THÈSE

dirigée par Patrick Rives
préparée à l'INRIA Sophia Antipolis
au sein du projet ICARE

et présentée à

L'ÉCOLE NATIONALE SUPÉRIEURE DES MINES DE PARIS
SOPHIA ANTIPOLIS

pour obtenir le grade de

DOCTEUR EN SCIENCES

Spécialité

Informatique Temps Réel, Automatique et Robotique

soutenue publiquement par

Christopher MEI

Couplage Vision Omnidirectionnelle et Télémétrie Laser pour la Navigation en Robotique

---

Laser-Augmented Omnidirectional Vision for 3D Localisation and Mapping

le Vendredi 9 Février 2007 devant le jury composé de :

M.	**Yves**	**ROUCHALEAU**	Président
MM.	**Simon**	**LACROIX**	Rapporteur
	El Mustapha	**MOUADDIB**	Rapporteur
MM.	**Delphine**	**DUFOURD**	Examinateur
	David	**FILLIAT**	Examinateur
	Patrick	**RIVES**	Directeur de Thèse
	Roland	**SIEGWART**	Examinateur

To my parents.

Remerciements

Je tiens à remercier:

- Monsieur Yves Rouchaleau pour avoir accepté de présider mon jury de thèse,
- Messieurs Simon Lacroix et El Mustapha Mouaddib pour leur travail de rapporteurs,
- Madame Delphine Dufour, Messieurs David Filliat et Roland Siegwