, pour rejeter les faux positifs, une distance maximale D_{max} et des pseudo appariements (D sera ici la distance quadratique : $D(x,y) = \|x-y\|^2$). Ceux-ci vont modifier la valeur mesurée de σ_m^2. C'est cette modification que nous allons étudier. Nous pourrions d'ailleurs calculer deux valeurs mesurées, une en prenant en compte les pseudo appariements du type point aberrant, l'autre sans les prendre en compte. Nous ne garderons que la seconde, forcément moins sensible... aux points aberrants !

$$\sigma_m^2 = \frac{1}{\sum_{ij \neq out} A_{ij}} . \sum_{ij \neq out} A_{ij} . D(e_{ij}, 0)$$

Par conséquent, σ_m^2 est une estimation statistique de $D(\mathbf{e},0)$ sachant que $D(\mathbf{e},0) < D_{max}^2$. On a donc :

$$\sigma_{m\,app}^2 \sim E\left[D(\mathbf{e},0) | D(\mathbf{e},0) < D_{max}\right] = \frac{\int_{\{x/D(x,0)<D_{max}\}} \left[D(x,0)\right] . \exp(-\frac{D(x,0)}{2.\sigma_r^2}).dx}{\int_{\{x/D(x,0)<D_{max}\}} . \exp(-\frac{D(x,0)}{2.\sigma_r^2}).dx}$$

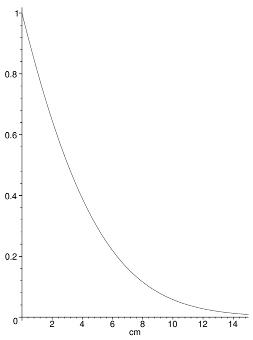
qui peut se ré écrire après manipulation des intégrales :

$$\sigma_{m\,app}^2 \sim \sigma_r^2.d.(1 - \delta(D_{max}/\sigma_r^2)) \qquad \delta(A) = \frac{4}{\mathcal{D}} \frac{(\frac{A}{2})^{(\frac{\mathcal{D}}{2})} . \exp(-\frac{A}{2})}{\int_{u=0}^{A} . (\frac{u}{2})^{(\frac{\mathcal{D}}{2}-1)} . \exp(-\frac{u}{2}).dx} \tag{A.1}$$

Nous allons étudier de plus près la fonction corrective $\delta(A)$. On peux constater qu'il s'agit de la dérivé du log du dénominateur, qui n'est autre que la fonction de répartition d'un χ^2 à $\mathcal{D}$ degrés de liberté :

$$\delta(A) = \frac{4}{\mathcal{D}} \frac{d \log(P(\chi_{\mathcal{D}}^2 < A))}{dA}$$

Il s'agit d'une fonction décroissante, que nous avons ici tracée en dimension 3 :



Nous pourrons donc approximer σ_r^2 en négligeant δ, ce qui ne sera valable que si D_{max}/σ_r^2 est suffisamment élevé :

$$\sigma_r^2 \sim \frac{\sigma_{m\,app}^2}{\mathcal{D}}$$

dans le cas contraire, il faudra résoudre l'équation A.1 pour connaître σ_r^2.

A.3 Distance de Mahalanobis

Les résultats précédents sont faciles à adapter lorsqu'on utilise la distance de Mahalanobis. Il suffit de ne pas confondre ce qui dépend de la variance σ_e^2 utilisé pour la calculer, et ce qui dépend de σ_r^2, valeur réelle de la variance.

La distance de Mahalanobis de l'algorithme peut se définir, à partir de la distance classique par :

$$\mu_e^2(x,y) = \frac{\|x-y\|^2}{\sigma_e^2}$$

On distingue donc l'estimation statistique σ_m^2 de la distance moyenne entre points, et l'estimation statistique $(\mu_e^2)_m$ de la distance de Mahalanobis μ_e^2 moyenne entre deux points :

$$\sigma_m^2 = \frac{1}{\sum_{ij\neq out} A_{ij}}.\sum_{ij\neq out} A_{ij}.\|e_{ij}\|^2$$

$$(\mu_e^2)_m = \frac{1}{\sum_{ij\neq out} A_{ij}}.\sum_{ij\neq out} A_{ij}.\mu^2(e_{ij},0) = \frac{\sigma_m^2}{\sigma_e^2}$$

La distance maximale utilisée est de la forme $D_{max} = \sigma_e^2.\mu_{max}^2$.

Le raisonnement est alors le même que dans le cas précédent et on obtient :

$$\sigma_m^2 \sim \sigma_r^2.\mathcal{D}.(1-\delta(\mu_{max}^2\frac{\sigma_e^2}{\sigma_r^2}))$$

$$(\mu_e^2)_m \sim \frac{\sigma_r^2}{\sigma_e^2}.\mathcal{D}.(1-\delta(\mu_{max}^2\frac{\sigma_e^2}{\sigma_r^2}))$$

A.4 EM

L'EM introduit un biais supplémentaire du fait de la présence d'une gaussienne dans les termes de pondération. Ainsi, notre estimation statistique de la distance de Mahalanobis est :

$$(\mu_e^2)_m = \frac{\sum_{ij\neq out}\overline{A_{ij}}.\mu_e^2(e_{ij},0)}{\sum_{ij\neq out}\overline{A_{ij}}} = \frac{\sum_{ij\neq out}\frac{\exp(-\mu_e^2(e_{ij},0)/2)}{\sum_k \exp(-\mu_e^2(e_{ik},0)/2)}.\mu_e^2(e_{ij},0)}{\sum_{ij\neq out}\frac{\exp(-\mu_e^2(e_{ij},0)/2)}{\sum_k \exp(-\mu_e^2(e_{ik},0)/2)}}$$

Nous allons approximer grossièrement ce terme : nous allons tout d'abord considérer que les contributions des termes $\sum_k \exp(-\mu_e^2(e_{ij},0)/2)$ se compensent :

$$(\mu_e^2)_m \sim \frac{\frac{1}{n_S n_M}\sum_{ij\neq out}\exp(-\mu_e^2(e_{ij},0)/2).\mu_e^2(e_{ij},0)}{\frac{1}{n_S n_M}\sum_{ij\neq out}\exp(-\mu_e^2(e_{ij},0)/2)}$$

Le numérateur est alors une estimation statistique de l'espérance de $\exp(-\mu_e^2(\mathbf{e})/2).\mu_e^2(\mathbf{e})$, et le terme au dénominateur une estimation du terme

$\exp(-\mu_e^2(\mathbf{e})/2)$. On a donc :

$$\mu^2_{m\,app} \sim \frac{E\left[\exp(-\mu_e^2(\mathbf{e})/2).\mu_e^2(\mathbf{e})|\mu_e^2(\mathbf{e}) < \mu^2_{max}\right]}{E\left[\exp(-\mu_e^2(\mathbf{e})/2)|\mu_e^2(\mathbf{e}) < \mu^2_{max}\right]} = \frac{\int_{\{x/\mu_e^2(x)<\mu^2_{max}\}} \mu_e^2(x).\exp(-\frac{\mu_e^2(x,0)}{2}).\exp(-\frac{\mu_e^2(x,0).\sigma_e^2}{2.\sigma_r^2}).dx}{\int_{\{x/\mu_e^2(x)<\mu^2_{max}\}} \exp(-\frac{\mu_e^2(x,0)}{2}).\exp(-\frac{\mu_e^2(x,0).\sigma_e^2}{2.\sigma_r^2}).dx}$$

Les calculs sont alors analogues au cas précédents, et ont pour résultat :

$$\mu^2_{m\,app} \sim \frac{\frac{\sigma_r^2}{\sigma_e^2}}{1+\frac{\sigma_r^2}{\sigma_e^2}}.\mathcal{D}.(1-\delta(\mu^2_{max}(1+\frac{\sigma_e^2}{\sigma_r^2})))$$

Tout ce comporte donc comme si la variance réelle de la variable avait été remplacée de la manière suivante :

$$\frac{1}{\sigma_r^2} \to \frac{1}{\sigma_r^2} + \frac{1}{\sigma_e^2} \quad soit \quad \sigma_r^2 \to \frac{\sigma_r^2}{1+\frac{\sigma_r^2}{\sigma_e^2}}$$

Il est donc inutile de refaire les calculs, et on obtient :

$$(\mu_e^2)_m \sim \frac{\frac{\sigma_r^2}{\sigma_e^2}}{1+\frac{\sigma_r^2}{\sigma_e^2}}.\mathcal{D}.(1-\delta(\mu^2_{max}(1+\frac{\sigma_e^2}{\sigma_r^2})))$$

On peut ici borner δ. En effet, elle est décroissante, et donc :

$$\delta(\mu^2_{max}(1+\frac{\sigma_e^2}{\sigma_r^2})) < \delta(\mu^2_{max})$$

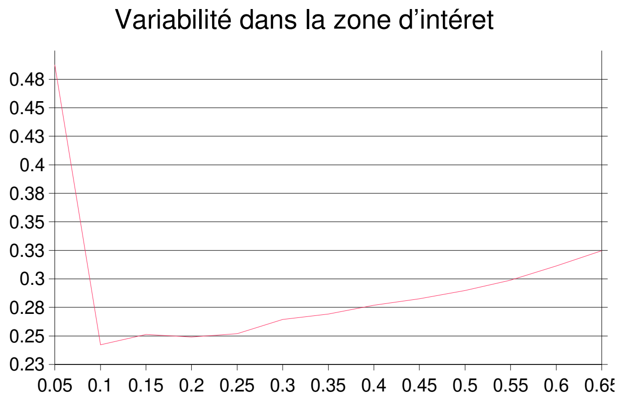
Pour μ^2_{max} suffisamment important, on peux donc négliger δ sans problème, et retrouver ainsi la valeur de σ_r^2 à l'aide de la valeur de σ_m^2 et $(\mu_e^2)_m$:

$$\sigma_r^2 = \frac{\sigma_m^2}{\mathcal{D} - (\mu_e^2)_m}$$

A.5 Résultats expérimentaux

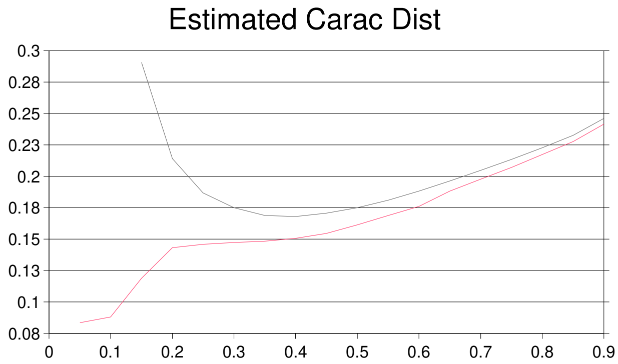
Nous avons réalisé, pour chaque Picza disponible, un série d'expérience de précisions sur le recalage avec un nuage de points per-opératoires (voir 6.3.2), en faisant varier l'échelle de l'algorithme (i.e. σ_e^2).

Voici les résultats pour le Picza à 0.2mm, avec, en abscisse, $\sqrt{\sigma_e^2}$, l'échelle utilisé par les algorithmes. Tout d'abord, la précision du recalage, exprimée comme l'écart-type de la TRE (les cibles étant réparties sur l'ensemble de la mâchoire) :

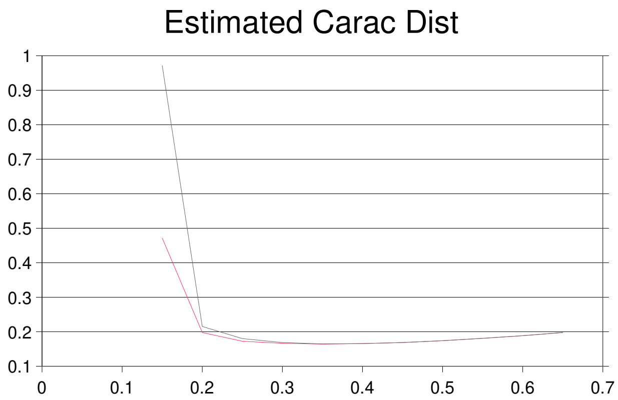


L'algorithme semble optimal dans la zone située pour une variance de l'ordre de 0.15 mm à 0.2mm, qui correspond effectivement, comme nous allons le voir, à la variance des points.

Voyons maintenant l'estimation des variances (en fait de sa racine carré, l'écart-type) correspondante; la courbe inférieure représente ce calcul, la courbe supérieure prend aussi en compte les points aberrants :



L'estimation des variance est stable dans la zone située entre 0.2mm et 0.6mm. L'écart type correspondant est de l'ordre de 0.175mm. L'estimation de la variance n'est donc stable que lorsque l'algorithme utilise une variance surestimée. Nous avons voulu savoir si ce défaut venait de δ. Pour ça, nous avons considérablement augmenté μ_{max}^2, en le fixant à 30, si bien que $\delta < 10^{-5}$ (au lieu de $\delta < 0.2$ pour $\mu_{max}^2 = 6.25$). L'estimation des variances possède alors la forme suivante :



L'estimation est alors encore plus stable dans la zone située entre 0.2mm et 0.6mm, mais reste hautement instable pour une variance estimée de l'ordre de la variance véritable. Ce résultat provient probablement de la baisse conséquente du nombre d'appariements pertinents, et de leur répartition quasi uniforme, et nous empêche d'utiliser directement l'estimation ici décrite pour la mise à jour de la variance utilisée par l'algorithme. Elle permet par contre d'estimer très précisément les variances pour chaque type de recalage. Ainsi, les variances estimées pour le recalage Picza/Série de pointés, avec un μ^2_{max} intermédiaire égal à 11.34 (test du χ^2 à 99%), sont les suivantes :

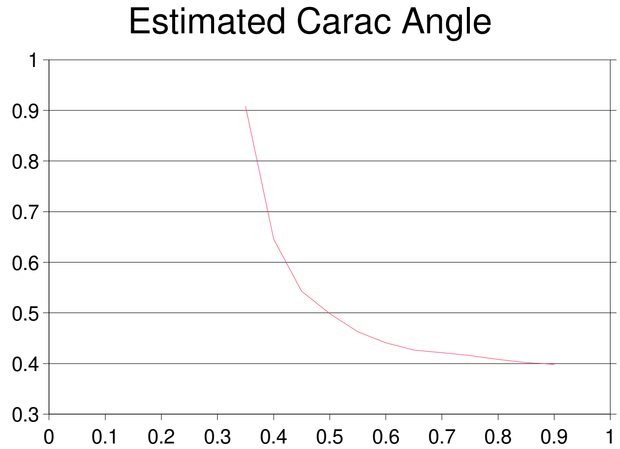
Résolution Picza (mm)	0.2	0.3	0.4	0.5	0.75	1
Distance caractéristique estimée (mm)	0.17	0.2	0.24	0.25	0.38	0.52
Précision (mm)	0.24	0.3	0.4	0.31	0.46	0.5

On a ainsi amélioré la précision de l'algorithme, en réglant correctement un de ses paramètres les plus importants.

A.6 Application aux normales

Bien que les formules développées ne soient plus tout à fait exactes pour les normales, nous avons essayé de calculer de la même façon les angles caractéristiques dans nos séries de pointés orientés.

On a obtenu les graphiques suivants :



Ces expériences montrent que le bruit sur les normales est de l'ordre de 25 degrés, ce qui semble réaliste, puisque les normales sont relativement précises sur le Picza, mais sont calculées à l'aide de l'orientation de la sonde pour l'imagerie per-opératoire.

Annexe B

Comportements de l'ICP/EM gaussien pour des variances extrêmes

B.1 EM limit for a low variance

Here is a proof that the EM algorithm behaves like standard ICP for a low variance. Previous proof can be found in [Wells, 1997].

We can rewrite Eq. 4.8 as:

$$\overline{(A_T)_{ij}} = \frac{1}{1 + \sum_{k \neq j} \frac{\overline{\pi_{ik}}}{\overline{\pi_{ij}}}.r_{ijk}} \quad \text{with} \quad r_{ijk} = \exp\left(\frac{d^2(s_i,T \star m_j) - d^2(s_i,T \star m_k)}{2.\sigma^2}\right)$$

The limit of this term for a low variance is obviously:

$$\lim_{\sigma^2 \to 0} r_{ijk} = \begin{cases} 0 & if \quad d^2(s_i,T \star m_j) < d^2(s_i,T \star m_k) \\ +\infty & if \quad d^2(s_i,T \star m_j) > d^2(s_i,T \star m_k) \end{cases}$$

Then, if $T \star m_j$ is the closest point (w.r.t. s_i), all r_{ijk} terms have a null limit, so that the limit of $\overline{(A_T)_{ij}}$ is 1. Otherwise, there exists a r_{ijk} with an infinite limit, so that the limit of $\overline{(A_T)_{ij}}$ is 0. Therefore, EM algorithm behaves as standard ICP.

B.2 EM limit for a high variance

In this section, we show that the EM algorithm simply aligns the barycentres and the inertia tensors (weighted by the a-priori probabilities of points) for a high variance.

Let $\alpha = \frac{1}{2.\sigma^2}$ be the variable that tends toward 0 and $D_{ij} = d^2(s_i,T \star m_j) =$

$\|s_i - (R.m_j + t)\|^2$. The criterion 4.6 can be rewritten as:

$$C(T) = -\sum_i \log\left(\sum_j \overline{\pi_{ij}}.p(s_i|m_j,T)\right) = -\sum_i \log\left(\sum_j \overline{\pi_{ij}}.\exp(-\alpha.D_{ij})\right)$$

B.2.1 Translation part

The first order Taylor expansion of this criterion w.r.t. α is:

$$\begin{aligned} C(T) &= -\textstyle\sum_i \log\left(\sum_j \overline{\pi_{ij}}.\exp(-\alpha.D_{ij})\right) \\ &= -\textstyle\sum_i \log\left(\sum_j \overline{\pi_{ij}}.(1-\alpha.D_{ij} + O(\alpha^2))\right) \\ &= -\textstyle\sum_i \log\left(1-\alpha.\sum_j \overline{\pi_{ij}}.D_{ij} + O(\alpha^2)\right) \\ &= \alpha.\textstyle\sum_{ij} \overline{\pi_{ij}}.D_{ij} + O(\alpha^2) \end{aligned}$$

Then, for a low α (e.g. for a high variance), minimising the criterion is equivalent to minimising $\sum_{ij}\overline{\pi_{ij}}.D_{ij} = \sum_{ij}\overline{\pi_{ij}}.d^2(s_i,T\star m_j) = \sum_{ij}\overline{\pi_{ij}}.\|s_i - (R.m_j+t)\|^2$. As usual, the optimal translation aligns the weighted barycentres:

$$t = \frac{\sum_{ij}\overline{\pi_{ij}}.s_i}{\sum_{ij}\overline{\pi_{ij}}} - R\frac{\sum_{ij}\overline{\pi_{ij}}.m_j}{\sum_{ij}\overline{\pi_{ij}}} = \overline{s_i} - R.\overline{m_j}$$

B.2.2 Rotation part

Reintroducing the translation into the criterion and using barycentric coordinates (each s_i is replaced by $s_i - \overline{s}$, each m_j is replaced by $m_j - \overline{m}$) leads to the following expression: $D_{ij} = \|s_i - R.m_j\|^2$. Thus, we are left with the minimisation of:

$$\sum_{ij}\overline{\pi_{ij}}.D_{ij} = \sum_{ij}\overline{\pi_{ij}}.\|s_i - (R.m_j)\|^2 = \sum_{ij}\overline{\pi_{ij}}.(\|s_i\|^2 + \|(R.m_j)\|^2 - 2.s_i^t.R.m_j)$$

As $\|s_i\|^2$ and $\|R.m_j\|^2 = \|m_j\|^2$ are constant, we can equivalently maximise $\sum_{ij}\overline{\pi_{ij}}.(s_i^t.R.m_j) = \sum_{ij}\overline{\pi_{ij}}.Tr(R.m_j.s_i^t) = Tr(R.K^t)$ with $K = \sum_{ij}\overline{\pi_{ij}}.s_i.m_j^t$.

However, when $\overline{\pi_{ij}}$is separable (i.e. $\overline{\pi_{ij}} = \overline{\pi_i}.\overline{\pi_j}$, which is the case when using uniform priors, as in our application), K is equal to $(\sum_{ij}\overline{\pi_{ij}}).\overline{s}.\overline{m}^t$, thus null since we are now in barycentric coordinates. Consequently, the first order expansion of our criterion is not sufficient to determine the rotation. In this case, we need to write the second order Taylor expansion:

$$\begin{aligned}
C(T) &= -\sum_i \log\left(\sum_j \overline{\pi_{ij}}.(1-\alpha.D_{ij}+\tfrac{1}{2}\alpha^2 D_{ij}^2 + O(\alpha^3))\right)\\
&= -\sum_i \log\left(1-\alpha.Cte+\tfrac{1}{2}\alpha^2 \sum_j \overline{\pi_{ij}}.D_{ij}^2 + O(\alpha^3)\right)\\
&= -\sum_i \left(\alpha.Cte+\tfrac{1}{2}\alpha^2 (\sum_j \overline{\pi_{ij}}.D_{ij}^2 - Cte^2)\right) + O(\alpha^3)
\end{aligned}$$

We finally have to maximise:

$$\begin{aligned}
\sum_{ij} \overline{\pi_{ij}}.D_{ij}^2 &= \sum_{ij} \overline{\pi_{ij}}.\left(\|s_i-(R.m_j)\|^2\right)^2\\
&= \sum_{ij} \overline{\pi_{ij}}.\left(\|s_i\|^2+\|m_j\|^2-2.s_i^t.R.m_j\right)^2\\
&= \sum_{ij} \overline{\pi_{ij}}.\left(\|s_i\|^2+\|m_j\|^2\right)^2 - 4.\sum_{ij} \overline{\pi_{ij}}.\|s_i\|^2.s_i^t.R.m_j\\
&\quad -4.\sum_{ij} \overline{\pi_{ij}}.\|m_j\|^2.s_i^t.R.m_j + 4.\sum_{ij}(s_i^t.R.m_j)^2
\end{aligned}$$

The first term is constant, the second term equals to

$$-4.\sum_i \overline{\pi_i}.\|s_i\|^2.s_i^t.R.\left(\sum_j \overline{\pi_j} m_j\right) = -4.\sum_i \overline{\pi_i}.\|s_i\|^2.s_i^t.R.\left(\sum_j \overline{\pi_j}\right).\overline{m}$$

and is thus null since we are in barycentric coordinates, so is the third term. Finally, we are left with the maximisation of the last term:

$$\begin{aligned}
\sum_{ij} \overline{\pi_{ij}}(s_i^t.R.m_j)^2 &= \sum_{ij} \overline{\pi_{ij}}(s_i^t.R.m_j.m_j^t.R^t.s_i)\\
&= \sum_i \overline{\pi_i}(s_i^t.R.(\sum_j \overline{\pi_j}.m_j.m_j^t).R^t.s_i)\\
&= Tr(R.(\sum_j \overline{\pi_j}.m_j.m_j^t).R^t.(\sum_i \overline{\pi_i}.s_i.s_i^t))\\
&= Tr(R.T_m.R^t.T_s)
\end{aligned}$$

where T_m and T_s are the scene and model inertia tensors. To account for the orthonormal constraints of the rotation matrix, we have to add an additional term in the Lagrangian:

$$\Lambda(R) = \frac{1}{2}Tr(R.T_m.R^t.T_s) - \frac{1}{2}Tr(L(R.R^t - Id))$$

where L is a symmetric matrix of Lagrange multipliers, to take into account constraint on R, which must be orthogonal. Expanding this Lagrangian gives:

$$\Lambda(R+\delta R) = \Lambda(R) + \frac{1}{2}Tr(\delta R.T_m.R^t.T_s + Tr(R.T_m.\delta R^t.T_s) - \frac{1}{2}Tr(L.R.\delta R^t + L.\delta R.R^t)$$

By definition of the derivative, we have $\Lambda(R+\delta R) = \Lambda(R) + Tr\left(\frac{\partial \Lambda}{\partial R}\delta R\right)$. Thus, the optimum rotations are characterised by:

$$\frac{\partial \Lambda}{\partial R} = 0 = T_s.R.T_m - L.R \tag{B.1}$$

Let $T_m = U.D_m.U^t$ and $T_s = V.D_s.V^t$ be diagonalisations of the data inertia tensors and $Q = V^t.R.U$, $M = V^t.L.V$ (remember that T_m, T_s, L and M are symmetric, U, V, R, Q are orthogonal, and D_m and D_s are diagonal). Equation B.1 becomes: $D_s.Q.D_m = Q.M$. Assuming that the eigenvalues d_{m_i} of D_m are strictly positive, we can rewrite that as: $Q^t.D_s.Q = M.D_m^{-1}$. Thus, the matrix $M.D_m^{-1}$ is obviously symmetric: $M.D_m^{-1} = D_m^{-1}.M$. Since M and D_m commute, they can be diagonalised in the same coordinate system, here the canonical one: M is thus a diagonal matrix D verifying $Q^t.D_s.Q = D.D_m^{-1}$.

Finally, we can conclude that Q is a rotation that exchanges rows or columns (i.e. a rotation of 90 or 180 degrees) and we have $d_i = \pm d_{s_i}.d_{m_{\pi(i)}}$ where $\pi(i)$ is a permutation of the indices. Reporting the values of L and R in the criterion gives $2\Lambda_{optimal} = Tr(D) = \sum_i \pm d_{s_i}.d_{m_{\pi(i)}}$. Assuming that the eigenvalues d_{s_i} and d_{m_i} of the inertia tensors are ordered by *strictly decreasing* values, the maximal criterion is obtained for $\pi(i) = i$ and a positive sign of d_i (thus $Q = Id$ and $R = V.U^t$) with value $\sum_i d_{s_i}.d_{m_i}$. if we are looking for a proper rotation and if the determinant of $V.U^t$ is -1, the maximum criterion is obtained (in 3D) by $d_3 = -d_{s_3}.d_{m_3}$ (i.e. for $R = V.Diag(1,1,-1).U^t$).

Annexe C

Quelques rappels sur les statistiques

C.1 Régression linéaire

La démarche que nous suivons dans cette thèse pour justifier l'ICP et calculer son incertitude est en-fait la transposition d'un raisonnement très général dans le monde des statistiques. Il est utilisé lorsqu'il s'agit de calculer des paramètres θ qui "expliquent" le mieux à une série de données $\chi = \{x_i\}$. Les grandes étapes de ce raisonnement sont les suivantes :

- Choisir une approche par maximisation de la vraisemblance plutôt que par maximisation de la probabilité a posteriori des données.
- Calculer la vraisemblance des données, i.e. la probabilité des données $p(\chi/\theta)$ en fonction des paramètres. On suppose en général pour cela que les données sont indépendantes entre elles, et suivent toutes une même loi gaussienne.
- Maximiser cette vraisemblance, ou plutôt minimiser le critère défini comme l'opposé du logarithme de cette vraisemblance. On retrouve alors en général une optimisation aux moindres carrés.
- Estimer la précision des paramètres ainsi estimés par propagation de l'incertitude.
- Calculer la limite de Cramer-Rao du problème (i.e. la meilleure précision que l'on puisse obtenir quelle que soit l'approche utilisée), établir l'égalité entre cette limite et la précision calculée précédemment, et montrer ainsi que l'approche par maximisation de la vraisemblance est la meilleure.

Nous allons détailler ces différentes étapes, en les illustrant sur un cas très simple, celui du calcul de la moyenne.

Le problème est ici le suivant : nous disposons d'une série de données $\chi = \{x_i\}$ qui proviennent d'une loi gaussienne isotrope de variance σ^2 connue et centrée autour d'une valeur μ inconnue. Le paramètre que nous allons chercher à estimer est cette moyenne μ.

C.1.1 Approche MAP et MV

La première chose à faire pour aborder notre problème est de choisir entre :

- une approche par maximisation de la probabilité a-posteriori $p(\theta/\chi)$ (approche MAP), i.e. de la probabilité des paramètres θ en fonction des données χ.
- une approche par maximisation de la vraisemblance (approche MV), i.e. de la probabilité des données χ en fonction des paramètres θ.

La seconde approche peut sembler beaucoup moins naturelle, puisqu'on n'y cherche pas directement les paramètres les plus probables. Elle est cependant plus pratique car, comme nous le verrons à la section suivante, la vraisemblance peut être calculée directement.

Pour calculer la probabilité a-posteriori, il faut en général utiliser la règle de Bayes :

$$p(\theta/\chi) = \frac{p(\chi/\theta)p(\theta)}{p(\chi)}$$

On voit que la probabilité a-posteriori est égale à la vraisemblance multipliée par un terme correctif faisant intervenir la probabilité a-priori (i.e. ne connaissant rien d'autre) des données $p(\chi)$ et des paramètres $p(\theta)$. Deux cas se présentent alors :

- Si on a aucune connaissance a-priori sur les données ou les paramètres : les probabilités a-priori des données $p(\chi)$ et des paramètres $p(\theta)$ sont constantes, la probabilité a-posteriori et la vraisemblance sont donc proportionnelles. Les approches MAP et MV sont équivalentes.
- Dans le cas contraire, l'approche MV pourra toujours être appliquée, mais ne permettra pas de prendre en compte la connaissance a-priori sur les données ou les paramètres. L'approche MAP le permettra, mais sera en général plus difficile à appliquer ??

C.1.2 Calcul et maximisation de la vraisemblance

Il nous faut donc ici calculer la vraisemblance, i.e. la probabilité des données connaissant les paramètres. C'est ici que nous allons devoir modéliser le processus qui a mené à la mesure de ces données. Nous ferons deux hypothèses :

- Que les données sont indépendantes entre elles.
- Que chaque donnée x_i suit une loi gaussienne de moyenne μ et de variance σ^2.

L'hypothèse d'indépendance permet alors d'écrire la probabilité des données $\chi = \{x_i\}$ comme le produit des probabilités de chaque donnée x_i :

$$p(\chi/\theta) = \prod_i p(x_i/\theta) = \prod_i \exp(-\frac{d^2(x_i,\mu)}{2\sigma^2})$$

Nous souhaitons maximiser cette vraisemblance. Mais il est strictement équivalent de minimiser le critère défini comme l'opposé de son logarithme :

$$C(\theta) = -\log p(\chi/\theta) = \sum_i \frac{d^2(x_i,\mu)}{2\sigma^2}$$

Comme nous le voyons, nous retombons sur un critère de type moindre carré.

Nous allons minimiser ce critère pour obtenir la valeur estimée de la moyenne, notée $\hat{\mu}$. Pour cela, nous annulons la dérivé :

$$\frac{\partial C}{\partial \hat{\mu}} = \sum_i (x_i - \hat{\mu})/2\sigma^2 = 0 \quad \Longrightarrow \quad \hat{\mu} = E\left[x_i\right]$$

Remarque : estimateur non-biaisé Nous supposerons par la suite que l'estimateur statistique que nous avons ainsi mis au point n'est pas biaisé. Il faut pour cela que l'estimée $\hat{\theta}(x_1,...,x_n)$ soit en moyenne égal à la véritable valeur θ du paramètre que l'on cherche à estimer :

$$E\left[\hat{\theta}\right] = \int_{x_1,\cdots,x_n} \hat{\theta}(x_1,\cdots,x_n).p(x_1|\theta).\cdots.p(x_n|\theta).dx_1.\cdots.dx_n = \theta$$

C.1.3 Calcul d'incertitude

Nous savons maintenant comment estimer à partir des données la valeur du paramètre θ. Dès-lors, on peut se demander quelle est l'incertitude sur cette estimation, i.e. la variabilité de l'estimée $\hat{\theta}$ autour de la bonne valeur θ. En termes probabilistes, il s'agit de calculer la covariance de $\hat{\theta}$ autour de θ. Lorsque l'estimateur n'est pas biaisé, il s'agit de la covariance de $\hat{\theta}$ (sous-entendu autour de sa moyenne). Cette dernière peut-être calculée par propagation de l'incertitude sur les données.

Le principe de ce calcul est expliqué en détail dans le cas du recalage, section 3.9.1. Dans le cas de la moyenne, il permet de retrouver la loi des grands nombres :

$$\sigma_\mu^2 = \frac{1}{n}\sigma^2$$

C.1.4 Limite de Cramer-Rao (optimalité)

Nous venons de calculer la précision théorique de notre estimateur au moindre carré. On peut maintenant se demander s'il s'agit de la meilleure précision que l'on puisse atteindre, i.e. si nous avons choisi le meilleur estimateur. La limite de Cramer-Rao d'un problème d'estimation statistique est précisément la meilleure précision que l'on puisse atteindre avec un estimateur non-biaisé. Nous allons ici présenter son calcul dans le cas de la moyenne.

Soit un estimateur non-biaisé :

$$E\left[\hat{\theta}\right] = \theta$$

on va dériver cette égalité par rapport à θ. On obtient[43] alors :

$$E\left[\sum_i \hat{\theta}.\frac{\mu x_i}{\sigma^2}\right] = Id \tag{C.1}$$

On note maintenant

$$e = \hat{\theta} - \theta$$
$$l = \sum \frac{\mu x_i}{\sigma^2}$$

Le premier vecteur est l'erreur de notre estimateur. Elle est d'espérance nulle[44] et sa covariance $\Sigma_{\theta\theta}$ est la covariance de notre estimateur, qui donne sa précision. Notre deuxième vecteur est lui aussi d'espérance nulle[45] de sorte que $E\left[el^t\right] = E\left[\left(\hat{\theta} - \theta\right) l^t\right] =$

43. On ré-écrit l'espérance dans l'hypothèse d'absence de biais :

$$\int_{x_1,\ldots,x_n} \hat{\theta}(x_1,\cdots,x_n).p(x_1|\theta).\cdots.p(x_n|\theta).dx_1.\cdots.dx_n = \theta$$

que l'on veut dériver par rapport à θ. Pour le terme de droite, pas de problème, on obtient l'identité. Pour le terme de gauche, on raisonne pour un échantillon $(x_1,\ldots,x_n)$ fixé. : l'estimation $\hat{\theta}(x_1,\cdots,x_n)$ reste alors constante, et on a :

$$\frac{d}{d\theta}\hat{\theta}(x_1,\cdots,x_n).p(x_1|\theta).\cdots.p(x_n|\theta) = \sum_i \hat{\theta}(x_1,\cdots,x_n).p(x_1|\theta).\cdots.\frac{dp(x_i|\theta)}{d\theta}.\cdots.p(x_n|\theta)$$

hors $\frac{dp(x_i|\theta)}{d\theta} = \frac{dd^2(x_i,\mu)}{2d\theta}p(x_i|\theta) = p(x_i|\theta).\frac{\mu x_i}{\sigma^2}$ et donc :

$$\frac{d}{d\theta}\hat{\theta}(x_1,\cdots,x_n).p(x_1|\theta).\cdots.p(x_n|\theta) = \sum_i \hat{\theta}(x_1,\cdots,x_n).\frac{\mu x_i}{\sigma^2}.p(x_1|\theta).\cdots.p(x_n|\theta)$$

on intègre maintenant (i.e. on calcule l'espérance) :

$$\frac{d}{d\theta}\int_{x_1,\ldots,x_n} \hat{\theta}(x_1,\cdots,x_n).p(x_1|\theta).\cdots.p(x_n|\theta).dx_1.\cdots.dx_n$$
$$= \int_{x_1,\ldots,x_n} \frac{d}{d\theta}\hat{\theta}(x_1,\cdots,x_n).p(x_1|\theta).\cdots.p(x_n|\theta).dx_1.\cdots.dx_n$$
$$= \int_{x_1,\ldots,x_n} \sum_i \hat{\theta}(x_1,\cdots,x_n).\frac{\mu x_i}{\sigma^2}.p(x_1|\theta).\cdots.p(x_n|\theta).dx_1.\cdots.dx_n$$

et donc :

$$\frac{dE\left[\hat{\theta}\right]}{d\theta} = E\left[\sum_i \hat{\theta}.\frac{\mu x_i}{\sigma^2}\right]$$

44. Toujours parce que notre estimateur n'est pas biaisé.
45. En dérivant $E[1] = 1$ comme dans la note précédente.

$E\left[\left(\hat{\theta}-\theta\right)l^t\right]+E\left[\theta l^t\right]=E\left[\hat{\theta}.l^t\right]$ et donc, en reprenant l'eq. C.1 : $E\left[el^t\right]=Id$. Enfin nous définissons la matrice d'information :

$$J=E\left[ll^t\right]=n.E\left[\frac{\mu x_i}{\sigma^2}.\frac{\mu x_i}{\sigma^2}^t\right]$$

On peut alors calculer la matrice symétrique $\begin{pmatrix} e \\ l \end{pmatrix}\begin{pmatrix} e \\ l \end{pmatrix}^t=\begin{pmatrix} ee^t & el^t \\ le^t & ll^t \end{pmatrix}$, qui est symétrique positive par construction (nous noterons $\succ 0$ pour signifier que la matrice est positive), et calculer son espérance (qui garde les mêmes propriétés) :

$$E\left[\begin{pmatrix} ee^t & el^t \\ le^t & ll^t \end{pmatrix}\right]=\begin{pmatrix} E\left[ee^t\right] & E\left[el^t\right] \\ E\left[le^t\right] & E\left[ll^t\right] \end{pmatrix}=\begin{pmatrix} \Sigma_{\theta\theta} & Id \\ Id & J \end{pmatrix}\succ 0 \tag{C.2}$$

Or si A est définie positive, alors BAB^t l'est aussi (quel que soit B). En prenant A définie dans l'eq. C.2 et $B=\begin{pmatrix} Id & -J^{-1} \\ 0 & J^{-1} \end{pmatrix}$, on obtient :

$$\begin{pmatrix} Id & -J^{-1} \\ 0 & J^{-1} \end{pmatrix}\begin{pmatrix} \Sigma_{\theta\theta} & Id \\ Id & J \end{pmatrix}\begin{pmatrix} Id & 0 \\ -J^{-1} & J^{-1} \end{pmatrix}=\begin{pmatrix} \Sigma_{\theta\theta}-J^{-1} & 0 \\ 0 & J^{-1} \end{pmatrix}\succ 0$$

et donc finalement :

$$\Sigma_{\theta\theta}\succ J^{-1} \tag{C.3}$$

Dans le cas de l'estimation de la moyenne, on a :

$$J=E\left[ll^t\right]=n.E\left[\frac{\mu x_i}{\sigma^2}.\frac{\mu x_i}{\sigma^2}^t\right]=n.\frac{\sigma^2}{\sigma^2.\sigma^2}=\frac{n}{\sigma^2}$$

et donc :

$$\sigma_\mu^2\geq\frac{\sigma^2}{n}$$

C.1.5 Conclusion : optimalité de l'estimateur par maximum de vraisemblance

Nous venons de prouver que la précision d'un estimateur non-biaisé de la moyenne ne pouvait être meilleure que la limite de Cramer-Rao, égale à $\frac{\sigma^2}{n}$. Or il s'agit de la précision prédite pour l'estimateur par maximum de vraisemblance, i.e. l'estimateur aux moindres carrés. Nous pouvons dès-lors affirmer que cet estimateur est optimal en terme de précision.

De manière général, qu'il s'agisse de l'estimation de la moyenne ou du recalage, on peut prouver ainsi que l'estimateur par maximum de vraisemblance est optimal...

C.2 l'EM

L'algorithme EM a été introduction par [Dempster *et al.*, 1977]. Nous allons ici en donner le principe général.

Les problèmes que l'EM va permettre de résoudre sont les problèmes d'estimation de paramètres θ par maximum de vraisemblance faisant intervenir deux catégories de données :

- Les données connues $D = \{d_j\}$: il s'agira des données qui sont disponibles.
- Les données inconnues, dites cachées $C = \{c_j\}$: il s'agit de données qui, si elles avaient été disponibles, auraient permis d'écrire simplement la vraisemblance et de résoudre directement le problème. Ces données seront en général des variables binaires représentant une classification.

La première étape (en général évidente) est donc d'exhiber les variables cachées et d'écrire la vraisemblance connaissant ces variables cachées :

$$p(D|C,\theta)$$

C.2.1 Applications

C.2.1.1 Mixture de gaussiennes

Une mixture de gaussienne est une loi de probabilité définie comme la somme pondéré de plusieurs lois gaussiennes :

$$p(x) = \sum_i \pi_i G(\overline{x}_i,\sigma_i^2) \qquad \pi_i > 0 \quad \sum_i \pi_i = 1$$

on retrouve souvent ces lois lorsqu'on travaille sur une population formée de plusieurs groupes aux comportements différents : chaque gaussienne représente le comportement d'un groupe donné, et les poids π_i correspondent à la proportion de chaque groupe dans la population globale. Ainsi, si on s'intéresse à la taille des humains, il est pertinent de distinguer les femmes des hommes. Lorsqu'on prend une personne au hasard, on a π_{femme} chances de tomber sur une femme, et donc de mesurer une réalisation de la loi de probabilité $G(\overline{x}_{femme},\sigma^2_{femme})$ que suit la taille des femmes. Finalement, nos mesures seront les réalisations de la loi $\pi_{femme}G(\overline{x}_{femme},\sigma^2_{femme}) + \pi_{homme}G(\overline{x}_{homme},\sigma^2_{homme})$.

Le problème auquel peut s'appliquer l'EM est l'estimation de tout ou partie (si certains sont déjà connus) des paramètres de la mixture de gaussienne ($\overline{x}_i$, σ_i^2 et/ou π_i), aux vues d'un certains nombre de réalisation x_j de cette loi. Les données connus sont précisément ces réalisations. On utilisera comme données inconnues le groupe d'origine de chacune de ces réalisations. En-effet, si elles avaient été connues, le calcul des paramètres par maximum de vraisemblance aurait été immédiat.

Ainsi, avec les tailles des humains, on pourra rechercher la taille moyenne des hommes et des femmes à partir d'un sondage dans lequel le sexe n'a pas été noté.

C.2.1.2 Segmentation du cerveau et estimation du biais dans les IRM

Dans le domaine de l'imagerie médicale, l'EM a été utilisé avec succès pour corriger le biais en intensité sur les IRM du cerveau [Van Leemput *et al.*, 1999; Prima, 2001]. Pour simplifier, le problème est le suivant : chaque voxel est issu d'un tissu donné (matière blanche, matière grise ou liquide céphalo-rachidien) et l'intensité suit une loi gaussienne différente pour chaque tissu. De plus l'intensité dans l'IRM est corrompu par un biais spatial dû aux in-homogénéités du champs magnétique, et qui rend par exemple l'image plus claire à gauche qu'à droite.

Les paramètres à estimer sont donc les caractéristiques (moyenne et écart-type des gaussiennes), et les paramètres du biais spatial. Les données connues sont les intensités des voxels. Le calcul des paramètres par maximisation de la vraisemblance aurait été direct si la segmentation (i.e. le tissu d'origine de chaque voxel) était connu, et cette dernière sera donc notre variable cachée.

C.2.2 Principe

Nous avons donc, d'une manière ou d'une autre, reformulé le problème en écrivant la vraisemblance connaissant les données cachées :

$$p(D|C,\theta)$$

Une première approche, évidente, est celle du maximum de vraisemblance pur : on va estimer les données cachées comme les paramètres, en maximisant la vraisemblance. Il est en général impossible d'optimiser conjointement les deux, et il faut donc faire une optimisation alternée, qui exige une première estimation de l'un des deux. Dans la plupart des cas, on utilisera une première estimation des paramètres θ, et l'algorithme est donc le suivant :

Initialisation : Calculer une première estimation des paramètres θ
Répéter

- **Étape E** : Estimer les variables cachées par maximum de vraisemblance : $C = argmax\, p(D|C,\theta)$
- **Étape M** : Ré-estimer les paramètres par maximum de vraisemblance : $\theta = argmax\, p(D|C,\theta)$

Jusqu'à convergence

Il peut cependant sembler déplacé de faire jouer le même rôle aux paramètres et aux variables cachées. D'où l'idée principale de l'EM : comme on ne s'intéresse pas directement aux variables cachées, on ne va pas se contenter d'estimer le jeu de variables cachées le plus probable, mais on va estimer la probabilité $p(C|D,\theta)$ de chaque jeu de variables cachées possible, calculer l'espérance de la vraisemblance $q(D|\theta) = E_{p(C|D,\theta)}\left[p(D|C,\theta)\right]$ et maximiser cette dernière par rapport aux paramètres. Cette approche a l'avantage de prendre en compte d'éventuelles ambiguïtés dans l'estimation des variables cachées.

Ainsi présenté, l'algorithme EM peut sembler impossible à implémenter efficacement car il est trop coûteux de calculer la probabilité de chaque jeu de variable caché C possible. Cette difficulté sera contournée par une petite astuce mathématique qui permet, lorsque les données sont indépendantes, de calculer directement $q(D|\theta)$ sans avoir à passer par chaque C possible (voir section 4.3.2). L'algorithme prend alors la forme suivante :

Initialisation : Calculer une première estimation des paramètres θ

Répéter

- **Étape E** (étape de calcul de l'espérance) : Calculer l'espérance du critère en fonction de la loi de probabilité sur les variables cachées : $q(D|\theta) = E_{p(C|D,\theta)}\left[p(D|C,\theta)\right]$
- **Étape M** (étape de maximisation du critère) : Ré-estimer les paramètres en maximisant le critère $q(D|\theta)$

Jusqu'à convergence

C.2.3 Pour en savoir plus

Nous ne ne détaillerons pas plus l'EM dans le cas général, préférant le détailler sur le cas particulier du recalage. Le lecteur intéressé pourra se reporter à [Couvreur, 1996; Neal and Hinton, 1998] pour une présentation plus détaillé de l'algorithme même et de ses variantes, et à [McLachlan and Krishnan, 1997] pour de nombreux exemples d'applications.

C.3 Démineur

Nous allons ici illustrer l'utilisation des techniques usuelles de statistiques sur une première application originale[46] et ludique, celle de la résolution automatique du démineur, qui illustre parfaitement l'utilisation de l'information, les problèmes d'indépendance entre variable, et les algorithmes de minimisation alternée.

46. Nous n'avons pas fait de recherche bibliographique sur ce sujet, sorte de défi personnel probablement inconsciemment motivé par mon rattachement à l'Ecole des Mines ! Mais la méthode a sûrement déjà été imaginée, pour d'autres applications ou dans un contexte plus général.

CaptureDemineurCas1 not found\protect ! CaptureDemineurCas2 not found\protect !

FIG. C.1 – *Exemples de situations de jeu aux démineur :*
A gauche : la case centrale indique qu'elle est entourée de 3 mines, et il ne reste que 3 cases couvertes alentours. Ces cases sont certainement minées.
A droite : ici, on ne peut trouver de case certaine. On sait par exemple qu'il y a une mine située dans l'une des deux cases marquées d'un point d'interrogation, mais on ne peut préciser laquelle.

C.3.1 Le jeu

Le jeu de démineur se présente ainsi : On commence avec un damier de cases recouvertes. On sait seulement que n mines ont été réparties aléatoirement sur le damier. A chaque coup, on découvre une case. Peuvent alors se passer deux choses : soit il y a une mine dans cette case, et... mieux vaut ne pas y penser. Soit il n'y a pas de mine, et la case découverte révèle un chiffre indiquant le nombre de mines dans les 8 cases voisines (voir figure C.1). Bien-sûr, les premiers coups doivent être joués au hasard, mais les chiffres découverts peuvent ensuite permettre de repérer des cases potentiellement minées ou saines. Deux situations peuvent se présenter :

– Souvent, on peut trouver au moins une case dont la nature est certaine (voir figure C.1 gauche). On peut alors la marquer (s'il s'agit d'une mine) ou la découvrir (si elle est saine) pour faire avancer le jeu sans risque.
– Parfois, aucune case n'est certaine (voir figure C.1 droite). On ne peut donc pas jouer en toute sécurité.

Pour le joueur, il s'agit donc de savoir si on est dans le premier cas en trouvant une case certaine, et si on n'y arrive pas, reste à chercher une case presque certaine pour minimiser le risque de perdre.

C.3.2 Formalisation mathématique

Nos inconnues seront le contenu des cases j encore couvertes. On les notera x_j et elles pourront prendre deux valeurs :

– $x_j = 1$ s'il y a une mine dans la case j.
– $x_j = 0$ s'il n'y en a pas.

Les cases découvertes i donneront chacune une équation sur les x_j :

$$\sum_j A_{ij} x_j = B_i$$

avec $A_{ij} = 1$ si la case j est voisine de la case i, 0 sinon, et B_i égal au nombre de mines voisines indiquée par la case découverte. Toutes ces équations seront regroupées dans le système linéaire :

$$AX = B$$

Ce système sera en général sous-déterminé, autorisant plusieurs solutions X possibles.

Notre idée maîtresse sera de chercher une loi de probabilité $p\left((x_1,...,x_n)\right)$ sur le contenu des cases qui soit compatible avec ce système linéaire. Ceci nous permettra de calculer la probabilité et l'espérance de chaque case :

$$p\left(x_j = 1\right) = E\left[x_j\right] = \overline{x_j} \qquad p\left(x_j = 0\right) = 1 - \overline{x_j}$$

On cherchera à trouver la loi de probabilité qui maximise l'entropie, i.e. qui n'apporte pas plus d'information que celle vraiment contenue dans le système $AX = B$. Ainsi, on ne peut trouver une case certaine ($p\left(x_j = 1\right) = 1$ ou $p\left(x_j = 0\right) = 1$), que si le système $AX = B$ impose vraiment cette valeur[47].

Nous allons voir deux façons d'appliquer cette idée :

- En factorisant la loi de probabilité $p\left((x_1,...,x_n)\right) = p\left(x_1\right) * ... * p\left(x_n\right)$, ce qui revient à supposer les x_i indépendants. Cette hypothèse n'est bien évidemment pas vérifiée, mais elle permet d'écrire un algorithme efficace qui trouve correctement les cases certaines.
- En revenant à la loi de probabilité "avec dépendance" $p\left((x_1,...,x_n)\right)$. La minimisation de l'entropie mène alors à une loi uniforme : toutes les configurations $(x_1,...,x_n)$ possibles sont équiprobables, et calculer la probabilité d'une case x_j revient alors à dénombrer le nombre de configurations possibles avec $x_j = 1$ et le nombre de configurations possibles avec $x_j = 0$. Nous présenterons une solution basée sur le premier algorithme, qui permet de parcourir l'ensemble des solutions possibles de façon relativement efficace.

C.3.3 Loi indépendante

Nous allons donc supposer dans un premier temps que les x_j sont indépendants, ce qui permet de factoriser la loi de probabilité. Cette hypothèse est fausse (les cases sont liées par le système d'équation $AX = B$), sauf au moins pour deux catégories de cases :

- Celles qui n'interviennent pas dans le système linéaire : on trouvera pour ces cases une probabilité $p\left(x_j = 0\right) = p\left(x_j = 1\right) = 1/2$ correcte.

47. On peut alors penser à une méthode non-probabiliste pour trouver les cases certaines. Il s'agit de chercher, en utilisant par exemple la méthode du pivot de Gauss, à former une équation de la forme $x_i = 0$ ou $x_i = 1$ par combinaison linéaire des équations du système $AX = B$. Le système étant vérifié, l'équation ainsi créée sera automatiquement vérifiée, et on sera donc certain du contenu de la case. Nous avons essayé cette approche, mais n'avons pas trouvé de méthode de recherche complètement exhaustive.

- Celles qui sont certaines (i.e. qui prennent toujours la même valeur, 0 ou 1, quelle que soit la valeur des autres cases) : on devrait donc trouver pour ces cases des valeurs correctes.

Pour les autres cases, les probabilités trouvées seront souvent fausses.

Écrivons donc la loi de probabilité factorisée :

$$p\left((x_1,\ldots,x_n)\right) = p\left(x_1\right) * \ldots * p\left(x_n\right)$$

L'information apportée par cette loi est la somme des informations apportée par chacune des lois $p\left(x_j\right)$. Pour une loi de ce type, x_j peut prendre les valeurs 0 et 1, et l'information est donc $I\left[p\left(x_j\right)\right] = p\left(x_j = 0\right) . \log p\left(x_j = 0\right) + p\left(x_j = 1\right) . \log p\left(x_j = 1\right)$, soit finalement :

$$I = \sum_j I\left[p\left(x_j\right)\right] = -\sum_j \left(\overline{x_j}. \log \overline{x_j} + (1 - \overline{x_j}). \log(1 - \overline{x_j})\right)$$

Le contrainte $AX = B$ se transmet aux valeur moyennes :

$$\forall i \qquad \sum_j A_{ij}.\overline{x_j} = B_i$$

Algorithme Pour minimiser l'information en prenant en compte ces contraintes, nous allons reformuler notre problème en écrivant le critère avec des multiplicateurs de Lagrange λ_i :

$$\min_{\overline{x_1},\ldots,\overline{x_n},\lambda_1,\ldots,\lambda_m} L = -\sum_j \left(\overline{x_j}. \log \overline{x_j} + (1 - \overline{x_j}). \log(1 - \overline{x_j})\right) + \sum_i \lambda_i (\sum_j A_{ij}.\overline{x_j} - B_i)$$

L'optimisation de ce critère peut se faire par minimisation alternée par rapport aux multiplicateurs λ_i et aux valeurs moyennes $\overline{x_j}$:

Initialisation : on choisit une première valeur de λ (nulle par exemple)

Boucle : à l'étape I :

- Mise à jour des $\overline{x_j}$: Minimisation du Lagrangien pour λ constant.
- Mise à jour des λ_i : Minimisation du Lagrangien pour $\overline{x_j}$ constant.

Jusqu'à convergence : les variations des $\overline{x_j}$ sont négligeables (inférieures à ϵ).

On peut facilement montrer que le critère est convexe, et ne possède donc qu'un seul minimum (local et global). En diminuant à chaque fois le critère, on est donc sûr de converger quelle que soit la valeur initiale de λ. Il nous reste à détailler chaque étape.

Mise à jour des $\overline{x_j}$ On trouve alors, en annulant la dérivé du critère par rapport à $\overline{x_j}$:

$$\frac{dL}{d\overline{x_j}} = -\left(\log \overline{x_j} + \log(1-\overline{x_j})\right) + \sum_i \lambda_{i \cdot j} A_{ij} = 0$$

$$\frac{\overline{x_j}}{1-\overline{x_j}} = \exp\left(\sum_i \lambda_{i \cdot j} A_{ij}\right) = 0$$

$$\overline{x_j} = \frac{1}{1+\exp(\sum_i \lambda_i . A_{ij})}$$

On peut remarquer que $\overline{x_j}$ est ainsi par construction compris dans l'intervalle [0,1] (les probabilités 0 et 1 pouvant être atteintes avec des λ_i proche de $+\infty$ ou $-\infty$).

Mise à jour des λ_i Il ne nous reste plus qu'à trouver les valeurs des multiplicateurs de Lagrange λ_i qui permettent de minimiser le critère, i.e. de vérifier effectivement la contrainte $AX = B$. Nous n'allons en-fait pas minimiser le critère à chaque itération (i.e. imposer $AX - B = 0$), mais simplement le diminuer (i.e. se rapprocher de $AX - B = 0$), en utilisant son approximation linéaire. La convergence du schéma itératif sera plus lente mais ne sera pas remise en cause.

L'approximation linéaire est donc : $AX(\lambda + d\lambda) - B = AX(\lambda) - B + A\frac{dX}{d\lambda}.d\lambda$. On veut rendre cette erreur nulle, on va donc mettre à jour λ en l'incrémentant de (on note $H = A\frac{dX}{d\lambda}$) :

$$d\lambda = -H^{-1}.(AX - B)$$

Pour cela, il nous suffit de calculer H (qui est symétrique, de taille égale aux nombres d'équations) :

$$H_{ii'} = \sum_j \overline{x_j}.(1-\overline{x_j}).A_{ij}.A_{i'j}$$

puis de l'inverser. Il faut donc éviter les valeurs propres nulles de H, qui apparaissent lorsque le système $AX = B$ est redondant. Pour éviter cela, on peut repérer et supprimer les équations redondantes en essayant de diagonaliser partiellement le système par pivotage de gauss.

Implémentation L'implémentation efficace de cet algorithme est intéressante en elle-même. Il doit en-effet être appliqué de nombreuses fois et doit donc être rapide. Nous l'avons implanté dans kmines, le démineur de KDE qui a l'avantage d'être en source libre, suivant le schéma suivant : le joueur joue normalement (pour découvrir quelques mines en début de jeu), puis lance la résolution automatique. Si une case certaine est trouvée, on supprime l'inconnue dans les équations en la remplaçant par sa valeur certaine. De plus, la case est découverte si elle est prévue comme non-minée, rajoutant une nouvelle équation au système. On peut alors relancer l'algorithme pour le système d'équations mis à jour. La résolution automatique s'arrête lorsque le jeu est fini, ou lorsque la position est "indécidable" (on ne trouve aucune case certaine).

Nous avons utilisé deux idées qui permettent une bonne optimisation des temps de calcul :

- L'utilisation d'une structure de matrice creuse pour A avec une en-tête et un parcours efficace de chaque colonne (qui représente chaque inconnue, et contient en particulier la valeur moyenne $\overline{x_j}$) et de chaque ligne (qui représente une équation, et contient en particulier B_i et λ_i) permet un calcul efficace des valeurs moyennes $\overline{x_j}$ pour λ donné, et un calcul efficace des erreurs $(AX - B)_i$ et des valeurs de H pour $\overline{x_j}$ donnés.
- L'utilisation d'une technique efficace de mise à jour : lorsque le système d'équations est mis à jour (parce que des cases ont été découvertes), on repart des anciennes valeurs de λ (et $\lambda_i = 0$ pour la nouvelle équation), et on ne traite que les lignes et colonnes (i.e. les λ_i et $\overline{x_j}$) invalidées à cause des changements dans le système $AX = B$. On réduit ainsi beaucoup le nombre d'itération nécessaire pour minimiser l'information.

Résultats Les résultats de cet algorithme correspondent à nos attentes :

- Il n'a jamais fait explosé de bombes, ni prévu de bombes là où il n'y en avait pas : les probabilités certaines (0 ou 1) prédites sont donc fiables. Autrement dit, il n'y a pas de faux positifs.
- Il est rapide : le démineur en mode expert est entièrement résolu, lorsque c'est possible, en moins de $3s$. Le nombre d'équation, et donc le taille de la matrice symétrique H qui doit être inversée, dépasse pourtant souvent la centaine.

C.3.4 Approche Corrélée

L'algorithme précédent produit des résultats valables sur les cases certaines, mais les autres probabilités sont en revanche souvent fausses (on ignore d'ailleurs s'il peut y avoir des faux négatifs, i.e. si on peut rater des cases certaines), et on ne peut donc conseiller l'utilisateur lorsque la position est "indécidable" (l'algorithme n'a trouvé aucune case certaine).

Pour obtenir des probabilités correctes, il faut prendre en compte la corrélation entre les valeurs des différentes cases. La loi $p\left((x_1,...,x_n)\right)$ doit donc prendre en compte toutes les configurations $(x_1,...,x_n)$ possibles, et ne peut plus se ramener aux seules valeurs moyennes des cases $\overline{x_j}$. La minimisation de l'information va alors nous donner une loi de probabilité uniforme (toutes les configurations possibles sont équiprobables), et on ne peut donc pas (à notre connaissance) éviter le comptage des configuration possibles.

L'algorithme présenté ici peut être rapproché du principe des probabilités composées :

$$Pr(x_1,...,x_n) = Pr(x_n|x_{n-1},...,x_1).Pr(x_{n-1}|x_{n-2},...,x_1)...Pr(x_2|x_1).Pr(x_1)$$

On va prendre ce principe à l'envers : on va essayer les deux possibilités pour une case (on choisit $x_1 = 0$ puis $x_1 = 1$) puis par récursion essayer les possibilités compatible avec le système $AX = B$ pour une deuxième case (x_2), et ainsi de suite, jusqu'à tomber sur une position "entièrement décidable" (i.e. où toutes les cases sont certaines). Ces positions seront les solutions du problème, et la récursion permettra de toutes les parcourir.

On va bien-sûr coupler cette idée avec notre premier algorithme de résolution pour accélérer le processus en limitant au maximum les cas où il faut essayer les deux valeurs de x_j.

On va enfin compter le nombre total de solutions S et pour chaque case le nombre de solutions s_j avec $x_j = 1$:

Initialisation : $S = 0$, $s_j = 0$ et version courante du système d'équation $AX = B$

Appel récursif de :

Fonction $Comptage(var\,S, var\,s_j, (AX = B))$

Appliquer la résolution "indépendante" du système $AX = B$.

Suivant le résultat :

- Si le jeu n'a pas été entièrement résolu (il reste des équations formant le système "indécidable" $A'X = B'$) : on choisit une case j non certaine, et on appelle le comptage par récurrence en supposant la case vide $Comptage(S, s_j, (A'X = B')$ et $(x_j = 0))$ puis pleine $Comptage(S, s_j, (A'X = B')$ et $(x_j = 1))$.
- Si le jeu a été entièrement résolu (il n'y plus aucune équation) : on a trouvé une solution (que l'on peut éventuellement noter) et on incrémente S et les s_j correspondant aux mines prévues.

Calculer les probabilités des cases : $Pr(x_j = 1) = 1 - Pr(x_j = 0) = \overline{x_j} = s_j/S$

Nous n'avons pas testé cet algorithme, mais étant donné la rapidité de la résolution indépendante (de l'ordre de la seconde) et le faible nombre de positions indécidables (rarement supérieur à 2, en tout cas dans les modes de jeux classiques), il devrait être relativement efficace.

Annexe D

Quelques éléments sur les primitives aléatoires

D.1 Géométrie des ensembles de primitives

La théorie développée par [Pennec, 1996] s'appuie sur une métrique Riemanienne, qui permet d'identifier localement la variété à un espace euclidien. Par cet intermédiaire, on peut intégrer des fonctions réelles sur des trajectoires (en particulier la vitesse, et donc calculer la longueur de ces trajectoires, ce qui permet de définir une distance entre primitives) ou sur tout ou partie de la variété (ce qui permet de définir une densité de probabilité et calculer l'espérance d'une variable aléatoire réelle).

Un cas particulier est celui des variétés usuelles dans le monde du recalage et de la vision : l'ensemble des normales orientées $\mathcal{S}_{\mathfrak{D}-1}$ ou non-orientées $\mathcal{P}_{\mathfrak{D}-1}$ (analogues respectivement à une sphère et un hémisphère), l'ensemble des rotations $\mathcal{SO}_{\mathfrak{D}}$, et quelques autres, et par conséquent l'ensemble des points orientés $\mathbb{R}^{\mathfrak{D}} \times \mathcal{S}_{\mathfrak{D}-1}$ ou semi-orientées $\mathbb{R}^{\mathfrak{D}} \times \mathcal{P}_{\mathfrak{D}-1}$ et l'ensemble des transformations rigides $\mathbb{R}^{\mathfrak{D}} \times \mathcal{SO}_{\mathfrak{D}}$. Ces ensembles de primitives sont vis-à-vis du groupe des transformations rigides globalement invariants (l'image d'une primitive par une transformation rigide est une primitive) et homogènes (on peut atteindre toute primitive à partir d'une autre primitive par application d'une transformation rigide). La théorie devra, pour être cohérente, être elle aussi invariante vis-à-vis de ce groupe de transformation rigide. Il suffira pour cela de choisir une métrique Riemanienne invariante.

Nous présenterons dans cette section un résumé concis de la mise en place de cette théorie, passant parfois sous silence certaines hypothèses et complications techniques. Il serait en effet bien trop long de présenter la théorie dans tout ces détails et toute sa généralité, et le lecteur pourra se reporter à une littérature extrêmement variée pour une présentation plus complète des variétés différentielles [Klingenberg, 1982; Carmo, 1992].

Nous commencerons par nous restreindre au cas où la variété $\mathcal{M}$ est une sous-variété d'un espace euclidien E[48], globalement invariante et homogène pour un groupe de transformation $\mathcal{G}$, les transformations rigides par exemple.

Nous verrons alors comment définir les courbes (i.e. les trajectoires), et serons ainsi amené à considérer l'espace tangent en un point de la la variété, espace vectoriel où sont regroupées les dérivés des courbes (i.e. les vitesses sur les trajectoires).

Une métrique Riemanienne sur ces espaces tangent permettra alors de définir la longueur d'une courbe (en intégrant la norme de la vitesse sur l'ensemble du parcours), puis la distance entre points (la longueur de la courbe la plus courte, appelée géodésique).

En utilisant les géodésiques partant d'un point pour "aplatir" la variété sur son espace tangent en ce point, nous définirons ensuite la carte exponentielle, qui généralisera la notion de coordonnées curvilignes (définie pour les courbes) aux variétés de dimension quelconque.

On pourra enfin, à l'aide de la métrique, définir la forme volume qui permettra d'intégrer une fonction directement sur la variété ou en se ramenant, grâce à la carte exponentielle, à un espace euclidien.

D.1.1 Groupe de transformation - ensemble de primitives invariant homogène

Nous utiliserons très souvent dans la suite la notion d'invariance et d'homogénéité d'un ensemble de primitives vis à vis d'un groupe de transformation donnée. Nous allons ici détailler ce que cette notion implique.

Donnons nous donc un ensemble de transformations de E, i.e. d'applications bijectives de E dans lui-même. Cette ensemble devra être un groupe pour la composition. Nous noterons alors l'ensemble $\mathcal{G}$, la composition $\circ$, l'inversion $^{-1}$ et l'élément neutre Id. Sauf mention contraire, $\mathcal{G}$ sera l'ensemble des transformations rigides en 3D, qui seront notées $T = (R,t)$, et appliquées à un point de la manière suivante :

$$(R,t) \star x = R.x + t$$

La composition, l'inversion et l'élément neutre sont définis comme suit :

$$(R,t) \circ (R',t') = (R.R',t + R.t')$$

$$(R,t)^{-1} = (R^t, - R^t.t)$$

$$Id = (Id,0)$$

48. le théorème de Whitney prouve en fait que l'on peux toujours plonger une variété différentielle de dimension dans un espace vectoriel de dimension au plus $\mathfrak{D} = 2\mathcal{D} + 1$

Cet ensemble ne sera intéressant que lorsqu'on pourra définir l'application d'une transformation $\mathtt{f}$ de $\mathcal{G}$ à une primitive $\mathtt{x}$ de $\mathcal{M}$. Le résultat sera alors noté $\mathtt{f} \star \mathtt{x}$.

Nous nous intéresserons aux ensembles de primitives globalement invariants vis à vis de $\mathcal{G}$: l'image d'une primitive de $\mathcal{M}$ par une transformation de $\mathcal{G}$ devra être elle-même une primitive de $\mathcal{M}$:

$$\forall \mathtt{x} \in \mathcal{M}, \forall \mathtt{f} \in \mathcal{G} \qquad \mathtt{f} \star \mathtt{x} \in \mathcal{M}$$

Ce sera en particulier le cas de du groupe de transformation $\mathcal{G}$ lui même (en prenant $\mathtt{f} \star \mathtt{g} = \mathtt{f} \circ \mathtt{g}$), et, pour les transformations rigides, le cas de l'ensemble des points, des repères orientés, semi-orientés ou non-orientés, des normales orientées ou non orientées, des points orientés (munis d'une normale orientée) et semi-orientés (munis d'une direction, i.e. une normale non-orientée). Nous donnons ici les formules pour la composition à gauche avec les normales et les rotations :

$$(R,t) \star \mathtt{n} = R.\mathtt{n}$$

$$(R,t) \star R' = R.R'$$

Nous exigerons de plus que l'ensemble de primitives soit homogène vis à vis de $\mathcal{G}$: nous choisirons un point de $\mathcal{M}$ que nous appellerons origine et noterons $\mathtt{o}$, et devrons pouvoir atteindre tous les autres points de $\mathcal{M}$ par application d'une certaine transformation à notre origine $\mathtt{o}$:

$$\forall \mathtt{x} \in \mathcal{M} \qquad \exists \mathtt{f} \in \mathcal{G} / \mathtt{x} = \mathtt{f} \star \mathtt{o}$$

Notons que le choix de l'origine sera parfois canonique (l'identité s'impose dans le cas des transformations, par exemple), parfois arbitraire (dans le cas des normales, aucune n'a de propriété particulière).

Les transformations permettant d'atteindre la primitive $\mathtt{x}$ à partir de l'origine seront appelées fonctions de placement en $\mathtt{x}$, et rassemblées dans l'ensemble $F_{\mathtt{x}}$ appelé coset de $\mathtt{x}$:

$$F_{\mathtt{x}} = \{\mathtt{f} \in \mathcal{G} / \mathtt{f} \star \mathtt{o} = \mathtt{x}\}$$

Nous utiliserons souvent dans les calculs une de ces fonctions de placement en $\mathtt{x}$, que nous appellerons fonction de placement canonique et noterons $\mathtt{f}_{\mathtt{x}}$. On veillera bien-sûr à choisir la fonction la plus simple possible.

Un cas particulier est le coset de l'origine (i.e. l'ensemble des transformations qui ne déplacent pas l'origine) noté $\mathcal{H}$:

$$\mathcal{H} = F_{\mathtt{o}} = \{\mathtt{h} \in \mathcal{G} / \mathtt{h} \star \mathtt{o} = \mathtt{o}\}$$

On montre facilement que ce coset particulier est un sous-groupe de $\mathcal{G}$ appelé sous groupe d'isotropie. On peut retrouver les autres cosets à partir du sous groupe d'isotropie et de la fonction de placement canonique :

$$F_{\mathtt{x}} = \mathtt{f}_{\mathtt{x}} \circ \mathcal{H}$$

On peut donc identifier l'espace quotient $\mathcal{G}/\mathcal{H}$ avec $\mathcal{M}$, et aussi identifier le coset $F_{\mathtt{x}} = \mathtt{f}_{\mathtt{x}} \circ \mathcal{H}$, la fonction de placement standard $\mathtt{f}_{\mathtt{x}}$ et la primitive $\mathtt{x}$.

Lorsque $\mathcal{M}$ est l'ensemble des transformations lui même, il n'existe qu'une seule fonction de placement associée à une transformation $\mathtt{g}$ qui est $\mathtt{g}$ elle même. Le coset de $\mathtt{g}$ est donc réduit au singleton $F_{\mathtt{g}} = \{\mathtt{g}\}$ et le sous groupe d'isotropie est réduit à l'identité : $\mathcal{H} = \{Id\}$.

Il est bien évident que tout ce qui suit doit, pour avoir un comportement satisfaisant, être invariant vis à vis du groupe de transformation. Par exemple, nous n'utiliserons que des distances invariante, qui vérifient :

$$\forall \mathtt{x},\mathtt{y} \in \mathcal{M}, \forall \mathtt{f} \in \mathcal{G} \qquad d(\mathtt{f} \star \mathtt{x}, \mathtt{f} \star \mathtt{y}) = d(\mathtt{x},\mathtt{y})$$

Cette propriété d'invariance devra en particulier être vérifiée pour le sous-groupe d'isotropie. On scindera donc en général cette propriété d'invariance en deux parties, en faisant apparaître le sous-groupe d'isotropie et les fonctions de placement canoniques :

$$\forall \mathtt{x},\mathtt{y} \in \mathcal{M}, \forall \mathtt{h} \in \mathcal{H} \qquad d(\mathtt{h} \star \mathtt{x}, \mathtt{h} \star \mathtt{y}) = d(\mathtt{x},\mathtt{y})$$

$$\forall \mathtt{x},\mathtt{y} \in \mathcal{M}, \forall \mathtt{f}_{\mathtt{x}} \qquad d(\mathtt{f}_{\mathtt{x}} \star \mathtt{x}, \mathtt{f}_{\mathtt{x}} \star \mathtt{y}) = d(\mathtt{x},\mathtt{y})$$

On pourra en particulier se ramener à la distance à l'origine (qui peut être considérée comme une norme, notée N, bien qu'elle ne soit pas homogène[49]) à l'aide de la fonction de placement en $\mathtt{x}$:

$$d(\mathtt{f}_{\mathtt{x}}^{-1} \star \mathtt{x}, \mathtt{f}_{\mathtt{x}}^{-1} \star \mathtt{y}) = d(\mathtt{o}, \mathtt{f}_{\mathtt{x}}^{-1}(\mathtt{y})) = N(\mathtt{f}_{\mathtt{x}}^{-1}(\mathtt{y}))$$

Fonction de placement pour les normales en 3D Nous avons déjà vu, que dans le cas des normales, l'application d'une transformation rigide prenait la forme suivante : $(R,t) \star \mathtt{n} = R.\mathtt{n}$. Dans le cas 3D, les rotations peuvent être définies par leur axe a (lui aussi une direction, une normale) et leur angle θ, ou directement par leur vecteur $v = \theta a$. Nous noterons une telle rotation sous la forme $R_{a,\theta}$ ou R_v.

Choisissons une origine arbitraire à l'ensemble des normales, $\mathtt{o} = (1,0,0)$ par exemple. Le sous-groupe d'isotropie est constitué de l'ensemble des transformations rigides qui laissent cette normale inchangée. La translation n'ayant aucune importance, reste à savoir quelles rotations laissent une normale inchangée. Il s'agit bien-sûr des rotations ayant

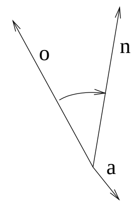
49. Une norme est dite homogène si elle vérifie $N(\lambda \mathtt{x}) = |\lambda|\, N(\mathtt{x})$. Hors $\lambda \mathtt{x}$ n'est tout simplement pas défini dans la variété.

pour axe cette normale, et un angle quelconque, qui seront donc notées $R_{\mathtt{o},\theta}$. En résumé, le sous-groupe d'isotropie est :

$$\mathcal{H} = \{(R_{\mathtt{o},\theta},t); t,\theta\, quelconque\}$$

Reste à trouver une fonction de placement simple permettant d'atteindre la normale $\mathtt{n}$ à partir de la normale origine $\mathtt{o}$. Nous prendrons une translation nulle, et la rotation autour d'un axe perpendiculaire à ces deux normales est un choix simple, car cet axe est obtenu par produit vectoriel : $a = \frac{\mathtt{o}\wedge\mathtt{n}}{\|\mathtt{o}\wedge\mathtt{n}\|}$, et l'angle de la rotation est alors l'angle entre les deux normales $(\widehat{\mathtt{o},\mathtt{n}})$. Finalement, la fonction de placement canonique est :

$$\mathtt{f}_{\mathtt{n}} = R_{(\frac{\mathtt{o}\wedge\mathtt{n}}{\|\mathtt{o}\wedge\mathtt{n}\|},(\widehat{\mathtt{o},\mathtt{n}}))}$$



Notons que dans le cas où la normale $\mathtt{n}$ est égale à $\mathtt{o}$ ou à son opposée, le vecteur $\mathtt{o}\wedge\mathtt{n}$ est nul, et la formule ci dessus ne peut être appliquée. Lorsque $\mathtt{n} = \mathtt{o}$, l'identité est la fonction de placement la plus simple. Lorsque $\mathtt{n} = -\mathtt{o}$, n'importe quelle rotation d'angle 180 degrés et d'axe perpendiculaire à $\mathtt{o}$ peut convenir :

$$\mathtt{f}_{\mathtt{o}} = Id$$

$$\mathtt{f}_{-\mathtt{o}} = R_{(a,\pi\, rad)} \quad avec\ a \perp \mathtt{o}$$

D.1.2 Éléments de géométrie différentielle

Nous allons maintenant voir de plus près comment aborder la géométrie non-linéaire de notre ensemble de primitives $\mathcal{M}$, qui sera, dans les exemples, l'ensemble des normales ou l'ensemble des transformations rigides. Nous verrons en particulier les cartes locales, qui permettent de se ramener à un ouvert de $\mathbb{R}^{\mathcal{D}}$, plus pratique pour faire les calculs.

C'est dans cette partie que nous ferons, sans même le préciser, le plus de simplifications. Le lecteur se reportera à tout bon cours ou livre sur les variétés différentielles pour plus de rigueur [Klingenberg, 1982; Carmo, 1992].

D.1.2.1 Sous-Variété différentielle d'un espace euclidien et carte locale

Nous ne présentons pas la définition générale, mais complexe, d'une variété différentielle, préférant utiliser l'espace euclidien ambiant E, qui est une variété différentielle. Il suffit donc de caractériser les sous-variétés différentielles de E. Intuitivement, il s'agit des courbes (i.e. de fonctions d'un réel t), des surfaces (i.e. des fonctions de couple de réel (u,v)), et plus généralement d'objets que l'on peux mettre sous la forme de fonction d'un n-uplet de réels, avec n fixe (c'est la dimension de la variété). La définition générale suit ces idées, de façon locale et donc sans aucune contrainte topologique (comme la connexité). Elle prend en revanche en compte les aspects analytiques, comme les possibilités de différentiation :

Définition D.1 *Sous-variété différentielle - classe - dimension*

Une sous-variété différentielle de E est une partie $\mathcal{M}$ de E localement difféomorphe en tout point $\mathtt{x}$ à un espace euclidien $T_{\mathtt{x}}$ de dimension $\mathcal{D}$.

$\mathcal{D}$ doit être constant et est appelé dimension de la sous-variété.

S'il est possible de trouver en tout point un difféomorphisme de classe C^k, la sous-variété est dite de classe C^k.

Une sous-variété différentielle est elle-même une variété différentielle.

Ceci signifie qu'en tout point $\mathtt{x}$ de $\mathcal{M}$, il existe un difféomorphisme de classe C^k (i.e. une application bijective, C^k, et d'inverse C^k, où $k \geq 1$) défini entre un ouvert D de $T_{\mathtt{x}}$ et un voisinage $\mathtt{D}$ de $\mathtt{x}$ dans $\mathcal{M}$. Une telle application permet de définir les coordonnées d'un point de $\mathtt{D}$. Elle est appelée carte locale :

Définition D.2 *Carte locale*

Une carte locale $\pi_{\mathtt{x}}$ de $\mathcal{M}$ est un difféomorphisme de classe C^k défini comme suit :

$$\pi_{\mathtt{x}} \quad : \quad \begin{array}{ccc} D_{\pi_{\mathtt{x}}}\, \text{ouvert de}\, T_{\mathtt{x}} & \rightarrow & \mathtt{D}_{\pi_{\mathtt{x}}} \subset \mathcal{M} \\ \overrightarrow{xy} = \begin{bmatrix} \overrightarrow{xy}_1 \\ \vdots \\ \overrightarrow{xy}_{\mathcal{D}} \end{bmatrix} & \mapsto & \mathtt{y} = \pi_{\mathtt{x}}(\overrightarrow{xy}) \end{array}$$

De plus, la carte est dite centrée en $\mathtt{x}$ lorsque $0 \in D_\pi$ et $\pi(0) = \mathtt{x}$.

Si $\mathtt{y} \in \mathtt{D}_{\pi_{\mathtt{x}}}$, il existe un unique $\overrightarrow{xy} = \begin{bmatrix} \overrightarrow{xy}_1 \\ \vdots \\ \overrightarrow{xy}_{\mathcal{D}} \end{bmatrix}_i \in D_{\pi_{\mathtt{x}}}$ *tel que $\mathtt{y} = \pi_{\mathtt{x}}(\overrightarrow{xy})$.*

$\overrightarrow{xy}_1,...,\overrightarrow{xy}_{\mathcal{D}}$ sont appelées coordonnées de $\mathtt{y}$ dans la carte.

Nous nous restreindrons dans la suite aux sous-variétés suffisamment régulières, i.e. de classe C^k où $k \geq 2$.

Les cartes locales sont en pratiques très importantes. Elles permettent en-effet de se ramener à un ouvert d'un espace vectoriel, et donc d'effectuer certains calculs. Il convient cependant de bien distinguer un point de la sous-variété et ses coordonnées co- ou contra-variantes (voir D.1.3.2) dans l'espace ambiant (représenté en général par une minuscule droite : $\mathtt{y}$,$\mathtt{n}$,$\mathtt{y}_i$,$\mathtt{n}_i$,$\mathtt{y}^i$,$\mathtt{n}^i$,...), qui ne dépendent pas de la carte utilisée, et sa représentation dans une carte locale centrée en $\mathtt{x}$ (notée $\overrightarrow{xy}$, pour bien indiquer que l'on travaille dans la carte centrée en $\mathtt{x}$) et ses coordonnées dans cette carte (notées $\overrightarrow{xy}_i$,...), qui sont relatives à la carte utilisée.

Il faut de plus noter qu'il n'est pas toujours possible de définir une carte couvrant la totalité de la sous-variété. Ainsi, dans la cas de la sphère, on doit exclure au moins un point. C'est l'exemple typique de la cartographie de la terre, où on est obligé d'utiliser un atlas regroupant plusieurs cartes locales.

Cas d'un groupe de Lie Nous avons vu que le groupe des transformations $\mathcal{G}$ était globalement invariant et homogène vis à vis de lui même. Un cas particulier est la cas où ce groupe est un groupe de Lie. Il possède alors une structure de variété différentielle, si bien qu'on peut utiliser dans la suite $\mathcal{M} = \mathcal{G}$. Ce sera le cas des transformations rigides.

Exemples

Espace Affine Un sous-espace affine F de E est bien-entendu une sous-variété différentielle. En-effet, si on dispose d'une base f_1,...,f_n de F, on peut définir très facilement une carte locale centrée en un point $\mathtt{x}$ de F et couvrant la totalité de l'espace avec la fonction : $\overrightarrow{xy} = \begin{bmatrix} \overrightarrow{xy}_1 \\ \vdots \\ \overrightarrow{xy}_{\mathcal{D}} \end{bmatrix} \in \mathbb{R}^{\mathcal{D}} \mapsto \mathtt{y} = \mathtt{x} + \overrightarrow{xy}_1.f_1 + ... + \overrightarrow{xy}_{\mathcal{D}}.f_{\mathcal{D}}$.

Courbe Les sous-variétés différentielles de E de dimension 1 et de classe C^k ne sont autres que les courbes de E de classe C^k. La courbe est souvent connexe et donnée sous forme paramétrée : $\gamma : [a,b] \to E$. On utilise en général une fonction γ ayant des propriétés de régularités suffisantes pour qu'elle soit une carte locale.

Normales Il va sans dire que l'ensemble des normales orientées et non-orientées de $\mathbb{R}^{\mathfrak{D}}$ sont tous deux des sous-variétés C^∞ de dimension $\mathcal{D} = \mathfrak{D} - 1$. Nous donnerons un exemple de carte locale un peu plus loin.

Rotations L'ensemble des rotations est lui une sous-variété C^∞ de $L(\mathbb{R}^{\mathfrak{D}})$ (espace des applications linéaires de $\mathbb{R}^{\mathfrak{D}}$ dans lui-même), de dimension $\mathcal{D} = \frac{\mathfrak{D}(\mathfrak{D}-1)}{2}$ et même un groupe de Lie.

En dimension 3, on peux exhiber une carte locale centrée autour de l'identité. En effet, toute rotation est caractérisée par un axe, représenté par un vecteur unitaire $\mathtt{n}$ et un angle θ, et notée $R(n,\theta)$. On peut donc représenter toute rotation par le vecteur $\theta.\mathtt{n}$, situé dans $\mathbb{R}^3$. On vérifie que ceci définit bien une carte locale, lorsqu'on impose $\theta < \pi$.

D.1.2.2 Trajectoires & Courbes

La notion de courbe est simple à établir sur une variété différentielle. Il convient simplement de faire la distinction entre une trajectoire et une courbe. Une trajectoire est intuitivement une fonction régulière du temps, i.e. d'un intervalle réel $[a,b]$ dans $\mathcal{M}$. Deux trajectoires peuvent suivre la même courbe et différer simplement par la vitesse de parcours. On définit donc les courbes et trajectoires de la manière suivante :

Définition D.3 *Trajectoires et courbes*

Soit $\mathcal{M}$ une variété différentielle.

Une trajectoire de $\mathcal{M}$ est une fonction γ de $[a,b]$ dans $\mathcal{M}$ de classe au moins C^2, sans point fixe.

On définit une relation d'équivalence entre deux trajectoires de la façon suivante : $\gamma_1 : [a,b] \to \mathcal{M} \approx \gamma_2 : [c,d] \to \mathcal{M} \Leftrightarrow \exists f : [a,b] \to [c,d]\, croissante - surjective - C^2 / \gamma_1(t) = \gamma_2(f(t))$.

Les courbes de $\mathcal{M}$ sont les classes d'équivalences de l'ensemble des trajectoires de $\mathcal{M}$ pour $\approx$.

Si nous disposons d'une carte locale π, nous pouvons identifier la trajectoire γ dans $\mathtt{D}_\pi$ (donc dans $\mathcal{M}$) et $\overrightarrow{\gamma}$ sa représentation dans D_π (i.e. dans la carte locale), les deux étant reliées par $\gamma(t) = \pi(\overrightarrow{\gamma}(t))$.

D.1.2.3 Espace Tangent

L'espace tangent correspond à la notion de tangente dans le cas d'une courbe. Cette notion est relative à un point et s'appuie sur la dérivé de la courbe en ce point. Pour la généraliser, il faut donc se donner un point de la variété différentielle, et voir comment se répartissent les dérivés des courbes en ce point.

Considérons donc l'ensemble des trajectoires de $\mathcal{M}$ passant par un point $\mathtt{x}$ (on supposera $\gamma(0) = \mathtt{x}$ pour simplifier). Ces trajectoires peuvent être vues comme des trajectoires dans E. La notion de vecteur tangent est alors usuellement définie et il s'agit d'un vecteur de E. L'ensemble de ces vecteurs forme un sous-espace vectoriel de E, de dimension $\mathcal{D}$ et noté $T_{\mathtt{x}}\mathcal{M}$. On peut s'en convaincre et en exhiber une base en utilisant une carte locale π telle que $\mathtt{x} \in \mathtt{D}_\pi$. $\mathtt{x}$ est alors identifié à $\overrightarrow{x} \in D_\pi$. Soit alors une trajectoire quelconque définie à l'aide de la carte locale : $\gamma(t) = \pi\left(\begin{bmatrix} \overrightarrow{\gamma}_1(t) \\ \vdots \\ \overrightarrow{\gamma}_{\mathcal{D}}(t) \end{bmatrix}\right)$, on montre facilement que $\frac{d\gamma}{dt}(0) = \sum_i \frac{\partial \pi}{\partial \overrightarrow{x}_i}(\overrightarrow{x}) \frac{d\overrightarrow{\gamma}_i}{dt}(0)$. Les vecteurs $\frac{\partial \pi}{\partial \overrightarrow{x}_i}(\overrightarrow{x})$ forment donc une famille génératrice pour notre espace tangent. On montre aussi que le déterminant de cette famille, égal au jacobien de la carte locale, ne peut pas être nul. Il s'agit alors d'une famille libre, donc d'une base, prouvant ainsi que l'espace tangent est bien de dimension $\mathcal{D}$.

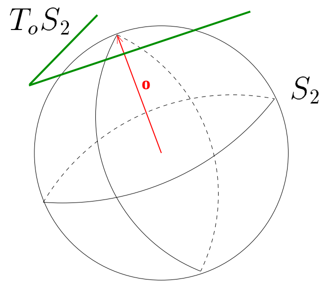


FIG. D.1 – *Plan tangent à la sphère des normales*

Ces quelques remarques sont rassemblées dans la définition suivante :

Définition D.4 *Espace tangent*

Soit $\mathtt{x}$ *un point de* $\mathcal{M}$*.*

L'ensemble $\{\dot\gamma(t_0);\ \gamma(t_0) = \mathtt{x}\}$ *forme un sous-espace vectoriel de* E*, de dimension* $\mathcal{D}$ *(dimension de* $\mathcal{M}$*), appelé espace tangent à* $\mathcal{M}$ *au point* $\mathtt{x}$ *et noté* $T_{\mathtt{x}}\mathcal{M}$*.*

Étant donné une carte locale π *et* $\overrightarrow{x}$ *tel que* $\pi(\overrightarrow{x}) = \mathtt{x}$*, les vecteurs* $\frac{\partial\pi}{\partial\overrightarrow{x}_i}(\overrightarrow{x})$ *forment une base de* $T_{\mathtt{x}}\mathcal{M}$*, appelée base naturelle dans la carte locale* π*.*

Il faut remarquer que la base naturelle dans une carte locale est rarement orthonormée.

Exemples

Espace Affine Dans un espace affine F, l'espace tangent en un point $\mathtt{x}$ n'est autre que l'espace vectoriel associé à l'espace affine en $\mathtt{x}$, i.e. l'ensemble $\{\mathtt{y} - \mathtt{x}; \mathtt{y} \in F\}$.

Normales L'espace des normales orientées est identifié à la sphère unité. L'espace tangent en une normale $\mathtt{n}$ est, comme le suggère l'intuition, le plan tangent à la sphère, d'équation $(\mathtt{x}|\mathtt{n}) = 1$ (voir fig. D.1). Nous identifierons, pour simplifier, ce plan au plan d'équation $(\mathtt{x}|\mathtt{n}) = 0$ (i.e. le plan parallèle mais passant par l'origine de l'espace vectoriel ambiant).

Rotations Nous allons ici détailler les calculs permettant de mettre en évidence le plan tangent.

Soit donc une trajectoire $R(t)$ sur $\mathcal{SO}_{\mathfrak{D}}$. Elle vérifie, pour tout t : $R(t).R(t)^t = R(t)^t.R(t) = Id$. En dérivant :

$$\left(\frac{dR(t)}{dt}.R(t)^t\right) + \left(\frac{dR(t)}{dt}.R(t)^t\right)^t = 0 \quad \left(R(t)^t.\frac{dR(t)}{dt}\right) + \left(R(t)^t.\frac{dR(t)}{dt}\right)^t = 0$$

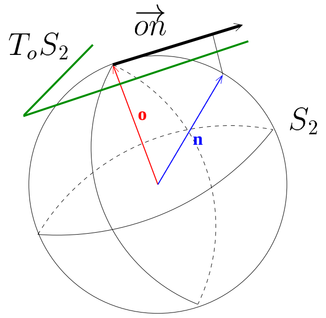


FIG. D.2 – *Carte locale pour les normales*

Les matrices $\left(\frac{dR(t)}{dt}.R(t)^t\right)$ et $\left(R(t)^t.\frac{dR(t)}{dt}\right)$ sont donc antisymétriques, et il existe donc deux matrices antisymétriques M_g et M_d telles que :

$$\frac{dR(t)}{dt} = M_d.R = R.M_g$$

L'espace tangent à l'identité $T_{Id}\mathcal{S}O_{\mathfrak{D}}$ est donc l'ensemble des matrices antisymétriques (le raisonnement précédent prouve l'inclusion, il suffit alors d'utiliser l'égalité des dimensions des deux espaces pour prouver l'égalité). On retrouve l'espace tangent en R en appliquant R à $T_{Id}\mathcal{S}O_{\mathfrak{D}}$:

$$T_R\mathcal{S}O_{\mathfrak{D}} = R \circ T_{Id}\mathcal{S}O_{\mathfrak{D}} = \{M.R/M + M^t = 0\} = \{R.M/M + M^t = 0\} \tag{D.1}$$

Espace tangent et carte locale L'espace tangent en un point $\mathtt{x}$ est l'espace de départ idéal pour définir une carte locale centrée en $\mathtt{x}$. Il faut cependant alors faire attention, car toutes les cartes ne sont pas définies avec le même espace de départ. Par exemple, $\overrightarrow{xy}$ - qui représente $\mathtt{y}$ dans la carte centré en $\mathtt{x}$ - sera un vecteur de $T_{\mathtt{x}}\mathcal{M}$, alors que $\overrightarrow{yx}$ - qui représente $\mathtt{x}$ dans la carte centré en $\mathtt{y}$ - sera un vecteur de $T_{\mathtt{y}}\mathcal{M}$. Toute égalité du genre $\overrightarrow{xy} = -\overrightarrow{yx}$ serait donc absurde...

Espace affine Pour un espace affine F, la carte locale centrée en $\mathtt{x}$ la plus simple définie à partir du plan tangent, i.e. de l'espace vectoriel E associé est la translation $t_{\mathtt{x}}$ qui amène l'origine de E sur $\mathtt{x}$.

Normales Essayons de définir une carte locale centrée autour d'une normale $\mathtt{o}$, dans le plan tangent en $\mathtt{o}$. Soit une normale $\mathtt{n}$. Il est naturel de représenter cette normale, dans la carte locale, par un vecteur colinéaire, et de même sens que le projeté orthogonal $\Pi_{\mathtt{o}\perp}(\mathtt{n}) = \mathtt{n} - \mathtt{o}(\mathtt{n}|\mathtt{o})$ de $\mathtt{n}$ sur le plan tangent ($\Pi_{\mathtt{o}\perp}$ est la projection orthogonale d'axe $\mathtt{o}$). Ce vecteur sera donc de la forme : $\overrightarrow{on} = \lambda.\Pi_{\mathtt{o}\perp}(\mathtt{n}) = \lambda.(\mathtt{n} - \mathtt{o}(\mathtt{n}|\mathtt{o}))$ (voir fig. D.2).

Plusieurs possibilités sont offertes pour la longueur de ce vecteur :

1. La distance angulaire : $\|\overrightarrow{on}\| = (\widehat{\mathtt{o},\mathtt{n}})$.

2. La distance vectorielle, définie à partir de la distance quadratique de $\mathbb{R}^{\mathfrak{D}}$: $\|\overrightarrow{on}\| = \|\mathtt{o} - \mathtt{n}\| = 2\sin(\frac{(\widehat{\mathtt{o},\mathtt{n}})}{2})$.
3. La distance tensorielle, définie par $\lambda = 1$, soit $\overrightarrow{on} = \Pi_{\mathtt{o}\perp}(\mathtt{n}) = (\mathtt{n} - \mathtt{o}(\mathtt{n}|\mathtt{o}))$ et $\|\overrightarrow{on}\| = \sin(\widehat{\mathtt{o},\mathtt{n}})$.

Nous utiliserons dans un premier temps la distance angulaire, qui est la distance intrinsèque car elle est la seule à correspondre à une métrique invariante. Nous reviendrons sur les deux autres distances dans la partie traitant des statistiques approchées.

Rotations 3D En 3D, les matrices antisymétriques, qui forment l'espace tangent en l'identité (voir eq. D.1), peuvent être écrites sous la forme :

$$\theta.S_n \quad avec\, S_n = \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix} et\ \|n\| = 1 \tag{D.2}$$

On a ainsi obtenu une carte locale définie à partir de l'espace tangent. On retrouve la carte définie dans la section précédente en identifiant $\theta.S_n$ et $\theta.n$.

La formule de Rodrigues permet de plus de faire le lien avec la matrice de la rotation :

$$R(n,\theta) = Id + \sin\theta.S_n + (1-\cos\theta).S_n^2 = \cos\theta.Id + \sin\theta.S_n + (1-\cos\theta).n.n^t \tag{D.3}$$

et inversement :

$$\theta = \arccos\left(\frac{Tr(R)-1}{2}\right) \quad et \quad S_n = \frac{R-R^t}{2.\sin\theta} \tag{D.4}$$

D.1.3 Distance induite par une métrique riemannienne sur une sous-variété différentielle connexe géodésiquement complète

Nous voudrions maintenant pouvoir mesurer des choses dans notre variété (des vitesses, des longueurs, des volumes). Étant donné que les espaces tangents sont vectoriels, on peut commencer par définir une métrique euclidienne sur ces espaces au travers d'un produit scalaire. Nous verrons ensuite comment l'utiliser.

D.1.3.1 Métrique Riemanienne - Métrique invariante

Définition D.5 *Métrique Riemanienne*

Pour définir une métrique Riemanienne sur $\mathcal{M}$ il faut définir une métrique euclidienne (ou de manière équivalent un produit scalaire $(.|.)$ ou un tenseur métrique G) sur chaque *espace tangent. On adoptera les notations suivantes :*

$$\forall v_x, w_x \in T_x\mathcal{M} \qquad (v_x|w_x) = v_x \overline{\otimes} G_{T_x\mathcal{M}} \overline{\otimes} w_x$$

$$\forall v_x \in T_x\mathcal{M} \qquad \|v_x\| = v_x \overline{\otimes} G_{T_x\mathcal{M}} \overline{\otimes} v_x$$

Il faudra de plus que la collection de produits scalaires ainsi formée soit continue et différentiable.

Le seul cas qui nous intéressera ici sera le cas où l'on utilise la restriction de la métrique euclidienne usuelle de l'espace ambiant E dans chaque espace tangent (soit pour les normales $(u|v) = Tr(u.v^t)$ et pour les rotations $(M_1|M_2) = Tr(M_1 M_2^t)$).

La section suivante permet de se familiariser avec cette métrique, en exposant comment mener les calculs dans la base naturelle d'une carte π avec le formalisme du calcul tensoriel.

D.1.3.2 Calcul tensoriel dans un sous-espace vectoriel

Nous avons vu que l'espace tangent $T_x\mathcal{M}$ était un sous-espace vectoriel de E. Nous avons de plus obtenu une base de cet espace (la base naturelle dans une carte donnée, en général quelconque). Nous voulons définir un produit scalaire sur $T_x\mathcal{M}$, comme la restriction du produit scalaire de E. Les vecteurs de $T_x\mathcal{M}$ peuvent alors être vus comme des tenseurs d'ordre 1, et nous souhaiterions pouvoir travailler avec les outils de base de la théorie des tenseurs euclidiens (base duale, coordonnées co- et contra-variantes, tenseur métrique, voir E.1) dans cette base. Ce travail sera d'autant plus utile que cette base n'est en général pas orthonormée.

Pour simplifier les notations, nous noterons F le sous-espace vectoriel. Il faudra bien évidement distinguer les coordonnées dans une base de E et les coordonnées dans une base de F. Pour cela, tous les indices seront précédés du nom de l'espace auquel ils correspondent. En particulier, un indice Ei (relatif à E) pourra varier entre 1 et $\mathfrak{D}$ (dimension de E), tandis qu'un indice Fi ne pourra varier qu'entre 1 et $\mathcal{D}$ (dimension de F, $\mathcal{D} \leq \mathfrak{D}$). Lorsque tous les indices correspondent au même espace, on pourra noter qu'une seule fois le nom de l'espace. Il conviendra aussi de distinguer le produit scalaire de E, noté $(\,|\,)$ et représenté par le tenseur métrique G, de coordonnées G_{Eij} et celui de F, noté $(\,|\,)_F$ et représenté par le tenseur métrique G_F, de coordonnées G_{Fij}.

Soit donc une base $\{e_{Ei}\}$ de E (en général orthonormée, auquel cas $[G_{Eij}] = Id$), et une base $\{f_{Fi}\}$ de F. La seconde est définie en fonction de la première par les relations suivantes : $f_{Fk} = e_{Ei} A^{Ei}_{Fk}$. La matrice $[A^{E\bullet}{}_{F\bullet}]$ rassemble donc les coordonnées contra-variantes de f_{Fk} dans sa colonne k. On peut aussi écrire ces relations sous la forme $f_{Fi} = A^{Ek}_{Fi} e_{Ek}$, la matrice associée vérifie : $[A_{F\bullet}{}^{E\bullet}] = [A^{E\bullet}{}_{F\bullet}]^t$, et rassemble donc les coordonnées contra-variantes de f_{Fi} dans sa ligne i.

On peut aussi utiliser les relations suivantes : $f_{Fi} = A_{Fi\,Ek} e^{Ek}$ ou $f_{Fk} = e^{Ei} A_{Ei\,Fk}$. La matrice $[A_{F\bullet E\bullet}]$ rassemble donc les coordonnées covariantes de f_{Fi} dans sa ligne i, et la matrice $[A_{E\bullet F\bullet}]$ rassemble les coordonnées covariantes de f_{Fk} dans sa colonne k.

Soient $[G_{E\bullet\bullet}]$ et $[G^{E\bullet\bullet}]$ les matrices du tenseur métrique dans notre base de E. On vérifie facilement que les règles de "l'ascenseur d'indice" sont respectées : $f_{Fi} = A_{Fi}^{Ek} e_{Ek} = A_{Fi}^{Ek} G_{E\,ki} e^{Ei}$ soit, sous forme matricielle, $[A_{F_\bullet}{}^{E\bullet}] = [A_{F\bullet E\bullet}][G^{E\bullet\bullet}]$, $[A^{F\bullet}{}_{E\bullet}] = [G^{E\bullet\bullet}][A_{E\bullet F\bullet}]$, et ainsi de suite.

Considérons maintenant le tenseur métrique dans F. On peut calculer ses coordonnées dans la base $\{f_{Fi}\}$ de la façon suivante : $G_{Fij} = (f_{Fi}|f_{Fj})_F = (A_{Fi}^{Ei} e_{Ei}|e^{Ej} A_{EjFj})_E = A_{Fi}^{Ek} A_{EkFj} = A_{Fi}^{Ei} g_{Eij} A_{Fj}^{Ej}$, soit, sous forme matricielle : $[G_{F\bullet\bullet}] = [A_{F_\bullet}{}^{E\bullet}][G_{E\bullet\bullet}][A^{E\bullet}{}_{F\bullet}]$. Les résultats classiques de la théorie des tenseurs restent valables lorsqu'on travaille dans F, considéré comme un espace vectoriel isolé. On peut donc calculer $[G^{F\bullet\bullet}] = [G_{F\bullet\bullet}]^{-1}$, et appliquer la règle de l'ascenseur d'indice : La base duale est définie par $f^{Fi} = G^{Fik} f_{Fk}$ et donc $[A^{F\bullet}{}_{E\bullet}] = [G^{F\bullet\bullet}][A_{F\bullet E\bullet}]$ par exemple.

Exemple d'utilisation Nous allons illustrer concrètement l'utilisation de ces résultats pour simplifier les calculs, lorsqu'on travaille dans une carte locale π. Nous prendrons l'exemple de la norme de la vitesse instantanée, exploitée dans la partie suivante. Le sous-espace vectoriel F sera l'espace tangent en un point $\pi(\vec{x})$, noté $T_{\pi(\vec{x})}\mathcal{M}$. Sa base sera la base naturelle $f_{Fi} = \frac{\partial \pi}{\partial \vec{x}_i}$, notée plus simplement $f_{\pi(\vec{x})\,i} = \frac{\partial \pi}{\partial \vec{x}_i}(\vec{x})$. Les coordonnées covariantes du tenseur métrique dans cette base seront notées plus simplement : $G_{\pi(\vec{x})\,ij}$, et calculable par la simple relation $g_{\pi(\vec{x})\,ij} = (f_{\pi(\vec{x})\,i}|f_{\pi(\vec{x})\,j}) = (\frac{\partial \pi}{\partial \vec{x}_i}(\vec{x})|\frac{\partial \pi}{\partial \vec{x}_j}(\vec{x}))$.

Supposons donc que nous ayons l'équation de cette trajectoire dans une carte locale : $\gamma(t) = \pi(\vec{\gamma}(t))$. Lorsqu'on dérive, on trouve $\dot{\gamma}(t) = \frac{d\vec{\gamma}_i}{dt}(t)\, f_{\pi(\vec{x})\,i}$. On a alors : $\|\dot{\gamma}(t)\|^2 = \frac{d\vec{\gamma}_i}{dt}(t)\, (f_{\pi(\vec{x})\,i}|f_{\pi(\vec{x})\,j})\, \frac{d\vec{\gamma}_j}{dt}(t)$, soit $\|\dot{\gamma}(t)\|^2 = \frac{d\vec{\gamma}_i}{dt}(t)\, G_{\pi(\vec{x})\,ij}\, \frac{d\vec{\gamma}_j}{dt}(t)$. La méthode de calcul est donc simple : calculer les coordonnées de $G_{\pi(\vec{x})\,ij}$ dans la base naturelle, et considérer les dérivées $\frac{d\vec{\gamma}_i}{dt}(t)$ comme les coordonnées dans cette base.

D.1.3.3 Distances et Géodésiques

Maintenant que nous savons mesurer une vitesse en un point d'une courbe, on peut en déduire la longueur de la courbe, tout simplement en intégrant la norme de la vitesse. On peut alors chercher la courbe la plus courte entre deux points et définir ainsi la distance entre ces points (comme dans un espace vectoriel, où la distance est la longueur du segment de droite, étant bien connu que le plus court chemin entre deux points est la ligne droite).

Donnons nous deux points $\mathtt{x}$ et $\mathtt{y}$ de $\mathcal{M}$. Nous allons donc ici considérer l'ensemble des trajectoires reliant ces deux points, que l'on notera dans cette section $\Theta_{\mathtt{xy}} = \{\gamma : [a,b] \to \mathcal{M} / \gamma(a) = \mathtt{x}, \gamma(b) = \mathtt{y}\}$. Nous savons, grâce à la métrique Riemanienne, calculer la vitesse instantanée en un point d'une trajectoire. Ceci permet de définir la longueur d'une trajectoire : $L_a^b(\gamma) = \int_a^b \|\dot{\gamma}(t)\| \, dt$. On vérifie que cette définition ne dépend en fait pas de la trajectoire choisie mais uniquement de la courbe suivie. La distance entre deux points $\mathtt{x}$ et $\mathtt{y}$ de $\mathcal{M}$ est définie comme la borne inférieure des longueurs des courbes joignant ces deux points : $d(x,y) = \inf_{\gamma \in \Theta_{\mathtt{xy}}} L_a^b(\gamma)$. Les courbes qui ont effectivement cette longueur seront par la suite importante, mais il est difficile de les calculer directement.

Or, le calcul des variations permet de calculer explicitement les courbes qui optimisent *localement* un second critère : $E_a^b(\gamma) = \int_a^b \|\dot{\gamma}(t)\|^2 \, dt$. Deux résultats permettent de faire le lien entre les deux critères. D'une part, on montre que, pour une courbe donnée, les trajectoires qui minimisent *localement* $E_a^b(\gamma)$ sont les trajectoires parcourues à vitesse constante. D'autre part, on montre aussi que si une courbe minimise *localement* $L_a^b(\gamma)$ sur $\Theta_{\mathtt{xy}}$, les trajectoires associées parcourues à vitesse constante minimisent *localement* $E_a^b(\gamma)$ sur $\Theta_{\mathtt{xy}}$.

Les trajectoires qui minimisent *localement* $E_a^b(\gamma)$ seront appelées géodésiques. D'après le calcul des variations, elles sont solutions - dans une carte locale donnée - d'une équation différentielle du deuxième ordre, qui fait intervenir les variations du tenseur métrique dans cette carte dues à la courbure de la surface. Ce résultat est important tant sur le plan pratique (il permet de calculer explicitement les géodésiques et donc les distances) que sur le plan théorique (il permet, comme on le verra dans la section suivante, de définir la carte exponentielle).

Les géodésiques qui minimisent *globalement* $L_a^b(\gamma)$ sur $\Theta_{\mathtt{xy}}$ seront appelées géodésiques minimisantes.

Prenons pour exemple la sphère unité dans $\mathbb{R}^n$, qui représente les normales. On montre facilement que les géodésiques suivent les grands cercles de la sphère. Elles peuvent effectuer plusieurs fois le tour de ces grands cercles. Les géodésiques minimisantes sont celles qui effectuent moins d'un demi-tour.

La variété est dite géodésiquement complète si le domaine de définition de toute géodésique peut-être étendu à $\mathbb{R}$. On peut par exemple prolonger une géodésique de la sphère en faisant un nombre infini de fois le tour du grand cercle qu'elle suit. Enlevons maintenant le pôle nord de notre sphère, les géodésiques suivent toujours les grands cercles, mais, dans le cas des méridiens, elles ne peuvent passer par le pôle manquant. Leur domaine de définition ne peut dépasser un tour, et ne peut donc pas être étendu à $\mathbb{R}$ (la vitesse est constante sur une géodésique). On constate en outre qu'il n'existe pas de géodésique minimisante entre deux points d'un même méridien situés de part et d'autre du pôle manquant. Ceci correspond en fait à un résultat général : lorsque la variété est géodésiquement complète, le théorème de Hopf-Rinow-De Rham assure l'existence d'une géodésique entre chaque points, et la variété est complète pour la distance. Nous supposerons cette propriété vérifiée dans la suite.

On a donc le résultat suivant :

Définition D.6 *Distance et géodésiques*

Soient $\mathtt{x}$ *et* $\mathtt{y}$ *deux points de* $\mathcal{M}$*, et* $\Theta_{\mathtt{xy}}$ *l'ensemble des trajectoires qui relient ces deux points.*

La distance entre ces deux points est : $d(\mathtt{x},\mathtt{y}) = \inf_{\gamma\in\Theta_{\mathtt{xy}}} L_a^b(\gamma)$ *où* $L_a^b(\gamma) = \int_a^b \|\dot{\gamma}(t)\|\, dt$.

Les trajectoires de $\Theta_{\mathtt{xy}}$ *qui minimisent localement* $E_a^b(\gamma) = \int_a^b \|\dot{\gamma}(t)\|^2\, dt$ *sont appelées géodésiques. La vitesse y est constante.*

Les trajectoires de $\Theta_{\mathtt{xy}}$ *qui minimisent globalement* $L_a^b(\gamma)$ *et sont parcourues à vitesse constantes sont des géodésiques particulières, appelées géodésiques minimisantes.*

Leur existence est assuré si $\mathcal{M}$ *est géodésiquement complète.*

Exemples

Espace affine Les géodésiques suivent les lignes droites, et la distance induite est la distance euclidienne.

Courbes Les géodésiques suivent la courbe à vitesse constante. La distance entre deux points correspond à la longueur de la courbe entre ces deux points.

Normales Comme nous l'avons dit, les géodésiques suivent les grands cercles de la sphère. Une géodésique minimisante est donc un arc d'un de ces cercles unitaires. Sa longueur est proportionnelle à l'angulation entre les deux extrémités. La distance intrinsèque entre normale est donc l'angulation :

$$d(\mathtt{n}_1,\mathtt{n}_2) = (\widehat{\mathtt{n}_1,\mathtt{n}_2}) \tag{D.5}$$

Rotations 3D Lorsque la variété différentielle et sa mesure sont invariantes vis à vis du groupe de transformation $\mathcal{G}$, et a elle-même une structure de groupe de Lie compact (ce qui est par exemple le cas des rotations), les géodésiques partant de l'origine peuvent être caractérisées autrement : elles correspondent aux sous-groupes à un paramètre, et vérifient :

$$\gamma(s+t) = \gamma(s) \circ \gamma(t) = \gamma(t) \circ \gamma(s)$$

Dans le cas des rotations, les sous-groupes à un paramètre sont les ensembles de rotations de même axe :

$$R(t) = R(n_0,t.\theta_0)$$

Elles sont minimisantes pour $-\pi < t.\theta_0 < \pi$.

La distance entre une rotation $R(n,\theta)$ et l'identité est donc l'angle de cette rotation :

$$d(R(n,\theta),Id) = |\theta| \qquad avec\,\theta \in [-\pi,\pi] \tag{D.6}$$

La distance entre deux rotations quelconque peux toujours être calculée en utilisant la propriété d'invariance pour ramener l'une d'elle à l'identité :

$$d(R_1,R_2) = d(\mathtt{f}_{\mathtt{R}_1}^{-1}(R_2),Id) = d(R_1^t.R_2,0) \tag{D.7}$$

D.1.4 Carte exponentielle et lieu de coupure

Nous allons ici mettre en évidence, pour toute primitive $\mathtt{x}$ de la variété, une carte locale particulière centrée en $\mathtt{x}$ et définie à partir du plan tangent $T_{\mathtt{x}}\mathcal{M}$, appelée carte exponentielle. Cette carte locale sera obtenue en “aplatissant” la variété sur le plan tangent, le long des géodésiques, et recouvrira la presque totalité de la variété, et pourra être étendue (en perdant éventuellement la propriété d'injectivité ou d'ouverture de l'espace de départ) à la variété entière.

Soit donc une primitive $\mathtt{x}$ de $\mathcal{M}$. Les géodésiques sont solutions d'une équation différentielle du second ordre dans une carte locale. Il existe donc une unique géodésique vérifiant $\gamma(0) = \mathtt{x}$ et $\dot{\gamma}(0) = \overrightarrow{v_{\mathtt{x}}}$ où $\overrightarrow{v_{\mathtt{x}}} \in T_{\mathtt{x}}\mathcal{M}$, i.e. partant de $\mathtt{x}$ avec une vitesse $\overrightarrow{v_x}$ donnée. Nous noterons cette géodésique $\gamma_{(\mathtt{x},\overrightarrow{v_{\mathtt{x}}})}$. Son domaine de définition peut-être étendu à $\mathbb{R}$, et le point $\gamma_{(\mathtt{x},\overrightarrow{v_{\mathtt{x}}})}(1)$ est parfaitement défini. Ceci permet de définir l'application suivante, pour $\mathtt{x}$ fixé :

$$\exp_{\mathtt{x}} \quad : \quad \begin{array}{ccc} T_{\mathtt{x}}\mathcal{M} & \rightarrow & \mathcal{M} \\ \overrightarrow{v_{\mathtt{x}}} & \mapsto & \gamma_{(\mathtt{x},\overrightarrow{v_{\mathtt{x}}})}(1) \end{array}$$

Comme il existe toujours une géodésique minimisante entre deux points de $\mathcal{M}$, cette application est surjective. Cependant, elle doit, pour définir une carte locale, être un difféomorphisme, donc être injective et définie sur un ouvert. Nous allons donc la restreindre au plus grand domaine de définition ouvert où elle est injective. Pour cela, nous devons définir le lieu de coupure.

Définition D.7 *Lieu de coupure*

Soit $\mathtt{x}$ un point de $\mathcal{M}$.

Considérons la géodésique $\gamma_{(\mathtt{x},\overrightarrow{v_{\mathtt{x}}})}$. Elle est minimisante jusqu'à un temps $0 < t_0 \leq +\infty$, et ne l'est plus jamais après. Si $t_0 < +\infty$, le point $\mathtt{y} = \exp_{\mathtt{x}}(t_0.\overrightarrow{v_{\mathtt{x}}})$ est appelé point de coupure, et le vecteur tangent $t_0.\overrightarrow{v_{\mathtt{x}}}$ est appelé point de coupure tangent.

On regroupera les points de coupure tangents dans lieu de coupure tangentiel $C(\mathtt{x})$, et les points de coupure dans le lieu de coupure $\mathtt{C}(\mathtt{x})$.

Ces deux ensembles sont de mesure nulle.

On peux de même regrouper les points du type $t.\overrightarrow{v_\mathtt{x}}$ avec $t < t_0$ dans le domaine $D(x)$, qui est donc ouvert. L'existence d'une géodésique minimisante entre tous points permet alors d'affirmer que la restriction de l'exponentielle à $D(x) \cup C(x)$ couvre la totalité de la variété. L'unicité des géodésiques minimisante permet d'affirmer que sa restriction à $D(x)$ est injective. Nous pouvons donc couvrir presque toute la variété de façon injective à partir de cet ouvert. On montre que la fonction ainsi définie est bel et bien un difféomorphisme et définissons ainsi la carte exponentielle, qui couvre presque toute la variété.

Définition D.8 *Carte exponentielle*

Soit $\mathtt{x}$ *un point de* $\mathcal{M}$. *On note* $D(\mathtt{x})$ *le domaine de* $T_\mathtt{x}\mathcal{M}$ *contenant* 0 *et délimité par le lieu de coupure* $C(\mathtt{x})$ *(exclu). L'application suivante définit alors une carte locale centrée en* $\mathtt{x}$ *:*

$$\exp_\mathtt{x} \; : \; \begin{array}{ccc} D(\mathtt{x}) \in T_\mathtt{x}\mathcal{M} & \rightarrow & \mathcal{M} \setminus \mathtt{C}(\mathtt{x}) \\ \overrightarrow{xy} & \mapsto & \exp_\mathtt{x}(\overrightarrow{xy}) = \gamma_{(\mathtt{x},\overrightarrow{xy})}(1) \end{array}$$

L'application inverse est notée $\log_\mathtt{x}$ *avec* $\log_\mathtt{x}(\mathtt{y}) = \overrightarrow{xy}$.

La géodésique suivie pour atteindre $\mathtt{y}$ depuis $\mathtt{x}$ est par définition parcourue pendant un temps unité à vitesse constante égale à $\|\overrightarrow{xy}\|$. Sa longueur est donc $\|\overrightarrow{xy}\|$. Hors cette longueur est par définition la distance $d(\mathtt{x},\mathtt{y})$ entre les deux extrémités. On a donc la propriété suivante :

Proposition D.1 *Norme dans la carte exponentielle*

La norme dans la carte exponentielle correspond à la distance dans la variété :

$$\|\overrightarrow{xy}\| = \|\log_\mathtt{x}(\mathtt{y})\| = d(\mathtt{x},\mathtt{y}) \tag{D.8}$$

Attention Cette égalité n'est vrai que pour la norme dans la carte, i.e. la distance au centre de la carte. Il n'existe en revanche aucune relation simple entre la distance entre deux points quelconques dans la carte et leur distance dans la variété.

Une conséquence est que les coordonnées du tenseur métrique dans la base naturelle à l'origine de la carte sont simples :

Proposition D.2 *Coordonnées du tenseur métrique dans la base naturelle de la carte exponentielle à l'origine*

Les coordonnées du tenseur métrique de l'espace tangent en $\mathtt{x}$ $T_\mathtt{x}\mathcal{M}$ *dans la base induite par la carte exponentielle centrée en* $\mathtt{x}$ *sont égales à l'identité :*

$$[G_{\exp_\mathtt{x}(\overrightarrow{0})\bullet\bullet}] = Id \tag{D.9}$$

Exemples

Espace affine La carte exponentielle centrée en $\mathtt{x}$ est la translation qui amène l'origine en $\mathtt{x}$. Elle couvre la totalité de l'espace (son lieu de coupure est à l'infini dans toutes les directions).

Courbes La carte exponentielle pour une courbe correspond à l'abscisse curviligne à partir d'un point donné (la géodésique, donc la courbe, est parcourue à vitesse constante). Bien entendu, l'espace tangent est une droite identifiée à la droite des réels, et on paramètre directement la carte à partir de $\mathbb{R}$. L'égalité entre coordonnée curviligne et distance sur une courbe n'est qu'un cas particulier de l'égalité entre norme dans la carte exponentielle et distance dans la variété ($\|\overrightarrow{xy}\| = d(\mathtt{x},\mathtt{y})$).

Normales Nous avons présenté trois cartes locales pour l'ensemble des normales (fin partie D.1.2.3). La première (avec la distance intrinsèque $\|\overrightarrow{on}\| = (\widehat{\mathtt{o},\mathtt{n}})$) est la carte exponentielle. Nous avons aussi expliqué à cette occasion comment calculer $\overrightarrow{on}$ à partir de $\mathtt{o}$ et $\mathtt{n}$, i.e. comment calculer le logarithme :

$$\overrightarrow{on} = \log_{\mathtt{o}}(\mathtt{n}) = (\widehat{\mathtt{o},\mathtt{n}}) \frac{\mathtt{n} - \mathtt{o}(\mathtt{n}|\mathtt{o})}{\|\mathtt{n} - \mathtt{o}(\mathtt{n}|\mathtt{o})\|} \tag{D.10}$$

Pour exprimer l'exponentiel, i.e. calculer $\mathtt{n}$ à partir de $\mathtt{o}$ et $\overrightarrow{on}$, il faut faire tourner la normale $\mathtt{o}$ d'un angle de $\|\overrightarrow{on}\| = (\widehat{\mathtt{o},\mathtt{n}})$, dans la direction indiquée par $\overrightarrow{on}$. Pour cela, il suffit d'appliquer à $\mathtt{o}$ la rotation d'axe $\mathtt{o} \wedge \overrightarrow{on}$ et d'angle $\|\overrightarrow{on}\|$. Étant donné que $\mathtt{o}$ est de norme unitaire et que $\mathtt{o}$ et $\overrightarrow{on}$ sont par construction perpendiculaires, le produit vectoriel $\mathtt{o} \wedge \overrightarrow{on}$ a effectivement pour norme $\|\overrightarrow{on}\|$. Finalement, il suffit d'appliquer à $\mathtt{o}$ la rotation de vecteur $\mathtt{o} \wedge \overrightarrow{on}$:

$$\mathtt{n} = \exp_{\mathtt{o}}(\overrightarrow{on}) = R_{\mathtt{o}\wedge\overrightarrow{on}}(\mathtt{o}) \tag{D.11}$$

Le lieu de coupure de cette carte exponentielle correspond à un quart (pour les normales non-orientées) ou un demi-tour (dans le cas orienté) sur la sphère, i.e. à $|\overrightarrow{on}| = \pi/2$ (l'ensemble des normales perpendiculaires à $\mathtt{o}$) pour les normales non-orienté et $|\overrightarrow{on}| = \pi$ (la normale $-\mathtt{o}$) pour le cas orienté.

Rotations 3D Nous n'avons présenté qu'une seule carte pour l'ensemble des rotations en 3D. Il s'agit effectivement de la carte exponentielle centrée autour de l'identité.

D.1.4.1 Cas invariant homogène : carte principale

Dans le cas invariant homogène, on peut toujours se ramener à l'origine, à l'aide des fonctions de placement. Nous n'utiliserons donc qu'une seule carte, la carte centrée autour de l'origine, appelée carte principale, et écrirons la quasi-totalité des calculs dans cette carte. Il faudra préalablement identifier les espaces tangents.

Lien entre les espaces tangents Nous allons ici montrer comment identifier l'espace tangent à l'origine $\mathtt{o}$ et l'espace tangent en un point $\mathtt{x}$ par le biais de la fonction de placement $\mathtt{f_x}$.

Soit une courbe γ avec $\gamma(0) = \mathtt{x}$. On peux écrire $\gamma(t) = \mathtt{f_x}(\gamma'(t))$ avec $\gamma'(0) = \mathtt{o}$. Dérivons par rapport à t en 0 : $\dot{\gamma}(0) = \left.\frac{d\mathtt{f_x}(\mathtt{e})}{d\mathtt{e}}\right|_{\mathtt{e}=\mathtt{o}} .\dot{\gamma}'(0)$. On peux ainsi faire le lien entre le vecteur tangent v_x dans $T_{\mathtt{x}}\mathcal{M}$ et v dans $T_{\mathtt{o}}\mathcal{M}$, en les considérant comme des vecteurs de E :

$$\begin{aligned} v_x &= J(\mathtt{f_x}).v \\ v &= J(\mathtt{f_x})^{-1}.v_{\mathtt{x}} \end{aligned} \qquad \text{avec } J(\mathtt{f_x}) = \left.\frac{d\mathtt{f_x}(\mathtt{e})}{d\mathtt{e}}\right|_{\mathtt{e}=\mathtt{o}}$$

On peut donc écrire quel doit être le lien entre les métriques des différents espaces tangents pour respecter la propriété d'invariance :

$$[G_{T_{\exp \mathtt{x}}\mathcal{M}}\bullet\bullet] = J(\mathtt{f_x})^{-t}.[G_{T_{\exp \mathtt{o}}\mathcal{M}}\bullet\bullet].J(\mathtt{f_x})^{-1}$$

Cette propriété doit en particulier être vérifiée pour le sous groupe d'isotropie $\mathcal{H}$ (section D.1.1) :

$$\forall \mathtt{h} \in \mathcal{H} \qquad [G_{T_{\exp \mathtt{o}}\mathcal{M}}\bullet\bullet] = J(\mathtt{h})^{-t}.[G_{T_{\exp \mathtt{o}}\mathcal{M}}\bullet\bullet].J(\mathtt{h})^{-1}$$

Carte principale Nous allons ici montrer comment travailler exclusivement dans la carte principale (qui est, rappelons le, la carte exponentielle centrée autour de l'origine de la variété). Les vecteurs dans cette carte seront simplement notés $\overrightarrow{x}$ (au lieu de $\overrightarrow{ox}$), et les fonctions associées seront simplement notées exp et log :

$$\overrightarrow{x} = \log(\mathtt{x})$$

$$\mathtt{x} = \exp(\overrightarrow{x})$$

On peux alors définir, dans la carte principale, l'action d'une transformation, et les fonctions de placements :

$$\mathtt{f} \star \overrightarrow{x} = \log(\mathtt{f} \star \mathtt{x}) \qquad \mathtt{f} \star \mathtt{x} = \mathtt{f} \star \exp(\overrightarrow{x})$$

$$\mathtt{f}_{\overrightarrow{x}} \star \overrightarrow{y} = \log(\mathtt{f_x} \star \exp(\overrightarrow{y}))$$

On peut de plus définir directement les vecteurs tangents par leurs coordonnées dans la base induite par la carte principale :

$$v_x = \sum_i \overrightarrow{v_{\overrightarrow{x}}}.\frac{\partial \exp}{\partial \overrightarrow{x}_i}(\overrightarrow{x}) \qquad v_{\mathtt{x}} \in T_{\mathtt{x}}\mathcal{M} \quad \overrightarrow{v_{\overrightarrow{x}}} \in T_{\overrightarrow{x}}\mathcal{M}$$

Le lien entre le vecteur tangent $\overrightarrow{v_{\vec{x}}}$ de $T_{\vec{x}}\mathcal{M}$ et le vecteur $\vec{v}$ de $T_0\mathcal{M}$ est alors :

$$\begin{aligned} \overrightarrow{v_{\vec{x}}} &= J(\mathtt{f}_{\vec{x}}).\vec{v} \\ \vec{v} &= J(\mathtt{f}_{\vec{x}})^{-1}.\overrightarrow{v_{\vec{x}}} \end{aligned} \qquad \text{avec } J(\mathtt{f}_{\vec{x}}) = \left.\frac{d\mathtt{f}_{\vec{x}}(\vec{e})}{d\vec{e}}\right|_{\vec{e}=0}$$

Et comme $[G_{\exp_{\vec{0}}\bullet\bullet}] = Id$, la propriété d'invariance de la métrique impose :

$$[G_{\exp_{\vec{\mathtt{x}}}\bullet\bullet}] = J(\mathtt{f}_{\vec{x}})^{-t}.[G_{\exp_{\vec{0}}\bullet\bullet}].J(\mathtt{f}_{\vec{x}})^{-1} = J(\mathtt{f}_{\vec{x}})^{-t}.J(\mathtt{f}_{\vec{x}})^{-1} \qquad \text{(D.12)}$$

Lien entre la carte principale et les autres cartes exponentielles On peut finalement faire le lien entre les coordonnées dans la carte principale et dans les autres cartes exponentielles :

$$\overrightarrow{xy} = \log_{\mathtt{x}}(\mathtt{y}) = J(\mathtt{f}_{\vec{x}}).(\mathtt{f}_{\vec{x}}^{-1} \star \vec{y})$$

On peut aussi écrire la distance dans une carte exponentielle quelconque, et, bien-sûr, dans la carte principale :

$$d(\mathtt{x},\mathtt{y}) = \overrightarrow{xy}^t.[G_{\exp_{\vec{\mathtt{x}}}\bullet\bullet}].\overrightarrow{xy} = (\mathtt{f}_{\vec{x}}^{-1} \star \vec{y})^t.(\mathtt{f}_{\vec{x}}^{-1} \star \vec{y})$$

Cas des transformations rigides elles-mêmes Dans le cas où $\mathcal{M} = \mathcal{G}$, il faut adapter les équations ci-dessus, en remarquant qu'une transformation est sa propre fonction de placement.

D.1.5 Forme volume et intégration d'une fonction réelle

Nous souhaitons maintenant pouvoir intégrer explicitement une fonction $f : \mathcal{M} \to \mathbb{R}$ intégrable. Le tenseur métrique de $T_{\mathtt{x}}\mathcal{M}$ permet de définir l'élément de volume infinitésimal (le raisonnement est analogue à celui permettant de calculer la norme de la dérivée) :

Proposition D.3 *Forme volume*

L'élément infinitésimal de volume au point $\mathtt{x}$ *de* $\mathcal{M}$ *est donné par :* $d\mathcal{M}(\mathtt{x}) = \sqrt{|[G_{T_{\mathtt{x}}\mathcal{M}}\bullet\bullet]|}\,d\mathtt{x}$.

Cette écriture est cependant peu pratique, et nous passerons par une carte locale π, la carte principale de préférence.

Soit donc une carte locale π. Nous supposerons pour simplifier qu'elle couvre la presque-totalité de la variété. Si notre fonction est donnée sous la forme $\vec{x} \to f(\pi(\vec{x}))$, on peut intégrer directement dans la carte, en utilisant la base naturelle. Il faudra cependant faire attention, car la carte ne couvre pas toujours la totalité de l'espace.

Proposition D.4 *Intégration d'une fonction réelle à l'aide d'une carte locale*

Soit π une carte locale couvrant presque toute la variété et une fonction $f : \mathcal{M} \to \mathbb{R}$ intégrable, d'intégrale nulle en dehors du domaine de la carte $\int_{\mathcal{M}/\mathtt{D}_\pi} f(\mathtt{x}).d\mathcal{M}(\mathtt{x}) = 0$.

On a :

$$\int_{\mathcal{M}} f(\mathtt{x}).d\mathcal{M}(\mathtt{x}) = \int_{D_\pi} f(\pi(\overrightarrow{x})).\sqrt{\left|[G_{\pi(\overrightarrow{x})\bullet\bullet}]\right|}.d\overrightarrow{ox} \tag{D.13}$$

Dans le cas homogène invariant, on a $[G_{\exp\overrightarrow{\mathtt{x}}\bullet\bullet}] = J(\mathtt{f}_{\overrightarrow{x}})^{-1}.J(\mathtt{f}_{\overrightarrow{x}})^{-t}$ et donc, en utilisant la carte principale :

$$\int_{\mathcal{M}} f(\mathtt{x}).d\mathcal{M}(\mathtt{x}) = \int_{D_\pi} \mathtt{f}(\exp(\overrightarrow{x})).\frac{d\overrightarrow{x}}{|J(\mathtt{f}_{\overrightarrow{x}})|} \tag{D.14}$$

Exemple avec les normales 3D Détaillons le cas de $\mathcal{P}_2$, ensemble des normales non orientées de $\mathbb{R}^3$. On utilisera la base canonique de $\mathbb{R}^3$, notée $\{i,j,k\}$.

Cet ensemble peut-être représenté par une demi-sphère de rayon unitaire. Nous fixons l'origine $\mathtt{o}$ au pôle, de coordonnées $(0,0,1)$, et le plan tangent en $\mathtt{o}$ est le plan horizontal passant par $\mathtt{o}$. Nous le munissons bien évidement de la base $\{i,j\}$.

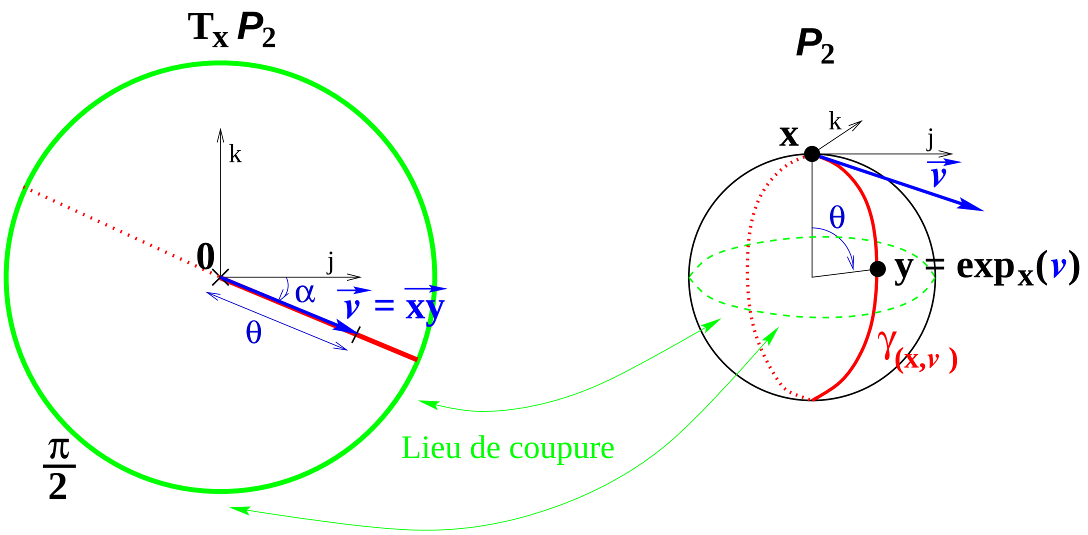
La carte principale est donc définie de la manière suivante :

$$\exp \quad : \quad \begin{array}{ccc} \{\overrightarrow{n} = (x_1,x_2)/x_1^2 + x_2^2 < \frac{\pi}{2}\} & \to & \mathcal{P}_2 \\ \overrightarrow{n} = (\theta\cos\alpha,\theta\sin\alpha) & \mapsto & \exp(\overrightarrow{n}) = (\sin\theta\,\cos\alpha, \sin\theta\,\sin\alpha, \cos\theta) \end{array}$$

Pour alléger les notations, on utilisera plutôt la carte équivalente (en coordonnées polaires) :

$$\mathtt{n} \quad : \quad \begin{array}{ccc} \overrightarrow{n} = (\theta,\alpha)/\theta \in [0,\pi/2[,\alpha \in]-\pi,\pi] & \to & \mathcal{P}_2 \\ \overrightarrow{n} = (\theta,\alpha) & \mapsto & (\sin\theta\,\cos\alpha, \sin\theta\,\sin\alpha, \cos\theta) \end{array}$$

Tout-ceci est illustré sur la figure suivante :



On peut alors calculer $\sqrt{\left|G_{\mathtt{n}(\theta,\alpha)\bullet\bullet}\right|}$:

$\frac{\partial \mathtt{n}}{\partial\theta}(\theta,\alpha) = (\cos\theta\,\cos\alpha, \cos\theta\,\sin\alpha, -\sin\theta)$ et $\frac{\partial \mathtt{n}}{\partial\alpha}(\theta,\alpha) = (-\sin\theta\,\sin\alpha, \sin\theta\,\cos\alpha, 0)$

$$[G_{\mathbf{n}(\theta,\alpha)\bullet\bullet}] = \begin{bmatrix} \cos^2\theta\,\cos^2\alpha + \cos^2\theta\,\sin^2\alpha + \sin^2\theta & \ldots \\ -\cos\theta\,\cos\alpha\sin\theta\,\sin\alpha + \cos\theta\,\sin\alpha\sin\theta\,\cos\alpha & \sin^2\theta\,\sin^2\alpha + \sin^2\theta\,\cos^2\alpha \end{bmatrix}$$

$$[G_{\mathbf{n}(\theta,\alpha)\bullet\bullet}] = \begin{bmatrix} 1 & 0 \\ 0 & \sin^2\theta \end{bmatrix} \text{ et donc}$$

$$\sqrt{\left|[G_{\mathbf{n}(\theta,\alpha)\bullet\bullet}]\right|} = \sin\theta \tag{D.15}$$

On peut alors intégrer sans problème les fonctions données sous la forme $f(\theta,\alpha)$.

D.2 Statistiques intrinsèques sur les primitive aléatoire

Nous pouvons maintenant construire la base du calcul des probabilités et des statistiques sur nos variétés à partir de deux éléments indépendants :

- La forme volume : la possibilité d'intégrer une fonction (ou une distribution) réelle définie à partir de l'ensemble des primitives permet de définir une densité de probabilité (à valeur réelles positive, d'intégrale unité), qui permettra de modéliser facilement une primitive aléatoire. Remarquons qu'on pourrait s'en passer, par exemple pour une primitive aléatoire n'ayant qu'un nombre fini de réalisation possibles car il suffit alors de connaître la probabilité de chacune (autrement dit, la densité de probabilité est une combinaison de diracs). Mais nous utiliserons quasi-systématiquement la densité par la suite.
- La distance intrinsèque : on pourra calculer l'espérance du carré de la distance entre une primitive aléatoire et une primitive fixe, et généraliser ainsi la notion de variance totale, puis de moyenne (le minimum de la variance totale, comme dans le cas vectoriel). On pourra aussi calculer une covariance (dans l'espace tangent), calculer l'information, et donc généraliser la notion de gaussienne (la loi qui minimise l'information pour une moyenne et une covariance donnée).

D.2.1 Primitive aléatoire

Le rôle de cette partie est de définir une primitive aléatoire, sa densité de probabilité, et de calculer l'espérance d'une observable, c'est à dire une variable aléatoire réelle définie comme l'image d'une primitive aléatoire par une fonction.

Il est important de noter que les définitions de cette partie reposent toutes directement sur la métrique utilisée. Ces définitions ne seront donc invariantes que lorsqu'on utilisera la métrique invariante.

D.2.1.1 Primitive aléatoire - application d'une fonction mesurable

Les notions classiques d'espace probabilisé, qui regroupe l'univers, la tribu d'évènements et la mesure de probabilité est très générales, et restent valables dans une variété différentielle. Ainsi, pour une primitive aléatoire, l'univers (i.e. l'ensemble des évènements élémentaires, i.e. des réalisations possibles de la primitive) est l'ensemble des primitives $\mathcal{M}$. On utilisera la tribu Borélienne $\mathbf{F}$. On définira enfin une mesure de probabilité, i.e. une fonction σ-additive de $\mathbf{F}$ dans [0,1] exprimant, pour tout événement complexe $A \in \mathbf{F}$, sa probabilité, i.e. la probabilité de trouver la primitive aléatoire $\mathbf{x}$ dans A :

$$Pr_{\mathbf{x}}(A) = Pr(\mathbf{x} \in A)$$

Rappelons qu'on peut, à l'aide d'une fonction mesurable f définie de l'univers $\mathcal{M}$ dans un autre ensemble B, associer les réalisations possibles de la primitive aléatoire $\mathbf{x}$ définie dans $\mathcal{M}$ et les réalisations possibles de $f(\mathbf{x})$. En veillant à utiliser les tribus adéquates, on définit ainsi une variable aléatoire $\mathbf{y} = f(\mathbf{x})$ et sa mesure de probabilité est donnée par :

$$Pr_{\mathbf{y}}(B) = Pr_{\mathbf{x}}(f^{-1}(B))$$

D.2.1.2 Densité de probabilité

La mesure de probabilité pourrait suffire pour définir l'espérance d'une observable. Il est cependant peu commode de travailler avec cette mesure car il s'agit d'une fonction réelle de la tribu (i.e. un ensemble de sous parties de l'univers). Il serait bien plus pratique de travailler, comme dans le cas vectoriel avec une densité de probabilité, i.e. une fonction réelle de l'univers qui synthétise l'information de la mesure de probabilité.

Nous supposerons donc, comme dans le cas vectoriel, que nos primitives aléatoires sont représentées par une densité de probabilité $p_{\mathbf{x}}(\mathtt{y})$ qui vérifie :

$$Pr(\mathbf{x} \in A) = \int_A p_{\mathbf{x}}(\mathtt{y}).d\mathcal{M}(\mathtt{y})$$

soit $dPr(\mathtt{y}) = p_{\mathbf{x}}(\mathtt{y}).d\mathcal{M}(\mathtt{y})$. La contrainte de normalisation impose : $\int_{\mathcal{M}} p_{\mathbf{x}}(\mathtt{y}).d\mathcal{M}(\mathtt{y}) = 1$.

Nous avons parlé de la densité de probabilité comme d'une fonction. Mais il pourra s'agir plus généralement d'une distribution, qui permet de représenter dans le même cadre les densités volumiques et les concentrations linéiques, surfaciques ou ponctuelles. Attention cependant, car ces distributions peuvent poser un certain nombre de problèmes, par exemple pour l'intégration dans une carte.

Exemple : densité uniforme La loi uniforme sur un sous-ensemble $X \subset \mathcal{M}$ de volume fini a une densité donnée par : $p_{\mathbf{x}}(\mathtt{y}) = \frac{\mathbb{1}_X(\mathtt{y})}{\int_X d\mathcal{M}}$.

Exemple : densité ponctuelle Si une variable aléatoire n'a qu'un nombre fini de réalisations possibles $\{\mathtt{x_i}\}$, on peut la représenter directement en utilisant la probabilité de ces réalisations $Pr(\mathbf{x} = \mathtt{x_i})$, ou utiliser une densité de probabilité qui est une mixture de diracs : $p_{\mathbf{x}}(\mathtt{y}) = \sum_i Pr(\mathbf{x} = \mathtt{x_i}).\delta_{\mathtt{x_i}}(\mathtt{y})$. Les intégrations deviendront alors des sommes pondérées, comme ici avec la contrainte de normalisation : $\int_{\mathcal{M}} p_{\mathbf{x}}(\mathtt{y}).d\mathcal{M}(\mathtt{y}) = \sum_i Pr(\mathbf{x} = \mathtt{x_i}) = 1$.

D.2.1.3 Espérance d'une observable

L'espérance d'une variable aléatoire définie comme l'image d'une primitive aléatoire par une fonction φ de $\mathcal{M}$ dans $\mathbb{R}$ (appelée observable en référence à la physique quantique) peut être calculée à l'aide de la densité de probabilité, comme dans le cas vectoriel :

$$E_{\mathbf{x}}[\varphi] = \int_{\mathcal{M}} \varphi(\mathtt{y}).p_{\mathbf{x}}(\mathtt{y}).d\mathcal{M}(\mathtt{y})$$

Nous avons déjà expliqué comment utiliser une carte locale pour calculer de telles intégrales (voir eq. D.13), lorsque l'intégrale est nulle en dehors du domaine de la carte $\int_{\mathcal{M}/\mathtt{D}_\pi} f(\mathtt{x}).d\mathcal{M}(\mathtt{x}) = 0$ (en particulier avec des distributions non nulles sur le lieu de coupure lorsqu'on utilise la carte exponentielle). On a dans ce cas :

$$E_{\mathbf{x}}[\varphi] = \int_{D_\pi} \varphi(\overrightarrow{x}).p_{\mathbf{x}}(\pi(\overrightarrow{x})).\sqrt{\left|[G_{\pi(\overrightarrow{x})\bullet\bullet}]\right|}.d\overrightarrow{x}$$

Et en particulier (c'est la contrainte de normalisation) :

$$E_{\mathbf{x}}[1] = \int_{D_\pi} p_{\mathbf{x}}(\pi(\overrightarrow{x})).\sqrt{\left|[G_{\pi(\overrightarrow{x})\bullet\bullet}]\right|}.d\overrightarrow{x} = 1$$

On peut même définir directement la densité de probabilité de $\mathbf{x}$ dans la carte locale :

$$p_{\overrightarrow{\mathbf{x}}}(\overrightarrow{x}) = p_{\mathbf{x}}(\mathtt{x}).\sqrt{\left|[G_{\pi(\overrightarrow{x})\bullet\bullet}]\right|} \tag{D.16}$$

Si bien que :

$$E_{\mathbf{x}}[\varphi] = \int_{D_\pi} \varphi(\overrightarrow{x}).p_{\overrightarrow{\mathbf{x}}}(\overrightarrow{x}).d\overrightarrow{x}$$

$$E_{\mathbf{x}}[1] = \int_{D_\pi} p_{\overrightarrow{\mathbf{x}}}(\overrightarrow{x}).d\overrightarrow{x} = 1$$

On peut facilement généraliser ces résultats pour une observable vectorielle, en isolant les coordonnées.

Propriétés On vérifie que l'espérance d'une observable est, comme dans le cas vectoriel, linéaire :

$$E_{\mathbf{x}}[\alpha.\varphi] = \alpha.E_{\mathbf{x}}[\varphi]$$

$$E_{\mathbf{x}}[\varphi_1 + \varphi_2] = E_{\mathbf{x}}[\varphi_1] + E_{\mathbf{x}}[\varphi_2]$$

Carte principale Dans le cas invariant homogène, on peut écrire les formules dans la carte principale (voir eq. D.14) :

$$p_{\overrightarrow{\mathbf{x}}}(\overrightarrow{x}) = \frac{p_{\mathbf{x}}(\mathtt{x})}{|J(\mathtt{f}_{\overrightarrow{x}})|}$$

$$E[\varphi(\mathbf{x})] = \int_{D_\pi} \varphi(\overrightarrow{x}).p_{\overrightarrow{\mathbf{x}}}(\overrightarrow{x}).d\overrightarrow{x}$$

$$Pr(\mathbf{x} \in \mathcal{M}) = \int_{D_\pi} p_{\overrightarrow{\mathbf{x}}}(\overrightarrow{x}).d\overrightarrow{x} = 1$$

Exemple : normale aléatoire uniforme en 3D La demi-sphère des normales non-orientées est de volume (en l'occurrence surface) fini. On peut donc définir une normale aléatoire uniforme sur la totalité de l'espace des normales.

Sa densité de probabilité sera de la forme $p_{\mathbf{n}}(n) = k$. Nous travaillerons dans la carte habituelle (la carte exponentielle en coordonnées polaires), où cette densité prend la forme $p_{\theta,\alpha}(\theta,\alpha) = k.\sin\theta$ (voir eq. D.16 et D.15), on peut alors calculer la constante de normalisation :

$$\int_{\theta=0}^{\frac{\pi}{2}} \int_{\alpha=-\pi}^{\pi} p_{\theta,\alpha}(\theta,\alpha)\, d\theta\, d\alpha = 1$$

$$\int_{\theta=0}^{\frac{\pi}{2}} \int_{\alpha=-\pi}^{\pi} k.\sin\theta\, d\theta\, d\alpha = k.2\pi.\int_{\theta=0}^{\frac{\pi}{2}} \sin\theta\, d\theta = k.2\pi = 1$$

Finalement : $k = 1/(2\pi)$.

On peut aussi calculer sa variance totale (que nous définirons dans la section suivante) par rapport à la normale correspondant à $\theta = 0$. Il s'agit de l'espérance de l'observable θ^2 :

$$E\left[\theta^2\right] = \frac{1}{2\pi} \int_{\theta=0}^{\frac{\pi}{2}} \int_{\alpha=-\pi}^{\pi} \theta^2 \sin\theta\, d\theta\, d\alpha = \int_{\theta=0}^{\frac{\pi}{2}} \theta^2 \sin\theta\, d\theta$$

$$= \left[-x^2\cos x + 2x.\sin x + 2\cos x\right]_0^{\pi/2} = \pi - 2 \quad \text{(D.17)}$$

une normale uniforme a donc un écart-type total (non-infini !) de $\sqrt{\pi - 2} = 61.2\, deg$.

Algorithme : tirage d'une normale aléatoire uniforme en nD Pour obtenir une normale aléatoire uniforme, la méthode la plus simple consiste à effectuer un tirage de vecteur uniforme à l'intérieur d'un volume à symétrie sphérique (comme une boule), et normaliser ensuite ce vecteur. Les directions étant réparties uniformément dans le volume, les normales ainsi obtenues le seront aussi.

Reste donc à tirer un vecteur aléatoire dans ce volume. Pour cela, il suffit de tirer un vecteur aléatoire dans n'importe quel ensemble contenant ce volume, et rejeter les vecteurs situés à l'extérieur.

L'ensemble où il sera le plus simple de tirer un vecteur aléatoire est le cube, car on peut tirer les coordonnées facilement (il s'agit d'un tirage uniforme) et indépendamment les unes des autres. Le volume à symétrie sphérique le plus simple sera la boule inscrite dans ce cube auquel on aura retiré une petite boule concentrique (pour éviter les divisions par une norme trop faible lors de la normalisation).

L'algorithme est donc le suivant :

Faire

. tirer un vecteur aléatoire uniforme : $x = (UniformRnd([-1,1]),UniformRnd([-1,1]),...)$

. calculer sa norme au carré : $r^2 = \sum_i x_i^2$

Jusqu'à $(r^2 > 0.01)$ et $(r^2 < 1)$

Normaliser $n = x/r$

D.2.2 Espérance et Covariance

Nous allons ici voir comment généraliser les notions statistiques de bases aux primitives. Nous nous baserons pour cela sur l'espérance d'une observable.

D.2.2.1 Espérance au sens de Fréchet

Dans un espace vectoriel, la moyenne possède deux définitions équivalentes :

- On peux définir la moyenne comme la somme des différentes réalisations possibles pondérées par leurs probabilités : cette définition a l'avantage d'être explicite, donc de donner une méthode de calcul directe, mais s'appuie sur la notion de combinaison linéaire (i.e. de somme pondérée), qui est propre aux espaces vectoriels.
- On peux aussi définir la moyenne comme le vecteur minimisant la variance totale $E\left[\|\mathbf{y}-\mathbf{x}\|^2\right] = E\left[d(\mathbf{y},\mathbf{x})^2\right]$. Cette définition a l'avantage d'être implicite et de ne s'appuyer que sur l'espérance d'une observable.

Fréchet a proposé de généraliser directement cette seconde définition de la moyenne (rebaptisée espérance pour éviter les confusions) à un espace métrique quelconque de la façon suivante :

Définition D.9 *Espérance au sens de Fréchet*

Soit $\mathbf{x}$ *une primitive aléatoire de* $\mathcal{M}$*.*

La variance totale est l'espérance du carré de la distance entre la primitive aléatoire et une primitive fixe $\mathtt{y}$ *:*

$$E\left[d(\mathtt{y},\mathbf{x})^2\right] = E\left[\|\overrightarrow{xy}\|^2\right] \tag{D.18}$$

L'espérance au sens de Fréchet est l'ensemble des primitives qui minimisent la variance totale : $\mathbb{E}[\mathbf{x}] = argmin_{\mathtt{y}} E\left[d(\mathtt{y},\mathbf{x})^2\right]$*. Une telle primitive est notée* $\overline{\mathbf{x}}$*.*

Important Il est important de remarquer que, contrairement au cas vectoriel, il n'y a plus unicité de l'espérance. Par exemple, la sphère des normales étant de volume fini, on peut définir une loi uniforme sur sa totalité. La variance est alors la même vis-à-vis de tous les points de la sphère, et l'espérance au sens de Fréchet rassemble toute la sphère.

Généralisation : Primitive centrale à l'ordre α On peut généraliser cette définition aux différentes primitives centrales : les primitives centrales d'ordre α sont les primitives qui minimisent la variance totale à l'ordre α :

$$\sigma_{\mathbf{x}}^{\alpha}(\mathtt{y}) = E\left[d(\mathtt{y},\mathbf{x})^{\alpha}\right]$$

Les primitives centrales d'ordre 0 sont les modes de la variable aléatoire (i.e. les points de densité de probabilité maximale), les primitives d'ordre 1 correspondent à la notion de médiane.

D.2.2.2 Espérance au sens de Karcher

L'espérance au sens de Fréchet correspond à un minimum global. Il est souvent plus simple de gérer un minimum local, car on peut alors le caractériser par une différentielle nulle. On définit ainsi l'espérance au sens de Karcher :

Définition D.10 *Espérance au sens de Karcher*

Soit $\mathbf{x}$ *une primitive aléatoire de* $\mathcal{M}$*.*

L'espérance au sens de Karcher est l'ensemble des primitives qui minimisent localement *la variance totale* $E\left[d(\mathtt{y},\mathbf{x})^2\right]$*.*

Elle contient donc l'espérance au sens de Fréchet.

D.2.2.3 Propriété et recherche de l'espérance au sens de Karcher

L'espérance de Karcher a été définie comme minimisant localement la variance totale. Elle peut donc être caractérisée par une dérivé nulle par rapport à la primitive $\mathtt{y}$. Nous le ferons dans la carte principale : on dérivera la variance totale, sous sa forme $E_{\mathbf{x}}\left[\|\overrightarrow{xy}\|^2\right]$, et on l'annulera.

Commençons par la dérivé, dans la carte locale, de la distance $\|\overrightarrow{xy}\|^2$:

$$\frac{\partial \|\overrightarrow{xy}\|^2}{\partial \overrightarrow{y}} = 2.[G_{\exp \overrightarrow{y} \bullet\bullet}].\overrightarrow{xy} = -2.[G_{\exp \overrightarrow{y} \bullet\bullet}].\overrightarrow{yx} = -2.J(\mathtt{f}_{\overrightarrow{y}})^{-t}.(\mathtt{f}_{\overrightarrow{y}}^{-1} \star \overrightarrow{x})$$

On montre (difficilement [Pennec,]) que l'on peut alors écrire :

$$\frac{\partial E_{\mathbf{x}}\left[\|\overrightarrow{xy}\|^2\right]}{\partial \overrightarrow{y}} = E_{\mathbf{x}}\left[\frac{\partial \|\overrightarrow{xy}\|^2}{\partial \overrightarrow{y}}\right]$$
$$= -2.[G_{\exp \overrightarrow{y} \bullet\bullet}].E_{\mathbf{x}}\left[\overrightarrow{yx}\right] = -2.J(\mathtt{f}_{\overrightarrow{y}})^{-t}.E_{\mathbf{x}}\left[(\mathtt{f}_{\overrightarrow{y}}^{-1} \star \overrightarrow{x})\right]$$

Cette dérivé doit être nulle en toute primitive $\overrightarrow{\overline{x}} = \log \overline{\mathtt{x}}$ appartenant à l'espérance de Karcher. $J(\mathtt{f}_{\overrightarrow{y}})^{-t}$ étant de rang maximum, il faut et il suffit pour cela que $E_{\mathbf{x}}\left[(\mathtt{f}_{\overrightarrow{y}}^{-1} \star \overrightarrow{x})\right]$ soit nul :

Proposition D.5 *Caractérisation de l'espérance de Fréchet*

Une primitive faisant partie de l'espérance de Karcher (et donc de l'espérance de Fréchet) vérifie :

$$E_{\mathbf{x}}\left[\overrightarrow{\overline{x}x}\right] = E_{\mathbf{x}}\left[\mathtt{f}_{\overrightarrow{\overline{x}}}^{-1} \star \overrightarrow{x}\right] = 0 \tag{D.19}$$

Cette condition est nécessaire, mais pas suffisante.

On peut rechercher une de ces primitives en faisant une descente de gradient de type Newton sur le critère. Nous savons pour cela que, à partir d'une estimation $\overrightarrow{y}$ de l'espérance, nous devons aller dans la direction inverse du gradient de la variance $\Phi = \frac{\partial \sigma_{\mathbf{x}}^2(\exp(\overrightarrow{y}))}{\partial \overrightarrow{y}}$ pour annuler celle-ci. Une façon de la pondérer est d'utiliser l'inverse de la matrice hessienne $H = \frac{\partial \Phi(\exp(\overrightarrow{y}))}{\partial \overrightarrow{y}}$. En effet l'approximation au premier ordre de Φ est donnée par $\Phi(\overrightarrow{y} + \overrightarrow{\delta y}) = \Phi(\overrightarrow{y}) + H.\overrightarrow{\delta y} + O(\|\overrightarrow{y}\|^2)$, et peut être annulée avec $\overrightarrow{\delta y} = -H^{-1}.\Phi$.

La matrice Hessienne peut ici être approximée par $-2.J(\mathtt{f}_{\overrightarrow{y}})^{-t}.J(\mathtt{f}_{\overrightarrow{y}})^{-1}$ (la distance est de la forme $x^t.x$, sa dérivé de la forme $2.x^t.\dot{x}$, et sa dérivé seconde doit donc être de la forme $2.\dot{x}^t.\dot{x} + 2.x^t.\ddot{x} \simeq 2.\dot{x}^t.\dot{x}$). Finalement, on peut mettre à jour l'espérance en la déplaçant de $\overrightarrow{\delta y} = -H^{-1}.\Phi = J(\mathtt{f}_{\overrightarrow{y}}).E_{\mathbf{x}}\left[(\mathtt{f}_{\overrightarrow{y}}^{-1} \star \overrightarrow{x})\right]$, soit $\overrightarrow{y}' = \mathtt{f}_{\overrightarrow{y}} \star E_{\mathbf{x}}\left[(\mathtt{f}_{\overrightarrow{y}}^{-1} \star \overrightarrow{x})\right]$. Autrement dit, la correction apportée à $\overrightarrow{y}$ correspond à la moyenne dans la carte exponentielle centrée en $\overrightarrow{y}$.

Algorithme : Calcul de l'espérance de Karcher par descente de gradient L'algorithme est donc le suivant :

- Initialisation : $\overrightarrow{y}^0$
- A l'itération I : on calcule la moyenne $E_{\mathbf{x}}\left[(\mathtt{f}_{\overrightarrow{y}^I}^{-1} \star \overrightarrow{x})\right]$ dans la carte exponentielle centrée en $\overrightarrow{y}^I$, et on en déduit la nouvelle estimée : $\overrightarrow{y}^{I+1} = \mathtt{f}_{\overrightarrow{y}^I} \star (E_{\mathbf{x}}\left[(\mathtt{f}_{\overrightarrow{y}^I}^{-1} \star \overrightarrow{x})\right])$

– Jusqu'à convergence : $d(\overrightarrow{y}^{I+1},\overrightarrow{y}^{I}) < \varepsilon$

On pourra de plus calculer la matrice de covariance sur le résultat obtenu.

D.2.2.4 Matrice de covariance

La matrice de covariance d'une primitive aléatoire par rapport à une primitive fixe $\mathtt{y}$ est simplement définie en utilisant la covariance de $\overrightarrow{yx}$ (qui est une observable vectorielle) définie dans la carte exponentielle centrée en $\mathtt{y}$:

Définition D.11 *Matrice de covariance*

Soit $\mathbf{x}$ *une primitive aléatoire de* $\mathcal{M}$.

Sa matrice de covariance par rapport à une primitive fixe $\mathtt{y}$ *est définie dans* $T_{\mathtt{y}}\mathcal{M}$ *par :*

$$\Sigma_{\mathtt{xx}}(\mathtt{y}) = E\left[\overrightarrow{y\mathbf{x}} \otimes \overrightarrow{y\mathbf{x}}\right] = E\left[\overrightarrow{y\mathbf{x}}.\overrightarrow{y\mathbf{x}}^t\right] \tag{D.20}$$

La matrice de covariance par rapport à l'espérance $\overline{\mathtt{x}}$ (lorsque celle-ci est unique) est notée $\Sigma_{\mathtt{xx}} = \Sigma_{\mathtt{xx}}(\overline{\mathtt{x}})$.

Propriété En écrivant la trace de la matrice de covariance $Tr(\Sigma_{\mathtt{xx}}(\mathtt{y})) = Tr\left(E\left[\overrightarrow{y\mathbf{x}}.\overrightarrow{y\mathbf{x}}^t\right]\right) = E\left[Tr(\overrightarrow{y\mathbf{x}}.\overrightarrow{y\mathbf{x}}^t)\right] = E\left[\|\overrightarrow{xy}\|^2\right]$, et en se rappelant des équations D.8 et D.18, on constate que le lien entre la matrice de covariance et la variance totale est analogue au cas vectoriel :

$$E\left[d(\mathtt{y},\mathbf{x})^2\right] = Tr(\Sigma_{\mathtt{xx}}(\mathtt{y}))$$

Cas invariant On peut se ramener à la carte principale :

$$\Sigma_{\mathtt{xx}}(\mathtt{y}) = J(\mathtt{f}_{\overrightarrow{y}}).E\left[(\mathtt{f}_{\overrightarrow{y}}^{-1} \star \overrightarrow{x}).(\mathtt{f}_{\overrightarrow{y}}^{-1} \star \overrightarrow{x})^t\right].J(\mathtt{f}_{\overrightarrow{y}})^t$$

D.2.2.5 Propagation de l'espérance et de la matrice de covariance dans le cas invariant

Dans le cas où on utilise la mesure invariante, on peut calculer la densité de probabilité d'une variable aléatoire définie par l'action d'une transformation rigide fixe $\mathtt{f}$ sur une autre variable aléatoire : si $\mathbf{y} = \mathtt{f} \star \mathbf{x}$, $p_{\mathbf{y}}(\mathtt{y}) = p_{\mathbf{x}}(\mathtt{f}^{-1} \star \mathtt{y})$.

On vérifie donc, et c'est heureux, que, pour une transformation rigide fixe $\mathtt{f}$, l'espérance de la transformée est bien la transformé de l'espérance :

$$\mathbb{E}[\mathtt{f} \star \mathbf{x}] = \mathtt{f} \star \mathbb{E}[\mathbf{x}]$$

La matrice de covariance se propage en utilisant les formules permettant de passer d'un espace tangent à l'autre :

$$\mathbf{y} = \mathtt{f} \star \mathbf{x} \quad \Rightarrow \quad \Sigma_{\mathtt{yy}} = J(\mathtt{f}).\Sigma_{\mathtt{xx}}.J(\mathtt{f})^t$$

Il n'existe en revanche aucun lien simple dans le cas où la transformation est elle aussi aléatoire.

D.2.3 Approximation par une loi Gaussienne et distance de Mahalanobis

Dans le cadre d'une implémentation informatique des primitives aléatoires, il est bien évident que l'on ne peut toujours stocker la densité de probabilité. On se limitera souvent à une approximation au deuxième ordre, en stockant l'espérance et la covariance de la primitive aléatoire.

Tout ceci rappelle les approximations gaussiennes dans le cas vectoriel, qui sont utilisées car elles maximisent l'entropie (et minimisent donc l'information) pour une moyenne et une covariance donnée. Nous allons donc généraliser la notion de gaussienne en généralisant la notion d'entropie.

D.2.3.1 Entropie d'une primitive aléatoire

Nous généralisons directement la définition dans le cas vectoriel : l'entropie d'une primitive aléatoire est définie par :

$$H[\mathbf{x}] = -E_{\mathbf{x}}\left[\log p_{\mathbf{x}}(\mathtt{x})\right]$$

On vérifie que la loi qui maximise l'entropie est la loi uniforme.

On vérifie aussi que l'entropie est invariante : pour une transformation déterministe $\mathtt{f}$, $H[\mathtt{f} \star \mathbf{x}] = H[\mathbf{x}]$.

D.2.3.2 Lois gaussiennes

Les lois gaussiennes sont, dans le cas vectoriel, les lois qui maximisent l'entropie pour une moyenne et une covariance imposée. Nous conservons la même définition pour les primitives. On montre alors qu'elles sont de la même forme que dans le cas vectoriel :

Proposition D.6 *Lois gaussiennes*

La loi gaussienne d'espérance $\overline{\mathtt{x}}$ et de covariance Σ est définie par la densité de probabilité de la forme :

$$N_{(\overline{\mathtt{x}},\Gamma)}(\mathtt{y}) = \frac{e^{-\frac{\overrightarrow{\overline{\mathtt{x}}\mathtt{y}}^t.\Delta^{-1}.\overrightarrow{\overline{\mathtt{x}}\mathtt{y}}}{2}}}{k} \tag{D.21}$$

où k (constante de normalisation) est solution de

$$k = \int_{\mathcal{M}} e^{-\frac{\overrightarrow{\bar{\mathbf{x}}\mathbf{y}}^t.\Delta^{-1}.\overrightarrow{\bar{\mathbf{x}}\mathbf{y}}}{2}}.d\mathcal{M}(\mathbf{y})$$

et Δ (matrice de dispersion) est solution de

$$\Sigma = \frac{\int_{\mathcal{M}} \overrightarrow{\bar{\mathbf{x}}\mathbf{y}}.\overrightarrow{\bar{\mathbf{x}}\mathbf{y}}^t.e^{-\frac{\overrightarrow{\bar{\mathbf{x}}\mathbf{y}}^t.\Delta^{-1}.\overrightarrow{\bar{\mathbf{x}}\mathbf{y}}}{2}}.d\mathcal{M}(\mathbf{y})}{k}$$

Dans le cas où notre variété différentielle est de volume fini $V = \int_{\mathcal{M}} d\mathcal{M}$, on peut utiliser une matrice de dispersion infinie. Ainsi, la loi gaussienne de matrice de dispersion d'inverse nulle (centrée autour d'une espérance arbitraire) n'est autre que la loi uniforme. La loi uniforme est donc un cas particulier de loi gaussienne.

Il est important de noter que la constante de normalisation et la matrice de covariance ne sont pas - comme dans le cas d'un espace vectoriel - simplement reliées à la matrice de dispersion. On montre cependant que dans le cas d'une *faible covariance*, les formules du cas vectoriel constituent de bonnes approximations :

$$\Sigma\,\text{faible} \quad \Rightarrow \quad \begin{array}{c} \Delta \simeq \Sigma \\ k \simeq (2\pi)^{(\frac{\mathfrak{D}}{2})}.\sqrt{|G_{T_{\bar{\mathbf{x}}}\mathcal{M}}|.\Sigma} \end{array} \tag{D.22}$$

On peut se demander alors ce qu'il se passe lorsque l'on ajoute une erreur gaussienne centrée à une primitive aléatoire qui suit déjà une loi gaussienne. Dans le cas vectoriel, ceci revient à convoluer les deux lois, et le résultat est une loi gaussienne dont la covariance est la somme des deux covariances. Nous pensons que dans le cas des primitives on obtient le même résultat, à ceci près que ce sont les matrices de dispersions qui s'additionnent.

Pour bien distinguer variances et dispersions, rappelons que nous utilisons le vocabulaire suivant dans ce manuscrit :

- Covariance : La covariance $\Sigma_{\mathbf{xx}} = E_{\mathbf{x}}\left[\overrightarrow{\bar{x}x}.\overrightarrow{\bar{x}y}^t\right]$
- Variance totale : La trace de la covariance, i.e. l'espérance de la distance carré : $Tr(\Sigma_{\mathbf{xx}}) = E\left[d(\bar{x},x)^2\right]$
- Variance : La variance par direction, qui correspond au paramètre $\sigma_{\mathbf{x}}^2$ pour une covariance isotrope : $\Sigma_{\mathbf{xx}} = \sigma_{\mathbf{x}}^2.Id$. La variance totale est alors donnée par $Tr(\Sigma_{\mathbf{xx}}) = \mathfrak{D}.\sigma_{\mathbf{x}}^2$
- Écart-type : La racine carré de la variance $\sigma_{\mathbf{x}} = \sqrt{\sigma_{\mathbf{x}}^2}$
- Matrice de dispersion : La matrice $\Delta_{\mathbf{xx}}$ qui apparaît dans une loi gaussienne.
- Dispersion totale : La trace de la matrice de dispersion $Tr(\Delta_{\mathbf{xx}})$.
- Dispersion : La dispersion par direction, qui correspond au paramètre $\delta_{\mathbf{x}}^2$ pour une matrice de dispersion isotrope : $\Delta_{\mathbf{xx}} = \delta_{\mathbf{x}}^2.Id$. La dispersion totale est alors donnée

par $Tr(\Delta_{\mathbf{xx}}) = \mathcal{D}.\delta^2_{\mathbf{x}}$

– Dispersion-type : La racine carré de la dispersion $\delta_{\mathbf{x}} = \sqrt{\delta^2_{\mathbf{x}}}$

D.2.3.3 Distance de Mahalanobis

Rappelons ici que la distance de Mahalanobis simple est définie, dans le cas vectoriel par $\mu^2(\mathbf{x},y) = \overrightarrow{\overline{x}y}^t.\Sigma^{-1}_{xx}.\overrightarrow{\overline{x}y}$. On peut alors se poser la question de savoir si on utilise la matrice de dispersion Δ où la matrice de covariance Σ pour la généraliser.

Deux arguments permettrons de nous décider pour ce dernier cas :

– Nous pouvons toujours calculer la covariance d'une primitive aléatoire, même si elle ne suit pas une loi gaussienne. La définition à partir de la matrice de covariance ne repose donc pas sur ces lois. En revanche, une primitive aléatoire non-gaussienne n'a pas de matrice de dispersion associée. On pourrait envisager d'utiliser la matrice de dispersion de la gaussienne qui possède la même moyenne et la même covariance, faisant ainsi jouer un rôle particulier à ces lois. Les calculs serait de plus compliqués.

– Nous verrons que, en utilisant la matrice de covariance, l'espérance de la distance de Mahalanobis entre une primitive aléatoire quelconque et sa moyenne est égale à la dimension de l'espace, comme dans le cas vectoriel.

Nous définirons donc la distance de Mahalanobis simple ainsi :

$$\mu^2(\mathbf{x},\mathtt{y}) = \overrightarrow{\overline{\mathtt{x}}\mathtt{y}}^t.\Sigma^{-1}_{\mathtt{xx}}.\overrightarrow{\overline{\mathtt{x}}\mathtt{y}} \tag{D.23}$$

Propriétés La distance de Mahalanobis est invariante : pour une transformation rigide $\mathtt{f}$ fixée :

$$\mu^2(\mathtt{f} \star \mathbf{x},\mathtt{f} \star \mathtt{y}) = \mu^2(\mathbf{x},\mathtt{y})$$

On peut donc se ramener à la carte principale :

$$\mu^2(\mathbf{x},\mathtt{y}) = \mu^2(\mathtt{f}^{-1}_{\mathtt{y}} \star \mathbf{x},0)$$

Calculons maintenant l'espérance de la distance de Mahalanobis entre une primitive aléatoire *quelconque* et son espérance (lorsque celle ci est unique) :

$$\begin{aligned} E\left[\mu^2(\mathbf{x},\overline{\mathtt{x}})\right] &= \int_{\mathcal{M}} \overrightarrow{\overline{\mathtt{x}}\mathtt{y}}^t.\Sigma^{-1}_{\mathtt{xx}}.\overrightarrow{\overline{\mathtt{x}}\mathtt{y}}.p_{\mathbf{x}}(\mathtt{y}).d\mathcal{M}(\mathtt{y}) \\ &= \int_{\mathcal{M}} \mathrm{Tr}(\Sigma^{-1}_{\mathtt{xx}}.\overrightarrow{\overline{\mathtt{x}}\mathtt{y}}.\overrightarrow{\overline{\mathtt{x}}\mathtt{y}}^t).p_{\mathbf{x}}(\mathtt{y}).d\mathcal{M}(\mathtt{y}) \\ &= \mathrm{Tr}\left(\Sigma^{-1}_{\mathtt{xx}}.\int^{-1}_{\mathcal{M}} \overrightarrow{\overline{\mathtt{x}}\mathtt{y}}.\overrightarrow{\overline{\mathtt{x}}\mathtt{y}}^t.p_{\mathbf{x}}(\mathtt{y}).d\mathcal{M}(\mathtt{y})\right) \\ &= \mathrm{Tr}(\Sigma^{-1}_{\mathtt{xx}}.\Sigma_{\mathtt{xx}}) = \mathrm{Tr}(Id) \\ &= \mathcal{D} \end{aligned}$$

Le résultat est bien identique à celui du cas vectoriel.

D.2.3.4 Test du χ^2

La distance de Mahalanobis pour une primitive gaussienne suit approximativement une loi du χ^2 à $\mathcal{D}$ degrés de libertés (en particulier lorsque la covariance est faible). On peut donc utiliser le test du χ^2 comme dans le cas vectoriel.

Annexe E

Quelques rappels sur les tenseurs

E.1 Tenseurs

Nous présentons ici les éléments de bases permettant de comprendre les techniques de calcul tensoriel utilisées dans cette thèse. Le lecteur pourra se reporter à [Garrigues, 1999] pour un cours plus complet.

E.1.1 Espace vectoriel et espace dual - coordonnées

E.1.1.1 Notations

La théorie des tenseurs est associée à un certains nombres de règles de notations, qui permettent en pratique de ne pas se tromper dans l'écriture des coordonnées, des changements de base et des matrices. Ces règles sont, dans certains cas de véritables conventions (les indices des vecteurs de la base en bas, par exemple), mais correspondent dans d'autre cas à des résultats de théorèmes (on vérifie par exemple qu'un changement de base pour les indices contravariants d'un tenseur suit bien la même loi que pour les coordonnées contravariantes d'un vecteur). Le but de ce résumé n'étant pas d'exposer les tenseurs de façon rigoureuse, mais de façon claire et pratique, ces résultats seront pour la plupart passés sous silence.

Lorsque l'on dispose d'une série d'éléments (vecteurs d'une base, coordonnées d'un vecteur), on peut noter ces éléments de façon indicée. On placera dans certains cas l'indice en haut, dans d'autre cas en bas. Par exemple : e_i, x^i.

La première règle utilisée dans le calcul tensoriel est la convention d'Einstein : dans une sommation, on omet le signe de sommation $\Sigma_{i=1}^{n}$. Ainsi $\Sigma_{i=1}^{n} x^i e_i$ sera simplement noté $x^i e_i$, et appelé monôme. L'indice correspondant à la sommation est appelé indice muet, car on peut changer son nom sans conséquence pour l'extérieur du monôme qui le contient. Ainsi les égalités $x = x^i e_i$ et $x = x^k e_k$ sont formellement identiques. Un monôme peut contenir plusieurs sommations simultanées : $A_i^k x^i e_k$.

Les règles les plus importantes sont les règles sur le placement des indices : un indice muet doit se apparaître exactement deux fois dans un monôme, une fois placé en haut, l'autre fois placé en bas (voir exemples précédents). Ainsi, si l'on trouve un résultat de la forme $x = x_i e_i$ (indices placés à la même hauteur) ou $x = x_i y_i e^i$ (l'indice apparaît plus de deux fois), celui-ci correspond certainement à une erreur de notation, de raisonnement ou de calcul.

De même, un indice non muet (indice réel) ne doit apparaître qu'une seule fois dans un monôme, et, dans une sommation ou une égalité de monômes, les indices réels doivent être les mêmes (même nom, même hauteur) dans chaque terme. Sauf cas particuliers, on omettra dans une égalité le signe $\forall i \in \{1,...,n\}$, où i est un indice réel. Ainsi, $x_i = A_i^k x_k$ est correct (et est valable pour tout i), $x_i = A^{ik} x_k$ correspond certainement à une erreur.

On rappelle enfin la définition de l'indice de Kronecker :

$$\delta_i^k = \begin{cases} 1 \; si \; i = k \\ 0 \; si \; i \neq k \end{cases}$$

E.1.1.2 Base d'un espace vectoriel - coordonnées contravariantes d'un vecteur

Les tenseurs sont définis dans le cadre d'espace vectoriels réels. Nous supposerons très vite ces espaces Euclidiens (munis d'un produit scalaire). Nous allons rappeler quelques résultats de bases sur les espaces vectoriels, ce qui permettra en outre de se familiariser avec les notations et le vocabulaire de la théorie des tenseurs.

Soit donc un espace vectoriel réel E de dimension n, et une base $\{e_i\} = \{e_i\}_{1 \leq i \leq n}$ de cet espace. Soit $v \in E$. v est donc un vecteur de E et s'écrit comme combinaison linéaire de la base $\{e_i\}$, ses coordonnées **contravariantes** sont définis comme suit :

$$v = v^i e_i$$

On peut ranger ces coordonnées dans un vecteur colonne :

$$[v^\bullet] = \begin{bmatrix} v^1 \\ v^2 \\ ... \\ v^n \end{bmatrix}$$

Soit maintenant $\{e'_i\}$ une nouvelle base de E. La nouvelle base se définit sur l'ancienne par des relations de la forme :

$$e'_k = e_i A^i_k$$

En considérant le premier indice comme celui de ligne, l'autre indice comme celui des colonnes, on peut ranger les A^i_k dans une matrice :

$$[A^{\bullet}_{\bullet}] = \begin{bmatrix} A^1_1 & A^1_2 & A^1_n \\ A^2_1 & A^2_2 & A^2_n \\ & & \\ A^n_1 & A^n_2 & A^n_n \end{bmatrix}$$

On reconnaît alors dans la colonne k les coordonnées contravariantes de e'_k dans la base $\{e_i\}$. On appelle cette matrice matrice de passage de $\{e_i\}$ vers $\{e'_j\}$ et on la note :

$$\begin{bmatrix} \{e_i\} \\ \{e'_i\} \end{bmatrix} = [A^{\bullet}_{\bullet}] \Longleftrightarrow e'_k = e_i \begin{bmatrix} \{e_i\} \\ \{e'_i\} \end{bmatrix}^i_k$$

On constate aisément que la matrice de passage de $\{e'_j\}$ vers $\{e_i\}$, notée $\begin{bmatrix} \{e'_i\} \\ \{e_i\} \end{bmatrix}$ est son inverse. En écrivant $v = v^i e_i = v^i e'_k (A^{-1})^k_i = v'^k e'_k$, on trouve les formules de changement de base, soit sous forme matricielle, soit sous forme indicée :

$$v'^k = \begin{bmatrix} \{e'_i\} \\ \{e_i\} \end{bmatrix}^k_i v^i \Longleftrightarrow [v'^{\bullet}] = \begin{bmatrix} \{e'_i\} \\ \{e_i\} \end{bmatrix} [v^{\bullet}]$$

E.1.1.3 Base duale - coordonnées covariantes

Nous examinons maintenant les formes linéaires sur l'espace E. Il s'agit des applications linéaires de E dans $\mathbb{R}$, qui forment l'espace E^*, appelé espace dual, qui est aussi de dimension n. Une telle application peut être caractérisée par son action sur une base $\{e_j\}$ de E (elle est linéaire). On peut ainsi associer à $\{e_j\}$ base de E, la base duale $\{e^{*i}\}$ de E^* définie par l'action de chaque forme linéaire e^{*i} sur la base $\{e_j\}$: $e^{*i}(e_j) = \delta^i_j$.

Nous allons maintenant nous restreindre au cas euclidien, qui permet d'identifier facilement forme linéaire et vecteur (bien que certains des résultats qui suivent restent valables dans le cas général, mais au prix d'un écriture plus lourde). En effet, étant donné un produit scalaire $(\ |\ )$, on a le résultat suivant : pour toute forme linéaire $v^* \in E^*$: $x \in E \mapsto v(x) \in \mathbb{R}$, il existe un unique $v \in E$, tel que $\forall x \in E$, $v^*(x) = (v|x) = (x|v)$. On peut ainsi identifier v^* et v, et donc E et E^*. Dans la suite, on se place toujours dans le cas Euclidien, et on omet la distinction entre v^* et v. Donnons nous ainsi une base $\{e_j\}$ de E. On lui associe la base duale $\{e^{*i}\}$ de E^* identifiée à $\{e^i\}$ base de E. On a donc $e^{*i}(e_j) = (e^i|e_j)$, et la définition de la base duale devient :

$$(e^i|e_j) = \delta^i_j \qquad \forall i,j$$

Ainsi la base duale peut se définir de façon plus géométrique par les propriétés suivantes : $e^i \perp e_{j \neq i}$ et $(e^i|e_i) = 1$. On constate en particulier qu'une base orthonormée est sa propre base duale. On définit alors les coordonnées covariantes de $v \in E$ comme les coordonnées dans cette base :

$$v = v_i e^i$$

On peut aussi ranger ces coordonnées dans un vecteur ligne[50] :

$$[v_\bullet] = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

De plus, on remarque que $(v|e_k) = v_i(e^i|e_k) = v_i\delta^i_k = v_k$ et $(v|w) = (v_ie^i|w^je_j) = v_iw^j(e^i|e_j) = v_iw^j\delta^i_j = v_iw^i$. Donc :

$$v_i = (v|e_i) \text{ et } v^i = (v|e^i) \text{ et } v_iw^i = v^iw_i = (v|w)$$

Les formules de changement de base sont obtenues en écrivant $v'_{_k} = (v|e'_{_k}) = (v|e_iA^i_k) = (v|e_i)A^i_k = v_iA^i_k$:

$$v'_{_k} = v_i \begin{bmatrix} \{e_i\} \\ \{e'_{_i}\} \end{bmatrix}^i_k \Longleftrightarrow [v'_\bullet] = [v_\bullet] \begin{bmatrix} \{e_i\} \\ \{e'_{_i}\} \end{bmatrix} \Longleftrightarrow [v'_\bullet]^t = \begin{bmatrix} \{e_i\} \\ \{e'_{_i}\} \end{bmatrix}^t [v_\bullet]^t$$

On constate qu'en fait $v'_{_k} = v_i\, A^i_k$ est formellement la même formule que $e'_{_k} = e_iA^i_i$, ce qui justifie le terme de coordonnées covariantes (qui varient comme la base). On comprend de même le sens de contravariant, et on constate que l'on a bien :

$$e'^k = (A^{-1})^k_i e^i$$

E.1.2 Tenseurs Euclidiens

E.1.2.1 Définition

Soit E un espace vectoriel réel de dimension n, muni d'un produit scalaire. Un tenseur d'ordre p est une forme p-linéaire de $E \times E \times \dots \times E$ dans $\mathbb{R}$. L'ensemble des tenseurs d'ordre p sur E, forme un espace vectoriel noté $\otimes^p E$ et de dimension n^p.

Par convention, un tenseur d'ordre 0 est un scalaire, i.e. un réel qui ne dépend pas de la base. Pour cette raison, x^i ou toute autre coordonnée n'est pas un scalaire, et ne peut-être considéré comme un tenseur. Dans le cas où $p = 1$, il s'agit d'une forme linéaire de E. Or nous avons déjà identifié ces formes au vecteurs de E. On identifie donc $\otimes^1 E$ à E.

50. Certains auteurs notent ce vecteur en colonne. Nous présenterons donc toutes les formules avec $[v_\bullet]$ (vecteur ligne) et $[v_\bullet]^t$ (vecteur colonne).

E.1.2.2 Coordonnées

Pour éviter des notations trop lourdes, nous travaillerons sur un tenseur d'ordre 3, mais les notions introduites ici sont générales. Soit $T\ :\ (x,y,z) \in E \times E \times E \mapsto T(x,y,z) \in \mathbb{R}$. Si on utilise par exemple les coordonnées covariantes de a et les coordonnées contravariantes de b et c, on a :

$$T(x,y,z) = T(x_i e^i, y^j e_j, z^k e_k) = x_i y^j z^k T(e^i, e_j, e_k)$$

On définit les coordonnées 1-contravariantes 2-3-covariantes de T (nous montrerons au paragraphe E.1.3.2 qu'il s'agit véritablement des coordonnées du tenseur dans une certaine base définie à partie des bases normales et duales de E).

$$T^i_{jk} = T(e^i, e_j, e_k)$$

On obtient les formules de changement de base de la façon suivante : ${T'}^i_{\ jk} = T(e'^i, e'_j, e'_k) = T((A^{-1})^i_{\ l} e^l, e_m A^m_j, e_p A^p_k) = (A^{-1})^i_{\ l} T(e^l, e_m, e_p) A^m_j A^p_k = (A^{-1})^i_{\ l} T^l_{\ mp} A^m_j A^p_k$. Ceci justifie la généralisation des termes covariants et contravariants.

E.1.3 Produits Tensoriels

E.1.3.1 Définition

Le produit tensoriel d'un tenseur P de $\otimes^p E$ et d'un tenseur Q de $\otimes^q E$ est un tenseur de $\otimes^{p+q} E$ défini de la manière suivante (ici $p = 2$, $q = 3$) :

$$(P \otimes Q)(u,v,x,y,z) = P(u,v) Q(x,y,z) \qquad \forall u,v,x,y,z \in E$$

Le produit tensoriel d'un tenseur par un scalaire (tenseur d'ordre 0) correspond au produit externe.

Le produit tensoriel contracté d'un tenseur P de $\otimes^p E$ et d'un tenseur Q de $\otimes^q E$ ($p,q \geq 1$) est un tenseur de $\otimes^{p+q-2} E$ défini de la manière suivante (ici $p = 2$, $q = 3$) : $(P \overline{\otimes} Q)^{ijm} = P^{ij}_{\ \ k} Q^{km} = P^{ijk} Q^m_k$. On vérifie que cette définition est correcte car le tenseur défini ne dépend pas du choix de la base et des emplacements des indices.

On peut définir de même un produit doublement contracté de deux tenseurs d'ordre supérieur ou égal à 2 de la façon suivante : $(P \overline{\overline{\otimes}} Q)^i = P^i_{jk} Q^{jk} = P^{ijk} Q_{jk} = P^{ij}_{\ \ k} Q^k_j = P^{i\ k}_{\ j} Q^j_{\ k}$.

E.1.3.2 Propriétés

Une manière plus simple d'aborder les produits tensoriels et d'utiliser leurs propriétés. On constate tout-d'abord que le produit tensoriel est bilinéaire, associatif, mais non commutatif :

$$P \otimes (T \otimes Q) = (P \otimes T) \otimes Q = P \otimes T \otimes Q$$

De plus, on a en particulier (ici, on utilise à nouveau des coordonnées 1-contravariante 2-3-covariante, mais le résultat se généralise de façon évidente à n'importe quel type de coordonnées) : $(e_i \otimes e^j \otimes e^k)(x,y,z) = (e_i|x)(e^j|y)(e^k|z) = x_i y^j z^k$. L'ensemble $\{(e_i \otimes e^j \otimes e^k)\}$ forme donc une base de l'espace vectoriel des tenseurs sur E d'ordre 3, et :

$$T = T^i_{jk}(e_i \otimes e^j \otimes e^k)$$

Le produit contracté est lui aussi bilinéaire. Il n'est en général pas commutatif. Cependant, le produit contracté de 2 tenseurs d'ordre 1 (vecteurs) n'est autre que le produit scalaire, et est donc commutatif :

$$x \overline{\otimes} y = y \overline{\otimes} x = x_i y^i = x^i y_i = (x|y)$$

et en particulier

$$e_i \overline{\otimes} e^j = \delta_i^j$$

Le produit contracté est associatif lorsque le tenseur central est d'ordre supérieur ou égal à 2 :

$$P \overline{\otimes} (T \overline{\otimes} Q) = (P \overline{\otimes} T) \overline{\otimes} Q = P \overline{\otimes} T \overline{\otimes} Q \quad \text{si ordre} \, T \geq 2$$

Enfin, les produits tensoriels et contractés sont associatifs entre eux :

$$P \otimes (T \overline{\otimes} Q) = (P \otimes T) \overline{\otimes} Q = P \otimes T \overline{\otimes} Q$$

$$P \overline{\otimes} (T \otimes Q) = (P \overline{\otimes} T) \otimes Q = P \overline{\otimes} T \otimes Q$$

On peut ainsi retrouver la définition du produit tensoriel contracté de la manière suivante : $(P \overline{\otimes} Q) = (P^{ijk}(e_i \otimes e_j \otimes e_k)) \overline{\otimes} (Q_l^m(e^l \otimes e_m))$. Par bilinéarité, il vient $(P \overline{\otimes} Q) = P^{ijk} Q_l^m((e_i \otimes e_j \otimes e_k) \overline{\otimes} (e^l \otimes e_m))$, mais par associativité, $(e_i \otimes e_j \otimes e_k) \overline{\otimes} (e^l \otimes e_m) = e_i \otimes e_j \otimes e_k \overline{\otimes} e^l \otimes e_m = e_i \otimes e_j \otimes (e_k \overline{\otimes} e^l) \otimes e_m = \delta_l^k (e_i \otimes e_j \otimes e_m)$. Donc, finalement : $(P \overline{\otimes} Q) = P^{ijk} Q_k^m (e_i \otimes e_j \otimes e_m)$.

On peut aussi exprimer l'effet d'un tenseur P de $\otimes^p E$ sur un n-uplet de E (ici $p = 3$) :

$$P(x,y,z) = (((P \overline{\otimes} z) \overline{\otimes} y) \overline{\otimes} x)$$

E.1.4 Tenseurs d'ordre 2

E.1.4.1 Caractéristiques générales

Les tenseurs d'ordre 2 seront ici les plus intéressants. Un tenseur d'ordre 2 est une forme bilinéaire sur un espace vectoriel. Soit T un tenseur d'ordre 2. l'expression de $T(x,y)$ se simplifie car $(T\overline{\otimes}y)$ est d'ordre 1 et donc $(T\overline{\otimes}y)\overline{\otimes}x = x\overline{\otimes}(T\overline{\otimes}y)$ et finalement :

$$T(x,y) = x^i T_{ij} y^j = x\overline{\otimes}T\overline{\otimes}y$$

On associe à un couple de vecteur (x,y) le tenseur $x \otimes y$. L'ensemble de ces tenseurs forme un sous-ensemble de l'espace des tenseurs d'ordre 2 (qui n'est pas un sous-espace vectoriel), et ils sont appelés diades. Cette notation permet d'effectuer facilement les calculs. On a par exemple : $(x \otimes y)(a,b) = a\overline{\otimes}(x \otimes y)\overline{\otimes}b = (a\overline{\otimes}x) \otimes (y\overline{\otimes}b) = (a.x) \otimes (b.y) = (a.x)(b.y)$.

On associe à T l'endomorphisme de E, noté $\widetilde{T}$ ou ${}^{l}T$ qui vérifie :

$$\widetilde{T}(x) = T\overline{\otimes}x = (T^i_j x^j)e_i \qquad \forall x \in E$$

On peut se rappeler que si $y = \widetilde{T}(x)$, on a $y^i = T^i_j x^j$, et de même pour toutes les autres conformations d'indices. Réciproquement, tout endomorphisme $\widetilde{T}$ est associé au tenseur défini par : $T(x,y) = x\overline{\otimes}\widetilde{T}(y) \qquad \forall x,y \in E$. On constate enfin que le tenseur associé à $\widetilde{P} \circ \widetilde{Q}$ est $P\overline{\otimes}Q$.

Les tenseurs d'ordre 2 permettent ainsi d'identifier formes bilinéaires et endomorphismes.

E.1.4.2 Matrices d'un tenseur

En regroupant les coordonnées d'un tenseur, en faisant correspondre le premier indice à un indice de ligne et le second à un indice de colonne, on peut définir 4 matrices différentes suivant le type de coordonnées qu'on utilise. Voici la notation dans le cas des coordonnées purement covariantes :

$$[T_{\bullet\bullet}] = \begin{bmatrix} T_{11} & T_{12} & & T_{1n} \\ T_{21} & T_{22} & & T_{2n} \\ & & & \\ T_{n1} & T_{n2} & & T_{nn} \end{bmatrix}$$

On constate alors que si $y = \widetilde{T}(x) = T\overline{\otimes}x$, alors $[y^\bullet] = [T^\bullet_\bullet]\,[x^\bullet] = [T^{\bullet\bullet}]\,[x_\bullet]^t$ et $[y_\bullet]^t = [T_{\bullet\bullet}]\,[x^\bullet] = [T_\bullet{}^\bullet]\,[x_\bullet]^t$.

E.1.4.3 Transposée - Tenseur Symétrique

On associe à tout tenseur T d'ordre 2 son transposé noté T^t et défini par :

$$T^t(x,y) = T(y,x) \qquad \forall x,y \in E$$

Les coordonnées d'un tenseur et de son transposé sont reliées par les relations :

$$(T^t)_i^{\ j} = T^j_{\ i} \quad (T^t)^i_{\ j} = T_j^{\ i} \quad (T^t)^{ij} = T^{ji} \quad (T^t)_{ij} = T_{ji}$$

Les matrices d'un tenseur et de son transposé son donc reliées par les relations suivantes :

$$\left[(T^t)_{\bullet\bullet}\right] = [T_{\bullet\bullet}]^t \quad \left[(T^t)^{\bullet\bullet}\right] = [T^{\bullet\bullet}]^t \quad \left[(T^t)^{\bullet}_{\ \bullet}\right] = [T_{\bullet}^{\ \bullet}]^t \quad \left[(T^t)_{\bullet}^{\ \bullet}\right] = [T^{\bullet}_{\ \bullet}]^t$$

Un tenseur est dit symétrique s'il est son propre transposé. On déduit alors facilement ses propriétés de ce qui précède, et on remarque en particulier que si T est symétrique, alors $[T^{\bullet\bullet}]$ et $[T_{\bullet\bullet}]$ sont symétriques, mais $[T^{\bullet}_{\ \bullet}]$ et $[T_{\bullet}^{\ \bullet}]$ ne le sont pas nécessairement.

E.1.4.4 Tenseur métrique

Le tenseur métrique, noté G, est défini par : $G(x,y) = (x|y) \qquad \forall x,y \in E$. Il s'agit donc d'un tenseur symétrique, Ses coordonnées sont :

$$g^{ij} = (e^i|e^j) \quad g_{ij} = (e_i|e_j) \qquad g^i_{\ j} = \delta^i_j \quad g_i^{\ j} = \delta^i_j$$

Ses matrices ont des caractéristiques particulières :

$$[g_{\bullet}^{\ \bullet}] = [g^{\bullet}_{\ \bullet}] = Id \quad \text{et} \quad [g_{\bullet\bullet}] = [g^{\bullet\bullet}]^{-1}$$

L'endomorphisme associé à G est l'identité :

$$\widetilde{G} = Id$$

Le tenseur métrique vérifie donc :

$$G\overline{\otimes}x = x\overline{\otimes}G = x \qquad \forall x \in E$$

Le principal intérêt du tenseur métrique est de permettre la conversion entre coordonnées covariantes et contravariantes. On l'appelle ainsi souvent ascenseur d'indice. On a pour les vecteurs : $x = G\overline{\otimes}x$, donc :

$$x^i = g^{ij} x_j \qquad x_i = g_{ij} x^j$$

Voici un exemple pour les tenseurs d'ordre supérieur : $T(x,y,z) = T^{ijk} x_i y_j z_k = T_i^{\ jk} x^i y_j z_k = T_i^{\ jk} g^{il} x_l y_j z_k$, donc $T^{ijk} = g^{il} T_l^{\ jk}$. Pour les tenseurs d'ordre 2, on a, en terme de matrices :

$$[T^{\bullet}_{\ \bullet}] = [g^{\bullet\bullet}]\,[T_{\bullet}^{\ \bullet}]\,[g_{\bullet\bullet}] = [T^{\bullet\bullet}]\,[g_{\bullet\bullet}] = [g^{\bullet\bullet}]\,[T_{\bullet\bullet}]$$

E.1.4.5 Trace et déterminant - invariants

Les tenseurs d'ordre 2 sur E pouvant être identifiés aux endomorphismes de E, on peut définir leur trace et leur déterminant :

$$Tr(T) = Tr(\widetilde{T}) = Tr([T^{\bullet}_{\ \bullet}]) = Tr([T_{\bullet}^{\ \bullet}])$$

$$Det(T) = Det(\widetilde{T}) = Det([T^{\bullet}_{\ \bullet}]) = Det([T_{\bullet}^{\ \bullet}])$$

On remarque en outre que la trace est invariant (par changement de base), appelé premier invariant, qui peut se ré-écrire :

$$\Theta_1 = Tr(T) = T\overline{\overline{\otimes}}G = g^{ij} T_{ij} = g_{ij} T^{ij} = T^i_{\ i} = T_i^{\ i}$$

On peut définir un second invariant par la formule suivante :

$$\Theta_2 = (T^1_{\ 1} T^2_{\ 2} - T^2_{\ 1} T^1_{\ 2}) + (T^1_{\ 1} T^3_{\ 3} - T^3_{\ 1} T^1_{\ 3}) + (T^2_{\ 2} T^3_{\ 3} - T^3_{\ 2} T^2_{\ 3})$$

Et le déterminant est un troisième invariant :

$$\Theta_3 = Det(T)$$

E.1.4.6 Valeurs propres et vecteurs propres

Les valeurs propres et vecteurs propres d'un tenseur d'ordre 2 sont définis aussi comme les valeurs propres et vecteurs propres de l'endomorphisme associé. Ainsi, λ est valeur propre de T si :

$$Det(T - \lambda G) = 0$$

Les composantes contravariantes des vecteurs propres de T sont obtenues en cherchant les colonnes propres de $[T^{\bullet}_{\ \bullet}]$. Les composantes covariantes des vecteurs propres de T sont obtenues en cherchant les colonnes propres de $[T_{\bullet}^{\ \bullet}]$. En revanche, les colonnes propres de $[T^{\bullet\bullet}]$ et $[T_{\bullet\bullet}]$ n'ont pas de signification particulière.

On a enfin les résultat suivant :

1. Un tenseur symétrique réel possède toujours n valeurs propres réelles (distinctes ou non).
2. Les espaces propres sont orthogonaux entre eux.
3. Dans une base construite sur les vecteurs propres, les matrices $[T^{\bullet}_{\bullet}]$ et $[T_{\bullet}{}^{\bullet}]$ sont diagonales, et les termes de la diagonale sont les valeurs propres.
4. Dans le cas d'un tenseur symétrique, on peut toujours construire une base orthogonale sur les espaces propres.

Annexe F

Détail des calculs du chapitre 8

F.1 Calcul du modèle plan

Nous montrons ici comment calculer le modèle plan avec bruit sur l'orientation (section 8.3.4.2), i.e. comment calculer la densité de probabilité de (s,n_s) connaissant le point m, la loi sur la normale $p_{\mathbf{n}_m}(n)$ et la distance u qui sépare les deux points.

Cette démonstration suit la démarche suivante : on va écrire le point s comme une fonction de n et des coordonnées (u_1,u_2) du point s dans une base (e_1,e_2) du plan (m,n) :

$$(ms,n_s) = (u_1 e_1 + u_2 e_2,n)$$

Heureusement, le problème est invariant par rapport au choix de cette base[51], ce choix ne peut donc influencer le résultat final.

On peut alors écrire la loi de probabilité de (s,n_s) en fonction de la loi sur n et de la loi sur (u_1,u_2) :

$$p(s,n_s) = p(n).p(u_1,u_2) / \left| \frac{\partial(s,n_s)}{\partial(n,u_1,u_2)} \right|$$

Il suffit enfin d'écrire la loi sur $p(u_1,u_2)$ pour conclure.

Choix de la base (e_1,e_2) en fonction de n Nous devons donc choisir une base (e_1,e_2) du plan (m,n), i.e. choisir deux vecteurs orthonormaux et perpendiculaires à n (l'origine de notre base sera bien-sûr le point m). Le problème est que la normale n va varier. Il faut donc choisir une base pour chaque normale n possible. On va alors utiliser l'astuce suivante : on va faire notre choix pour simplifier les calculs de $p(s)$ pour s situé dans une zone de travail aussi limitée que nécessaire.

La zone qui nous intéressera sera en l'occurrence un plan d'origine m, contenant $\overline{n}$ la moyenne de $\mathbf{n}$ et donné par sa normale a. Nous allons faire un choix de la base (e_1,e_2) en

51. C'est l'invariance par rapport aux rotations d'axe n.

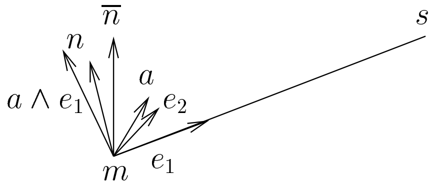


FIG. F.1 – *Choix de la base (e_1,e_2) du plan (m,n) adaptée au calcul de $p(s,n_s)$ pour s dans le plan (m,a) où la normale a est perpendiculaire à $\overline{n}$: le vecteur e_1 est perpendiculaire à a et n, et le vecteur e_2 est perpendiculaire à e_1 et n.*
On obtient ainsi une base orthonormale (e_1,e_2) du plan de normale n qui permet d'avoir $u_2 = 0$ dans le plan (m,a).

fonction de n qui permette d'avoir $u_2 = (e_2|ms) = 0$ dans ce plan de travail. Soit donc un plan quelconque donné par sa normale n. Pour annuler la coordonnée u_2 sur l'intersection entre ce plan et notre plan de travail, il suffit de prendre e_1 qui soit perpendiculaire à a :

$$e_1 = \frac{n \wedge a}{\|n \wedge a\|}$$

On complète alors la base avec :

$$e_2 = \frac{n \wedge e_1}{\|n \wedge e_1\|} = n \wedge \frac{n \wedge a}{\|n \wedge a\|}$$

Le point de coordonnées (u_1,u_2) est alors :

$$(s,n_s) = (u_1 e_1 + u_2 e_2, n)$$

Calcul de la loi sur $p(s,n_s)$ La loi de probabilité est donnée par :

$$p(s,n_s) = p(n).p(u_1,u_2) / \left| \frac{\partial(s,n_s)}{\partial(n,u_1,u_2)} \right|$$

Il ne nous reste donc qu'à calculer $\frac{\partial(s,n_s)}{\partial(n,u_1,u_2)}$. Ce calcul demande beaucoup d'attention, car les normales ne forment pas un espace vectoriel. Pour éviter toute erreur, nous utiliserons dans un premier temps de petites variations de la normale n et des coordonnées $u_1$et u_2 :

$$dn = dn_1.e_1 + dn_2.e_2 \qquad du_1 \quad du_2$$

et nous ne ferons le calcul que pour $u_2 = 0$, i.e. pour les points de notre plan de travail :

$$ds = u_1 \frac{\partial e_1}{\partial n} dn + du_1 e_1 + du_2 e_2$$

or on a :

$$\frac{\partial e_1}{\partial n} dn = \left(Id - \frac{(a \wedge n)(a \wedge n)^t}{\|a \wedge n\|^2} \right) \frac{a \wedge dn}{\|a \wedge n\|} = (Id - e_1.e_1^t).\frac{dn_1.a \wedge e_1 + dn_2.a \wedge e_2}{\|n \wedge a\|}$$

$$= \frac{dn_1.a \wedge e_1 + dn_2.(Id - e_1.e_1^t).a \wedge e_2}{\|n \wedge a\|}$$

et donc :

$$ds = u_1 \frac{dn_1.a \wedge e_1 + dn_2.(Id - e_1.e_1^t).a \wedge e_2}{\|n \wedge a\|} + du_1 e_1 + du_2 e_2$$

pour exprimer les variations de $n_s = n_m$, il faut remarquer que n_s ne peut être situé que dans le plan perpendiculaire à ms (de base $(a, a \wedge e_1)$) et peut donc être exprimé par le seul angle :

$$\theta = (\widehat{a \wedge e_1, n_s}) = \pi/2 - (\widehat{a, n_s}) = \pi/2 - \arccos(a|n_s)$$

dont les variations suivent :

$$d\theta = d(\widehat{a,n_s}) = \frac{d(a|n_s)}{\sqrt{1-(a|n_s)^2}} = \frac{(a|dn_1.e_1 + dn_2.e_2)}{\sqrt{1-(a|n_s)^2}} = dn_2.\frac{(a|e_2)}{\sqrt{1-(a|n_s)^2}} = dn_2$$

finalement, on peut exprimer le jacobien de (s,n_s) en fonction de (n,u_1,u_2) :

$$\frac{\partial(s,\theta)}{\partial(n_1,n_2,u_1,u_2)} = \begin{pmatrix} \frac{\partial s}{\partial n_1} & \frac{\partial s}{\partial n_2} & \frac{\partial s}{\partial u_1} & \frac{\partial s}{\partial u_2} \\ \frac{\partial \theta}{\partial n_1} & \frac{\partial \theta}{\partial n_2} & \frac{\partial \theta}{\partial u_1} & \frac{\partial \theta}{\partial u_2} \end{pmatrix} = \begin{pmatrix} u_1 \frac{a \wedge e_1}{\|n \wedge a\|} & u_1 \frac{(Id - e_1.e_1^t).a \wedge e_2}{\|n \wedge a\|} & e_1 & e_2 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

dont on calcule le déterminant :

$$\left| \frac{\partial(s,\theta)}{\partial(n_1,n_2,u_1,u_2)} \right| = \left| \begin{array}{ccc} u_1 \frac{a \wedge e_1}{\|n \wedge a\|} & e_1 & e_2 \end{array} \right| = |u_1| . \frac{\left| \begin{array}{ccc} a \wedge e_1 & e_1 & n \wedge e_1 \end{array} \right|}{\|n \wedge a\|}$$

en remarquant que $\left| \begin{array}{ccc} a \wedge e_1 & e_1 & n \wedge e_1 \end{array} \right| = (a \wedge n | e_1) = \|n \wedge a\|$, on obtient finalement :

$$\left| \frac{\partial(s,\theta)}{\partial(n_1,n_2,u_1,u_2)} \right| = |u_1|$$

et donc :

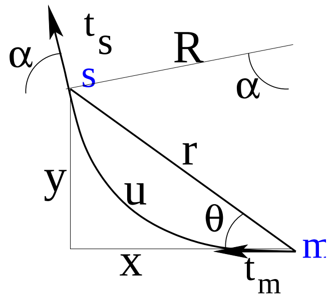
$$p(s,n_s|u_2 = 0) = \frac{\delta_u(\|ms\|).\delta_0((ms|e_2)).p_{\mathbf{n}}(n_s).p(u_1,0)}{|u_1|}$$

Conclusion Il ne nous reste qu'à écrire la probabilité $p(u_1,u_2)$. Celle-ci doit être isotrope (pour assurer l'invariance par transformation rigide), et se relie à la loi sur l'éloignement par $p(u_1,u_2) = p_{\mathbf{u}}(\sqrt{u_1^2 + u_2^2}) = p_{\mathbf{u}}(\|ms\|)$ et donc $p(u_1,0) = p_{\mathbf{u}}(u_1)$. On obtient finalement :

$$p(s,n_s|u) = \frac{\delta_u(\|ms\|).p_{\mathbf{n}}(n_s).p_{\mathbf{u}}(u)}{|u|}$$

F.2 Calcul du modèle sphérique

Nous présentons ici le calcul du modèle sphérique à éloignement connu (section 8.5.1.1). On utilisera les notation suivantes :



Rappelons le lien entre le rayon du cercle et sa courbure, et entre les angles :

$$R = 1/C \qquad \alpha = 2\theta$$

les coordonnées cartésiennes sont données par :

$$x = r\cos\theta = R\sin 2\theta = 2R\sin\theta\cos\theta$$

$$y = r\sin\theta = R(1-\cos 2\theta) = 2R\sin\theta\sin\theta$$

on peut alors tout calculer à partir de r (la norme de ms), x (le produit scalaire entre ms et t), y et θ (qui se déduisent des précédents) :

$$R = \frac{r}{2\sin\theta} = \frac{r^2}{2y} \qquad C = \frac{2y}{r^2} \qquad u = R.2\theta = \frac{\theta.r}{\sin\theta} \sim_{\theta\to 0} r \tag{F.1}$$

Et dans l'autre sens (à partir de C et u) :

$$\theta = \frac{u}{2R} = \frac{Cu}{2} \qquad x = \frac{\sin(Cu)}{C} \qquad y = \frac{1-\cos(Cu)}{C} \qquad r = \frac{2\sin(Cu/2)}{C}$$

On peut donner la probabilité de s, en le paramétrant directement par l'éloignement u, la courbure C et l'angle α de la tangente en s :

$$p(u,C,\alpha|u_0) = \delta_{u_0}(u).p_{\mathbf{C}}(C).\delta_{2\theta}(\alpha)$$

Pour passer en coordonnées cartésiennes, il faut calculer le jacobien du changement de paramétrage :

$$J = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial C} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial C} \end{bmatrix} = \begin{bmatrix} \cos(Cu) & \frac{Cu\cos(Cu)-\sin(Cu)}{C^2} \\ \sin(Cu) & \frac{Cu\sin(Cu)-(1-\cos(Cu))}{C^2} \end{bmatrix}$$

$$|J| = \begin{vmatrix} \cos(Cu) & \frac{Cu\cos(Cu)-\sin(Cu)}{C^2} \\ \sin(Cu) & \frac{Cu\sin(Cu)-(1-\cos(Cu))}{C^2} \end{vmatrix} = \begin{vmatrix} \cos(Cu) & \frac{-\sin(Cu)}{C^2} \\ \sin(Cu) & \frac{-(1-\cos(Cu))}{C^2} \end{vmatrix}$$

$$= \frac{\cos^2(Cu) - \cos(Cu) + \sin^2(Cu)}{C^2} = \frac{1-\cos(Cu)}{C^2} = \frac{2\sin^2(Cu/2)}{C^2} = r^2/2$$

On obtient, en coordonnées cartésiennes, la loi suivante :

$$p(x,y,\alpha|u) = \delta_u\left(\frac{\theta.r}{\sin\theta}\right).\frac{p_{\mathbf{C}}\left(2y/r^2\right)}{r^2}.\delta_{2\theta}(\alpha)$$

que l'on peut approximer par (en utilisant $u = \frac{\theta.r}{\sin\theta} \simeq r$ et $C = 2y/r^2 \simeq 2\theta/r$, approximations valables lorsque θ reste faible) :

$$p(x,y,\alpha|u) \simeq \delta_u\left(r\right).\frac{p_{\mathbf{C}}\left(2\theta/r\right)}{r^2}.\delta_{2\theta}(\alpha) \simeq \delta_u\left(r\right).\frac{p_{\mathbf{C}}\left(2y/r^2\right)}{r^2}.\delta_{2\theta}(\alpha)$$

F.3 Simulation du SCF 2D

Nous présentons ici la recherche d'une formule approchée du modèle stochastique à partir de la simulation (voir section 8.5.2.4).

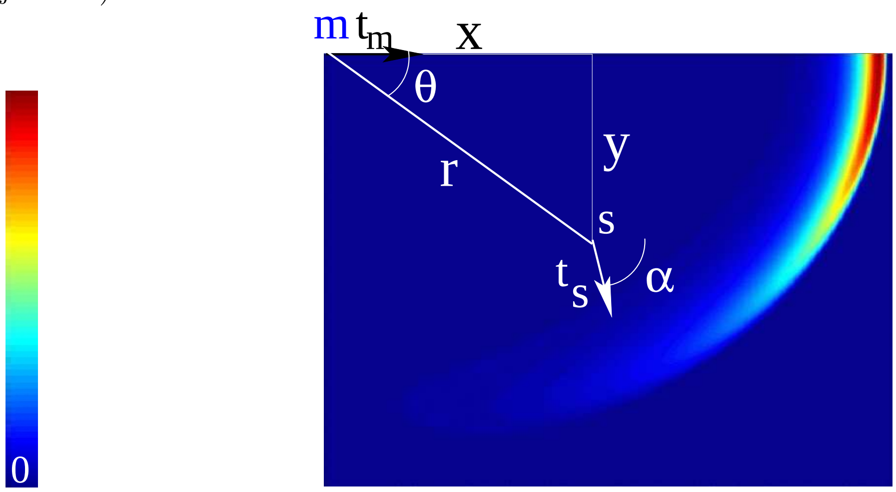
F.3.1 Éloignement connu

Nous allons dans un premier temps garder les informations relatives qu'aux points correspondant à un éloignement u imposé. Pour chaque trajectoire, si le point d'éloignement (i.e. d'abscisse curviligne) u passe par le point s avec l'orientation t_s alors nous additionnerons le tenseur $t_s.t_s^t$ au tenseur associé au pixel contenant s. L'image ainsi obtenue sera donc bien une version discrétisée de :

$$p(s,t_s|u)$$

ou plus exactement du champ de tenseur qui lui correspond (voir section 8.2.3). Nous analyserons séparément la trace des tenseurs qui correspond à la probabilité du point $p(s|u)$ et la forme des tenseurs $T(t_s|s,u)$ qui correspond à la loi de probabilité sur les normales en un point s connu.

Probabilité $p(s|u)$ du seul point (trace des tenseurs) La trace du champ de tenseurs est illustré dans l'image suivante (on a lancé ici 25 millions de trajectoires, stocké les tenseurs pour les points d'abscisse curviligne $u = 1\,mm$, avec un bruit gaussien sur l'orientation dont la variance est de $\sigma^2 = 1\,rad^2/mm$ et donc égale à 1 en fin de trajectoire) :



On constate que les points les plus probables sont répartis autour d'une ellipse centrée sur l'origine (i.e. le point connu m). Nous nous sommes donc intéressé de plus près à cette ellipse, en cherchant, dans chaque direction (i.e. pour chaque θ), le point le plus probable. Ceci nous a permis de vérifier qu'il s'agissait bien d'une ellipse[52] et que son grand rayon r_x était légèrement inférieur à u, et son petit rayon r_y égal à $0.9 * u$. Il semble alors naturel de remplacer les coordonnées polaires (r,θ) par les coordonnées (r',θ') qui tiennent compte de l'anisotropie de l'ellipse :

$$(x,y) = r'.(c_x.\cos(\theta'),c_y.\sin(\theta')) \qquad c_x = 0.98\, c_y = 0.87.c_x$$

$$r' = \sqrt{x^2/c_x^2 + y^2/c_y^2}$$

$$\theta' = \arcsin\left(\frac{y}{\sqrt{x^2.c_y^2/c_x^2 + y^2}}\right) \approx \frac{y}{\sqrt{x^2.c_y^2/c_x^2 + y^2}}$$

$$\theta' = \arctan\left(\frac{y/c_y}{x/c_x}\right) = \arctan\left(\frac{c_x}{c_y}\tan\theta\right)$$

avec ces coordonnées corrigées, l'équation de l'ellipse des points les plus probables devient tout simplement $r' = u$.

Nous avons ensuite étudié les probabilités cumulés des points dans chaque direction (i.e. pour chaque θ). On a alors pu constater qu'elles suivaient une parfaite gaussienne en

52. Pour tester qu'une série de points de coordonnées (x_i,y_i) suit une ellipse d'équation $x^2/r_x^2 + y^2/r_y^2 = 1$ et donc $y^2 = r_y^2 - x^2.r_y^2/r_x^2 = \alpha.x^2 + \beta$, on effectue une régression linéaire sur l'ensemble de données (x_i^2,y_i^2). On vérifie alors que le modèle est valable, et on calcule α et β et on en déduit $r_y = \sqrt{\beta}$ et $r_x = \sqrt{-\beta/\alpha}$.

θ[53] :

$$p(\theta|u) = \int p(s = (r\cos(\theta), r\sin(\theta))|u)dr = \exp(-\theta^2/2\sigma_u^2)/k_u$$

Nous avons ensuite cherché les expressions de σ_u^2 et $k_u$en fonction de u. On a trouvé que les modèles suivants étaient parfaitement satisfaisants[54] :

$$\sigma_u^2 = k * \sigma^2 * u \qquad k = 0.34$$

$$k_u \sim (\sigma_u^2)^{1.5}$$

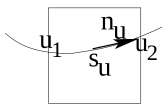
Répartition des normales (forme des tenseurs) La forme des tenseurs pour un éloignement fixé est en-fait très complexe à décrire, dans la mesure où de nombreuses situations différentes peuvent se présenter :

- Les trajectoires qui sont allées le plus loin à l'horizontale sont celles pour lesquelles l'orientation n'a pas variée. La direction y est donc horizontale et bien définie (variance faible).
- Les trajectoires qui ne sont pas allés très loin sont les plus tordues. La direction y est donc mal définie.

Nous ne chercherons donc pas à leur trouver une forme explicite, et attendrons pour cela la prise en compte des différents éloignements possibles, qui donnera des résultats plus simples.

F.3.2 Éloignement quelconque

Nous allons maintenant simuler le SCF en intégrant suivant la loi a-priori sur l'éloignement u : si une trajectoire passe par le pixel s avec l'orientation t_s (nous veillerons à prendre des pixels suffisamment petits pour que la variation d'orientation soit négligeable à l'intérieur d'un même pixel), entre les abscisses curvilignes u_1 et u_2 :



alors nous additionnerons le tenseur $t_s.t_s^t$ au tenseur $T(s)$ en s, pondéré par la probabilité a-priori de l'élément de trajectoire donnée par $P(u_1 < u < u_2) = \int_{u_1}^{u_2} p_{\mathbf{u}}(u)du$.

53. Pour tester qu'une série de données (x_i,y_i) suit une gaussienne, i.e. que $y = \exp(-x^2/2\sigma^2)/k$, ou de façon équivalente, que $\log y = -x^2/2\sigma^2 - \log k = \alpha x^2 + \beta$, il suffit de faire une régression linéaire sur l'ensemble de données $(x_i^2, \log y_i)$. On vérifie alors que le modèle est valable, et on calcule au passage α et β qui permettent de déduire les valeurs de $\sigma^2 = -1/2\alpha$ et de $k = \exp(-\beta)$.

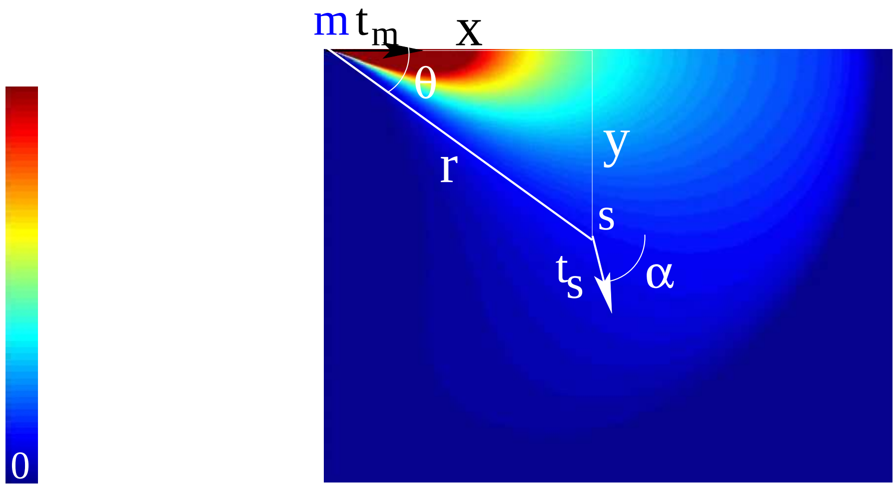
54. Pour tester qu'une série de données (x_i,y_i) suit une loi de type $y = k.x^\alpha$, ou de façon équivalente, que $\log y = \alpha \log x + \log k = \alpha \log x + \beta$, il suffit de faire une régression linéaire sur l'ensemble de données $(\log x_i, \log y_i)$. On vérifie alors que le modèle est valable, et on calcule au passage α et β qui permettent de déduire les valeurs de α et de $k = \exp(\beta)$.

L'image ainsi obtenue sera donc bien une version discrétisée de :

$$p(s,t_s) = \int p(s,t_s|u = u_1)p_{\mathbf{u}}(u_1)du_1$$

Comme dans le cas précédent, nous analysons séparément la trace $p(s)$ et la forme des tenseurs $T(t_s|s)$.

Probabilité $p(s|u)$ du seul point (trace des tenseurs) Nous avons tracé ici les traces des tenseurs (i.e. $p(s)$) ainsi obtenus (on a lancé ici 25 millions de trajectoires, avec $p_{\mathbf{u}}(u)$ uniforme sur [0,1], le plan restreint à un quart de plan par symétrie, avec un bruit gaussien sur l'orientation dont la variance est de $\sigma^2 = 1\, rad^2/mm$ et donc égale à 1 en fin de trajectoire) :



Bien évidement, certaines zones de cette figure sont mal définies, donc non exploitable : la partie inférieure gauche (où trop peu de trajectoires sont passées, car il faut une forte courbure pour y arriver), et l'extérieur du cercle de rayon 0.75 (à partir duquel on a des trajectoires qui se finissent). Nous nous concentrerons donc sur la zone centrale, bien définie.

Pour essayer de trouver une forme analytique à ce champ, nous allons raisonner à partir de la forme trouvée pour les éloignements (abscisses curvilignes) fixes. Nous allons faire l'approximation suivante : nous allons supposer que les points d'abscisses curvilignes u ne sont pas répartis autour de l'ellipse de rayons $r_x = c_x.u$ et $r_y = c_y.u$ mais sont exactement sur cette ellipse. Le point (x,y) est alors forcément à l'abscisse curviligne $u = \sqrt{x^2/c_x^2 + y^2/c_y^2}$, et on a toujours la probabilité suivante pour θ :

$$p(\theta|u) = \frac{\exp(-\theta^2/2.(k * \sigma^2 * u))}{(u)^{1.5}} = \frac{\exp(-\frac{\theta^2}{2.(k*\sigma^2*\sqrt{x^2/c_x^2+y^2/c_y^2})})}{(\sqrt{x^2/c_x^2 + y^2/c_y^2})^{1.5}}$$

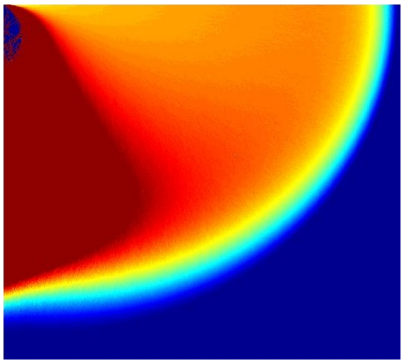
on peut finalement ré-écrire la probabilité d'un point de coordonnée (x,y) sachant qu'il est d'abscisse curviligne u (le Dirac δ permet d'exprimer la condition que le point est sur l'ellipse) :

$$p((x,y)|u) = \delta_u(\sqrt{x^2/c_x^2 + y^2/c_y^2}) \frac{\exp(-\frac{\arcsin(y/\sqrt{x^2+y^2})^2}{2.(k*\sigma^2*\sqrt{x^2/c_x^2+y^2/c_y^2})})}{(\sqrt{x^2/c_x^2 + y^2/c_y^2})^{1.5}}$$

lorsqu'on va intégrer sur u, pour prendre en compte la loi a-priori sur l'éloignement $p_{\mathbf{u}}(u)$, on va finalement obtenir :

$$p(x,y) = p_{\mathbf{u}}(\sqrt{x^2/c_x^2 + y^2/c_y^2}) \frac{\exp(-\frac{\arcsin(y/\sqrt{x^2+y^2})^2}{2.(k*\sigma^2*\sqrt{x^2/c_x^2+y^2/c_y^2})})}{(\sqrt{x^2/c_x^2 + y^2/c_y^2})^{1.5}}$$

nous allons donc vérifier que le champ simulé suit bien cette équation. Pour cela on calcule le rapport entre le champ simulé et le champ prévu. En voici l'image :



les oscillations du rapport dans la zone utile (en orange) sont de l'ordre de 20%, ce qui prouve que notre modèle est relativement bon.

En résumé, on a :

$$p(s) \simeq p_{\mathbf{u}}\left(\sqrt{\frac{x^2}{c_x^2} + \frac{y^2}{c_y^2}}\right) . \frac{\exp\left(-\frac{\arcsin\left(y/\sqrt{x^2+y^2}\right)^2}{2*k*\sigma^2*\sqrt{\frac{x^2}{c_x^2}+\frac{y^2}{c_y^2}}}\right)}{\left(\sqrt{\frac{x^2}{c_x^2} + \frac{y^2}{c_y^2}}\right)^{1.5}}$$

$$k = 0.34 = 1/2^{1.5} \qquad c_x = 0.98 \qquad c_y = 0.87.c_x$$

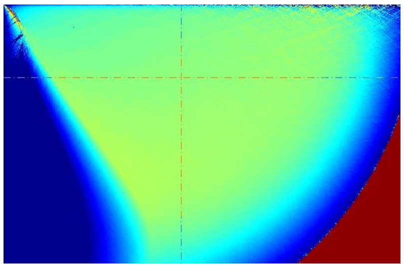
on peut aussi utiliser l'approximation suivante (en utilisant $c_x \approx 1$, $c_y \approx 1$ et $\arcsin(x) \approx x$) :

$$p(x,y) \simeq p_u(r) . \frac{\exp\left(-\frac{y^2}{2*k*\sigma^2*r^{1.5}}\right)}{r^{1.5}}$$

Répartition des normales (forme des tenseurs) Nous allons maintenant essayer de retrouver la forme de $T(t_s|s)$, i.e. la loi sur la tangente en chaque point. On regarde dans un premier temps la tangente la plus probable sur le champ simulé, en diagonalisant le tenseur et en étudiant le premier vecteur propre. On constate alors que, comme avec les modèles affine et sphérique, on a une relation très simple entre l'angle α formé par la direction t_s la plus probable en s et la direction t_m à l'origine et l'angle θ formé par le segment ms reliant les deux points et la tangente à l'origine :

$$\alpha = 1.5 * \theta$$

rappelons que nous avions $\alpha = \theta$ avec le modèle affine et $\alpha = 2\theta$ avec le modèle sphérique. On peut vérifier les variations du rapport α/θ sur l'ensemble de la figure : il varie, dans la zone bien définie (en vert) entre 1.42 et 1.55 :



Maintenant que nous connaissons la tangente moyenne, intéressons nous à sa variance σ^2, i.e. au rapport entre la deuxième valeur propre du tenseur et sa trace. On constate alors rapidement que les Iso-courbes de ce rapport sont des ellipses de grand axe parallèle à la tangente à l'origine, et de petit axe perpendiculaire. Le rapport entre les rayons sont de 1.5. Il semble de plus que la valeur du rapport soit proportionnel au rayon de l'ellipse, i.e. à u. On trace donc finalement $\sigma^2/\sqrt{x^2/1.5^2 + y^2}$:

le rapport oscille entre 0.47 et 0.52, dans la zone bien définie (en bleu). Le coefficient de proportionnalité est donc égal à 0.5, ce qui est un peu surprenant puisque nous nous sommes basé sur un modèle stochastique où la variance de la tangente était égale à $\sigma^2.u$, avec, dans cette simulation $\sigma^2 = 1$. On ne retrouve donc que la moitié de la variance prévue, pour des raisons que nous ignorons encore

F.4 Calculs des modèles avec bruit sur l'orientation

Nous présentons ici le détail du calcul permettant de prendre en compte le bruit sur l'orientation dans le modèle général (section 8.6.1).

Nous partirons de la formule générale sans bruit (eq. 8.8) :

$$p(x,y,\alpha) = p_u(r).\frac{\exp\left(-\frac{\theta^2}{2.\sigma_{forme}^2.r^{2(A-1)}}\right)}{r^A}.\exp\left(-\frac{(\alpha - A\theta)^2}{2.\sigma_{t_s}^2.r}\right)$$

nous allons nous demander ce qui change dans cette formule lorsque l'on fait varier la normale à l'origine (i.e. au point connu m), lorsqu'on la fait tourner d'un angle θ_o par exemple : r ne changera pas, mais θ va changer pour devenir $\theta - \theta_o$ et α va devenir $\alpha - \alpha_0$. La loi sur la normale à l'origine s'écrit alors comme une loi p_{θ_o} sur θ_o, que nous supposerons gaussienne (avec une variance $\sigma_{t_m}^2$, infinie pour représenter une loi uniforme). On peut alors ré-écrire l'intégrale :

$$p(x,y,\alpha) = \int_{\theta_0=-\pi}^{\pi} \left\{ \begin{array}{c} p_u(r).\frac{\exp\left(-\frac{(\theta-\theta_0)^2}{2.\sigma_{forme}^2.r^{2(A-1)}}\right)}{r^A}. \\ \exp\left(-\frac{((\alpha-\theta_0)-A(\theta-\theta_0))^2}{2.\sigma_{t_s}^2.r}\right).\exp\left(-\frac{\theta_0^2}{2.\sigma_{t_m}^2}\right) \end{array} \right\}.d\theta_o$$

Cette intégrale est relativement simple à calculer, à condition d'intégrer de $-\infty$ à $+\infty$ au lieu de $-\pi$ à π. Cette approximation sera correcte lorsque la dispersion sera faible (inférieure à 20 degrés, ce qui est en général le cas pour le bruit sur l'orientation des normales) ou lorsqu'elle sera infinie (i.e. lorsqu'on travaille avec le loi uniforme). On obtient alors, après simplifications, les résultats donnés dans la section 8.6.1.

F.5 Calcul de la variance de la dispersion autour de l'espace tangent

Calcul de la dispersion pour le modèle général sans bruit de mesure Nous présentons ici le calcul de la dispersion pour le modèle général sans bruit de mesure (voir section 8.5.3). Nous repartons de la formule pour le seule position du point (eq. 8.8) :

$$p(x,y) \simeq p_u(r).\frac{\exp\left(-\frac{y^2}{2.\sigma_{forme}^2.r^{2A}}\right)}{r^A}$$

Nous allons simplifier cette formule en utilisant les approximations $\sqrt{x^2+y^2} = x$ et $\theta = y/x$ (l'erreur commise reste inférieure à 15% pour $\theta < \pi/6$), qui permettent de découpler x et y :

$$p(x,y) \simeq p_u(x) . \frac{\exp\left(-\frac{y^2}{2.\sigma^2_{forme}.x^{2A}}\right)}{x^A}$$

On constate alors immédiatement que, pour x fixé, on a sur y une loi gaussienne de variance :

$$\sigma^2_{y|x} = \sigma^2_{forme}.x^{2A}$$

Calcul de la dispersion pour le modèle général avec bruit de mesure On peut calculer de la même façon la variance de la dispersion pour le modèle général avec bruit sur l'orientation (section 8.6.1, eq. 8.9) :

$$\sigma^2_{y|x} = \sigma^2_{forme}.x^{2A} + \sigma^2_{n_m}.x^2$$

Reste enfin à prendre en compte le bruit de mesure sur la position du point. Pour cela, on convolue notre modèle avec une loi gaussienne isotrope. Comme elle est isotrope, on peut donc appliquer la convolution en y puis la convolution en x. La première augmente simplement la variance de y :

$$\sigma^2_{y|x} = \sigma^2_{forme}.x^{2A} + \sigma^2_{n_m}.x^2 + \sigma^2_m$$

Pour le lissage gaussien en x, les calculs seraient plus complexes. Mais ce dernier lissage a moins d'influence car les variations de la probabilité en x sont moins importantes qu'en y. Nous nous contenterons donc de cette expression, pour éviter des calculs trop complexes.

Calcul des paramètres sur une mesure empirique de la dispersion On peut vouloir estimer les paramètres A, σ^2_{forme}, $\sigma^2_{n_m}$ et σ^2_m lorsqu'on a tracé une courbe empirique de la variance $\sigma^2_{y|x}$ en fonction de x (voir section 8.7.5). La méthode que nous utilisons est la suivante :

- On fixe dans un premier temps $A = 2$. On a alors $\sigma^2_{y|x} = a + b.(x^2) + c.(x^2)^2$, et on estime la parabole qui interpole au mieux les couples $(x^2,\sigma^2_{y|x})$. Ceci nous donne une première estimée de nos paramètres : $\sigma^2_m = a$, $\sigma^2_{n_m} = b$, $\sigma^2_{forme} = c$ et $A = 2$.
- On optimise ensuite nos paramètres par un algorithme aux moindres carrés (*lsqcurvefit* dans *matlab*), qui, à partir de cette première estimée, minimise le critère $\sum_i \left(\sigma^2_{y|x_i} - \left(\sigma^2_{forme}.x_i^{2A} + \sigma^2_{n_m}.x_i^2 + \sigma^2_m\right)\right)^2$, sous les contraintes σ^2_{forme}, $\sigma^2_{n_m}$, σ^2_m positifs et $A \geq 1$.

Bibliographie

[Abrahams and Kalyanpur, 1995] J. Abrahams and A. Kalyanpur. Dental implant and dental CT software programs. *Seminar in ultrasound, CT and MRI*, 16(6):468–486, 1995.

[Amenta and Bern, 1999] N. Amenta and M. Bern. Surface reconstruction by voronoi filtering. *Discrete and Computational Geometry*, 22(4):481–504, 1999.

[Arun *et al.*, 1987] K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[Assémat-Tessandier and Amzalag, 1993] X. Assémat-Tessandier and G. Amzalag. *La décision en implantologie*. CdP, Guide Clinique, 1993.

[Audette *et al.*, 2000] M.A. Audette, F.P. Ferrie, and T.M. Peters. An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis*, 2000. in press.

[August, 2001] J. August. *The Curve Indicator Random Field*. PhD thesis, Yale university, 2001.

[August, 2002] Jonas August. Volterra filtering of noisy images of curves. In *European Conference on Computer Vision (ECCV'02)*, volume 2352 of *LNCS*, pages 604–620, Copenhagen, Denmark, 2002. Springer.

[Ayache, 1991] N. Ayache. *Artificial vision for mobile robots - Stereo-vision and multisensory perception*. MIT-Press, 1991.

[Bass, 1991] S. Bass. The effects of pre-operative resorption and jaw anatomy on implant success: a report of 303 cases. *Clin Oral Impl Res*, 1(4):193–198, 1991.

[Bellaiche and Doyon, 1992] N. Bellaiche and D. Doyon. La tomodensimetrie dans le bilan preoperatoire en implantologie orale. *Journal de radiologie*, 73(1):23–33, 1992.

[Ben-Jemaa, 1998] R. Ben-Jemaa. *Traitement de données 3D denses acquises sur des objets réels complexes*. PhD thesis, Télécom Paris - France, 1998.

[Bergevin *et al.*, 1996] B. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multiview registration technique. *Computer Vision and Image Understanding*, 1(61):1–16, 1996.

[Besl and McKay, 1992] P.J. Besl and N.D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[Blais and Levine, 1995] G. Blais and M. Levine. Registering multiview range data to create 3D computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, 1995.

[Boissonnat and Cazals, 2000] Jean-Daniel Boissonnat and Frederic Cazals. Smooth surface reconstruction via natural neighbour interpolation of distance functions. In *Symposium on Computational Geometry*, pages 223–232, 2000.

[Bowyer *et al.*, 2001] K.W. Bowyer, M.H. Loew, H. S. Stiehl, and M. Viergever, editors. *Methodology of Evaluation in Computational Medical Imaging*. Schloss Dagsthul, Int. Conf. and Research Center for Computer Science, March 2001. Dagstuhl Seminar 01111.

[Branemark *et al.*, 1969] P. Branemark, U. Breien, R. Adell, B. Hansson, H. Lindstrom, and A. Ohlsson. Intraosseous anchorage of dental prostheses. *Scand J Plastic Reconst Surg*, 3(21):81–100, 1969.

[Branemark *et al.*, 1977] P. Branemark, B. Hansson, R. Adell, and al. Osseointegrated implants in the treatment of edntulous jaw: experiences aver 10 years period. *Scand J Plastic Reconst Surg*, suppl 16, 1977.

[Branemark *et al.*, 1985] P. Branemark, G. Zarb, and T. Albreckisson. *Tissue integrated prosthesis. Osseointegration in clinical dentistery*. Quintescence, 1985.

[Brunnstrom and Stoddart, 1995] K. Brunnstrom and A.J. Stoddart. Genetic algorithms for free-form surface matching. In *International Conference on Pattern Recognition (ICPR'96)*, 1995.

[Campbell and Flynn, 1999] R.J. Campbell and P.J. Flynn. Eigenshapes for 3D object recognition in range data. In *International Conference on Computer Vision and Pattern Recognition (CVPR'99)*, 1999.

[Carmo, 1992] M.d. Carmo. *Riemannian Geometry*. Birhauser (Boston, Basel, Berlin), 1992.

[Chen and Medioni, 1991] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. In *Proc. CRA'91*, pages 2724–2729, 1991.

[Chen and Medioni, 1992] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10:145–155, 1992.

[Chui and Rangarajan, 2000] H. Chui and A. Rangarajan. A feature registration framework using mixture models. In *Proc. MMBIA'2000*, pages 190–197, 2000.

[Cohen-Steiner and Morvan, 2002] David Cohen-Steiner and Jean-Marie Morvan. Restricted delaunay triangulations, normal cycle, and principal curvatures. Technical Report ECG-TR-184100-02, INRIA Sophia-Antipolis, 2002.

[Couvreur, 1996] C. Couvreur. The EM algorithm: A guided tour. In *Proc. 2d IEEE European Workshop on Computationaly Intensive Methods in Control and Signal Processing (CMP'96)*, pages 115–120, Pragues, Czech Republik, August 1996.

[Cunnington and Stoddart, 1999] S. J. Cunnington and A. J. Stoddart. N-view point set registration: A comparison. In *British Machine Vision Conference (BMVC'99)*, volume 1, pages 234–244, 1999.

[Delingette, 1999] Herve Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, (32):111–142, 1999.

[Dempster *et al.*, 1977] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood for incomplete data via the EM algorithm. *Journal of the royal statistical society B*, 39:1–38, 1977.

[Duda *et al.*, 2001] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification - Second Edition*. Wiley Interscience, 2001.

[Duncan *et al.*, 1991] J.S. Duncan, R.L. Owen, L.H. Staib, and P. Anandan. Measurement of non-rigid motion using contour shape descriptor. In *International Conference on Computer Vision and Pattern Recognition (CVPR'91)*, 1991.

[Dutreuil, 2001] J. Dutreuil. *Modélisation 3D et robotique médicale pour la chirurgie : développement d'un protocole d'implantologie dentaire assistée par ordinateur et étude d'un cas clinique*. PhD thesis, Ecole des Mines de Paris - France, 2001.

[Eckert and Wollan, 1998] S. Eckert and P. Wollan. Retrospective study of 1170 endosseous implants placed in partially edentulous jaws. *J Prosthet Dent*, 79(4):415–420, 1998.

[Eggert *et al.*, 1997] D.W. Eggert, A. Lorusso, and R.B. Fisher. Estimating 3-D rigid body transformations : A comparison of four major algorithms. *Machine Vision and Applications*, 9:272–290, 1997.

[Etienne and al., 2000] D. Etienne and al. A new approach for dental implant aided surgery. a pilot evaluation. In *(CARS'00)*, pages 927–931, 2000.

[Faugeras and Hebert, 1986] O. Faugeras and M. Hebert. The representation and recognition and locating of 3D objects. *Int. Jour. Rob. Researchs*, 5(3):27–52, 1986.

[Feldmar and Ayache, 1996] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. *The International Journal of Computer Vision*, 18(2), May 1996.

[Feldmar *et al.*, 1997a] J. Feldmar, N. Ayache, and F. Betting. 3D-2D projective registration of free-form curves and surfaces. *Journal of Computer Vision and Image Understanding*, 65(3):403–424, 1997.

[Feldmar *et al.*, 1997b] J. Feldmar, J. Declerck, G. Malandain, and N. Ayache. Extension of the ICP algorithm to non-rigid intensity-based registration of 3D volumes. *Computer Vision and Image Understanding*, 66(2):193–206, May 1997.

[Feldmar, 1995] J. Feldmar. *Recalage rigide, non rigide et projectif d'images médicales tridimensionnelles.* PhD thesis, Ecole Polytechnique - France, 1995.

[Field *et al.*, 1993] D.J. Field, A. Hayes, and R. Hess. Contour integration by the human visual system: evidence for a local 'association field'. *Vision Research*, 33:173–193, 1993.

[Fitzpatrick *et al.*, 1999] J. M. Fitzpatrick, J. B. West, and Jr. C. R. Maurer. Predicting error in rigid-body point-based registration. *IEEE Transactions on Medical Imaging*, 17(5):694–702, 1999.

[Fortin *et al.*, 1995] T. Fortin, J.L. Coudert, G. Chapleboux, P. Sautot, and S. Lavallee. Profile surgical template: a systematic approach to precise implant placement. a technical note. *J Image Guided Surg*, 1:53–58, 1995.

[Garrigues, 1999] J. Garrigues. Eléments d'algèbre et d'analyse tensorielle à l'usage des mécaniciens, 1999. http://esm2.imt-mrs.fr/gar/tenseurs.html.

[Gramkow, 2001] Claus Gramkow. On averaging rotations. *International Journal of Computer Vision*, 42(1):7–16, 2001.

[Granger and Pennec, 2002a] S. Granger and X. Pennec. Statistiques exactes et approchées sur les normales aléatoires. Research report RR-4533, INRIA, 2002.

[Granger and Pennec, 2002b] Sébastien Granger and Xavier Pennec. Multi-scale EM-ICP: A fast and robust approach for surface registration. In *European Conference on Computer Vision (ECCV'02)*, volume 2353 of *LNCS*, pages 418–432, Copenhagen, Denmark, 2002. Springer.

[Granger *et al.*, 2001a] S. Granger, X. Pennec, and A. Roche. Rigid point-surface registration using an EM variant of ICP for computer guided oral implantology. In W.J. Niessen and M.A. Viergever, editors, *4th Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI'01)*, volume 2208 of *LNCS*, pages 752–761, Utrecht, The Netherlands, October 2001.

[Granger *et al.*, 2001b] S. Granger, X. Pennec, and A. Roche. Rigid point-surface registration using oriented points and an EM variant of ICP for computer guided oral implantology. Research report RR-4169, INRIA, 2001.

[Granger *et al.*, 2003] S. Granger, J. Dutreuil, and X. Pennec. An experimental comparison of the accuracy of rigid registration in computer-guided surgery systems for dental implantology. In *(CARS'03)*, 2003. accepted.

[Guy and Medioni, 1997] G. Guy and G. Medioni. From sparse noisy 3D data to structured surface descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11), 1997.

[Guy, 1996] G. Guy. *Inference of multiple curves and surfaces from sparse data.* PhD thesis, USC-IRIS, 1996.

[Guéziec and Ayache, 1991] A. Guéziec and N. Ayache. Smoothing and matching of 3D space curves. Technical report, INRIA, 1991.

[Guéziec and Ayache, 1994] A. Guéziec and N. Ayache. Smoothing and matching of 3D space curves. *International Journal of Computer Vision*, 12(1), 1994.

[Hügli and Schütz, 1997] H. Hügli and C. Schütz. Geometric matching of 3D objects: Assessing the range of successful initial configuration. In *International Conference on Recent Advances in 3D digital imaging and modeling*, pages 101–112, 1997.

[Hobkirk and Havthoulas, 1998] J. Hobkirk and T. Havthoulas. The influence of mandibular deformation, implant numbers, and loading position on detected forces in abutments supporting fixed implant superstructures. *J Prosthet Dent*, 80:169–174, 1998.

[Horn *et al.*, 1988] B. Horn, M. Hilon, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Optical Society of America*, 5(7):1125–1137, 1988.

[Horn, 1987] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *Optical Society of America*, 4(4):629–642, 1987.

[Hussaini and Canela Pichardo, 1997] S. Hussaini and D. Canela Pichardo. Palatal impression template for a fully edentulous arch during stage I implant placement. *J. Prosthet Dent*, 77:630–632, 1997.

[Jacobs and van Steenberghe, 1998] R. Jacobs and D. van Steenberghe. *Radiographic planning and assessment of endosseous oral implants.* Berlin: Springer Verlag, 1998.

[Jannin *et al.*, 2002] P. Jannin, J.M. Fitzpatrick, D.J. Hawkes, X. Pennec, R. Sahidi, and M.W. Vannier. White paper: validation of medical image processing in image-guided therapy. In *(CARS'02)*, pages 299–305, 2002.

[Johnson and Hebert, 1998] A.E. Johnson and M. Hebert. Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing*, 16:635–651, 1998.

[Jolion, 1994] J-M. Jolion. *A Pyramid Framework for Early Vision.* Kluwer Academic, 1994.

[Kanatani, 1996] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice.* Elsevier Science (Amsterdam), 1996.

[Klingenberg, 1982] W. Klingenberg. *Riemannian Geometry.* Walter de Gruyter (berlin, New-York), 1982.

[Lautrou, 1998] A. Lautrou. *Anatomie dentaire, Abrégés Odonto-stomatologie, 2ème édition.* Masson, 1998.

[Lee, 2002] M.S. Lee. Communication personnelle, 2002.

[Lindeberg, 1994] T. Lindeberg. *Scale-Space Theory in Computer Vision.* Kluwer academic, 1994.

[Lorensen and Cline, 1987] W. Lorensen and H. Cline. Marching cube: a high resolution 3D surface constructing algorithm. In *(SIGGRAPH'87)*, pages 163–169, juillet 1987.

[Luo and Hancock, 1999] B. Luo and E.R. Hancock. Matching point-sets using procrustes alignment and the EM algorithm. In *British Machine Vision Conference (BMVC'99)*, volume 1, pages 43–52, 1999.

[Maintz and Viergever, 1998] J.B.A. Maintz and M.A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[Matei and Meer, 1999] B. Matei and P. Meer. Optimal rigid motion estimation and performance evaluation with bootstrap. In *International Conference on Computer Vision and Pattern Recognition (CVPR'99)*, volume 1, pages 339–345, 1999.

[McInerney and Terzopoulos, 1996] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[McLachlan and Krishnan, 1997] G.J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley (New-York), 1997.

[Médioni *et al.*, 2000] G. Médioni, M.S. Lee, and C.K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier Science, 2000.

[Metz, 1978] C.E. Metz. Basic principles of ROC analysis. *Semin. Nucl. Med.*, 8(4):283, 1978.

[Montagnat and Delingette, 2001] J. Montagnat and H. Delingette. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing*, 19(14):1023–1040, December 2001.

[Mumford, 1994] D. Mumford. Chapter elastica and computer vision. In *Algebraic Geometry and its Applications*, pages 491–506. Springer-Verlag, 1994.

[Neal and Hinton, 1998] R.M. Neal and G.E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in Graphical Models*, 1998.

[Nicolau *et al.*, 2003] S. Nicolau, X. Pennec, L. Soler, and N. Ayache. Evaluation of a new 3D/2D registration criterion for augmented reality in hepatic surgery. In *IS4TM'03*, 2003. submitted.

[Nicolau, 2001] S. Nicolau. Et bingo !, 2001.

[Nicolescu and Médioni, 2002] M. Nicolescu and G. Médioni. Perceptual grouping form motion cues using tensor voting in 4-D. In *European Conference on Computer Vision (ECCV'02)*, pages 423–437, 2002.

[Ourselin *et al.*, 2001] S. Ourselin, A. Roche, G. Subsol, X. Pennec, and N. Ayache. Reconstructing a 3D Structure from Serial Histological Sections. *Image and Vision Computing*, 19(1-2):25–31, January 2001.

[Ourselin, 2002] Sébastien Ourselin. *Recalage d'images médicales par appariement de régions - Application à la construction d'atlas histologiques 3D.* Thèse de sciences, Université de Nice Sophia-Antipolis, January 2002.

[Pennec,] X. Pennec. Probabilities and statistics on riemanian manifolds: A geometric approach. unpublished.

[Pennec *et al.*, 2000] X. Pennec, N. Ayache, and J.-P. Thirion. Chap. 31: Landmark-based registration using features identified through differential geometry. In I. Bankman, editor, *Handbook of Medical Imaging*, pages 499–513. Academic Press, September 2000.

[Pennec, 1996] X. Pennec. *L'incertitude dans les Problemes de Reconnaissance et de Recalage - Applications en Imagerie Medicale et Biologie Moleculaire.* PhD thesis, Ecole Polytechnique - France, 1996.

[Penney *et al.*, 2001] G.P. Penney, P.J. Edwards, A.P. King, J.M. Blackall, P.G. Batchelor, and D.J Hawkes. A stochastic iterative closest point algorithm (stochastICP). In Springer, editor, *Medical Image Computing and Computer-Assisted Intervention (MICCAI'01)*, volume LNCS 2208, pages 762–769, 2001.

[Picard, 1994] Jean Picard. Barycentres et martingales sur une variété. *Annales de l'Institut Henri Poincaré (Probabilités et Statistiques)*, 30(4):647–702, 1994.

[Prima, 2001] Sylvain Prima. *Étude de la symétrie bilatérale en imagerie cérébrale volumique.* PhD thesis, université Paris XI, Orsay, 2001.

[Quirynen *et al.*, 1990] M. Quirynen, Y. Lamoral, C. Dekeyser, P. Peene, D. Steenberghe, J. Bonte, and A. Baert. The CT scan standard reconstruction technique for reliable jaw bone volume determination. *Int J Oral Maxillofacial Implants*, 5:384–389, 1990.

[Rangarajan *et al.*, 1997a] A. Rangarajan, H. Chui, and F. Bookstein. The softassign procrustes matching algorithm. In *International Conference on Information Processing in Medical Images (IPMI'99)*, pages 29–42, 1997.

[Rangarajan *et al.*, 1997b] A. Rangarajan, H. Chui, E. Mjolsness, S. Pappu, L. Davachi, P. Goldman-Rakic, and J. Duncan. A robust point-matching algorithm for autoradiograph alignment. *Medical Image Analysis*, 1(4):379–398, 1997.

[Rangarajan *et al.*, 1999] A. Rangarajan, H. Chui, and J.S. Duncan. Rigid point feature registration using mutual information. *Medical Image Analysis*, 3(4):425–440, 1999.

[Rice and Silverman, 1991] J. A. Rice and B. W. Silverman. Estimating the mean and covariance structure nonparametrically when the data are curves. *J. of the Royal Statistical Soc. B*, 53(1):233–243, 1991.

[Roche *et al.*, 2000] A. Roche, G. Malandain, and N. Ayache. Unifying Maximum Likelihood Approaches in Medical Image Registration. *International Journal of Imaging Systems and Technology: Special Issue on 3D Imaging*, 11(1):71–80, 2000.

[Roche, 2001] Alexis Roche. *Recalage d'images médicales par inférence statistique.* Thèse de sciences, Université de Nice Sophia-Antipolis, February 2001.

[Rothman *et al.*, 1987] S. Rothman, N. Chaftez, M. Rhodes, and al. CT in preoperative assessment of the mandible and maxillia for endosseous implant surgery. *RSNA*, 168:171–175, 1987.

[Rousseeuw and Leroy, 1987] P.J. Rousseeuw and A.M. Leroy. *Robust regression and outlier detection.* Ed. John Wiley and Sons, 1987.

[Rusinkiewicz and Levoy, 2001] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM'01)*, 2001.

[Schnebelen, 1995] J. Schnebelen. Arrêt du vieillissement des maxillaires et de leur fonte osseuse, rétablissement des fonctions bucco-dentaires par l'utilisation des implants dentaires selon branemark. *Revue de Gériatrie*, 2(21):131–134, 1995.

[Schonenenberger, 1998] C. Schonenenberger. *Recalage de séquences spatiales d'images 3D en vue de l'analyse et de la reconstruction dimensionnelles de surfaces libres.* PhD thesis, Université Louis Pasteur de Strasbourg, 1998.

[Sethian, 1996] J.A. Sethian. A review of the theory, algorithms, and applications of level set methods for propagating interfaces. *Acta Numerica*, pages 309–395, 1996.

[Sha'ashua and Ullman, 1988] A. Sha'ashua and S. Ullman. Structural saliency: The detection of globally salient structures using a locally connected network. In *International Conference on Computer Vision (ICCV'88)*, pages 321–327, 1988.

[Shapira, 2002] Lior Shapira. real-time guidance of dental implant surgery in the operative field using CT-scan image. In *(CARS'02)*, pages 959–964, 2002.

[Sicilia *et al.*, 1998] A. Sicilia, B. Noguerol, J. Cobo, and I. Zabalegui. Profile surgical template: a systematic approach to precise implant placement. a technical note. *Int J Oral Maxillofacial Implants*, 13:109–114, 1998.

[Simon, 1996] Simon. *Fast and accurate shape-based registration.* PhD thesis, Carnegie Mellon University, Pittsburgh, USA, 1996.

[Stoddart and Hilton, 1996] A. Stoddart and A. Hilton. Registration of multiple point sets. In *International Conference on Pattern Recognition (ICPR'96)*, 1996.

[Tang and Medioni, 1999] C.K. Tang and G. Medioni. Robust estimation of curvature information form noisy 3D data for shape description. In *International Conference on Computer Vision (ICCV'98)*, pages 426–433, 1999.

[Taylor, 1998] T. Taylor. Prosthodontic problems and limitations associated with osseointegration. *J Prosthet Dent*, 79:74–78, 1998.

[Thévenaz *et al.*, 1998] P. Thévenaz, U.E. Ruttimann, and M. Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Transactions on Image Processing*, 7(1):27–41, 1998.

[Thornber and Williams, 1996] K.K. Thornber and L.R. Williams. Analytic solution of stochastic completion field. *Biological Cybernetics*, 75:141–151, 1996.

[Umeyama, 1991] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

[Van Leemput *et al.*, 1999] K. Van Leemput, F. Maes, D. Vandermeulen, and P. Suetens.

Automated model-based bias field correction of mr images of the brain. *IEEE Transactions on Medical Imaging*, 18(10):885–896, 1999.

[Vannier *et al.*, 1997] M. Vannier, C. Hildeboldt, G. Conover, R. Knapp, N. Yokoyama-Crothers, and G. Wang. Three-dimensional dental imaging by spiral CT. *Oral Surg Oral Med Oral Pathol, Oral Radiol Endodont*, 84:561–570, 1997.

[Walker *et al.*, 1991] M. Walker, L. Shao, and R. Volz. Estimating 3D location parameters using dual number quaternions. *Computer Vision, Graphics, and Image Processing*, 54(3):358–367, 1991.

[Wells, 1997] W. Wells. Statistical approaches to feature-based object recognition. *International Journal of Computer Vision*, 21(1):63–98, 1997.

[West *et al.*, 1997] J. West, J. M. Fitzpatrick, M. Y. Wang, B. M. Dawant, C. R. Maurer, Jr., R. M. Kessler, R. J. Maciunas, C. Barillot, D. Lemoine, A. Collignon, F. Maes, P. Suetens, D. Vandermeulen, P. A. van den Elsen, S. Napel, T. S. Sumanaweera, B. Harkness, P. F. Hemler, D. L. G. Hill, D. J. Hawkes, C. Studholme, J. B. A. Maintz, M. A. Viergever, G. Malandain, X. Pennec, M. E. Noz, G. Q. Maguire, Jr., M. Pollack, C. A. Pelizzari, R. A. Robb, D. Hanson, and R. P. Woods. Comparison and evaluation of retrospective intermodality brain image registration techniques. *Journal of Computer Assisted Tomography*, 21:554–566, 1997.

[Wiener, 1976] Norbert Wiener. *Collected Works - Vol. 1.* MIT Press, 1976.

[Williams and Jacobs, 1997] L. Williams and D. Jacobs. Stochastic completion field: A neural model of illusory contour shape and salience. *Neural Computation*, 9(4):837–858, 1997.

[Williams and Thornber, 2000] L.R. Williams and K.K. Thornber. A comparison of measures for detecting natural shapes in cluttered background. *International Journal of Computer Vision*, 34(2/3):81–96, 2000.

[Woods, 2000] R.P. Woods. Chap. 30: Validation of registration accuracy. In I. Bankman, editor, *Handbook of Medical Imaging*, pages 491–497. Academic Press, September 2000.

[Zhang, 1992] Z. Zhang. Iterative point matching for registration of free-form surfaces. Technical Report 1658, INRIA, 1992.

[Zhang, 1993] Z. Zhang. Le probleme de la mise en correspondence: L'etat de l'art. Technical report, INRIA, 1993.

[Zhang, 1994] Z. Zhang. Iterative point matching for registration of free-form surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[Zweck and Williams, 2000] John W. Zweck and Lance R. Williams. Euclidean group invariant computation of stochastic completion fields using shiftable-twistable functions. In *European Conference on Computer Vision (ECCV'00)*, pages 100–116, 2000.

Résumé

Le principal sujet de cette thèse est la mise au point d'algorithmes de recalage rigide de surfaces au sein de VirtualScope, un système de guidage per-opératoire dédié au percement des axes des implants dentaires.

Elle est basée sur une approche purement statistique, qui, en essayant de maximiser la vraisemblance calculée à partir d'une modélisation explicite du bruit, permet de justifier l'utilisation de l'ICP pour le recalage d'amers géométriques, puis de proposer l'ICP/EM multi-échelles, un peu plus précis et surtout beaucoup plus robuste et rapide. De nouveaux modèles de bruits sont proposés pour adapter l'algorithme aux surfaces échantillonnées et bruitées. La prédiction théorique de l'incertitude est abordée, et permet en particulier de guider l'acquisition des données. L'étude expérimentale très poussée des performances de l'algorithme permet de régler efficacement ses paramètres, mettre au point des systèmes de sécurité, et garantir ainsi un fonctionnement parfaitement satisfaisant au sein de VirtualScope.

La seconde partie de cette thèse aborde plus généralement le problème de la modélisation statistique des courbes et surfaces échantillonnées bruitées. En regroupant les travaux sur la saillance et le vote de tenseurs, elle présente la notion de champs de vote, qui permet d'exprimer la probabilité d'un élément de la courbe ou surface connaissant un autre élément. Elle donne des exemples rudimentaires mais facilement programmables de champs de votes, qui prennent en compte la forme de la surface et la façon dont les points ont été échantillonnés et bruités. Elle montre comment les appliquer avec succès au problème du recalage, puis indique comment ils pourraient servir pour dériver des algorithmes bayésiens pour de nombreuses autres applications concernant les courbes et surfaces. Ces travaux seraient alors susceptibles de déboucher sur la mise au point d'un canevas statistique et multi-échelles commun à toutes ces méthodes.

Mots-clés: recalage rigide, statistiques, multi-échelles, validation, ICP, EM, surfaces échantillonnées bruitées, décimation, normales, saillance, vote de tenseurs, guidage per-opératoire

Abstract

The main subject of this work is the rigid registration of surfaces dedicated to VirtualScope, a per-operative guiding system designed for oral implants surgery.

It is based on a purely statistical approach. We first show how to compute and maximize a likelihood, based on a model of the data noise, for the landmarks registration problem. This approach justifies the use of the ICP algorithm, and a new multi-scale variant named ICP/EM, which improves accuracy, speed and robustness. We introduce new noise models specifically designed for the registration of sampled and noised surfaces. We discuss about the theoretical prediction of the registration accuracy, and use it for guiding the data acquisition. We analyze in detail the experimental performances of the algorithm, and provide methods for setting optimally the parameters and ensuring the correctness of registration results. The resulting algorithm is perfectly suited to the VirtualScope application.

The second part of this work deals with the more general problem of the statistical modelisation of sampled and noised curves and surfaces. Based on previous works on Saliency and Tensor Voting notions, it defines a vote field that represent the probability of a curve or surface element, knowing another element. We provide basic yet easy to implement examples of such a field, which can handle the surface shape, the sampling strategy and the measurement errors. We apply them successfully to the registration problem, and suggest to use them to derive Bayesian methods to virtually all other computer vision problems involving sampled and noised curves and surfaces. This work could lead to the design of a common statistical and multi-scale framework for these various methods.

Keywords: rigid registration, statistics, multi-scale, validation, ICP, EM, sampled and noised surfaces, decimation, normals, saliency, tensor-voting, per-operative computer-guided surgery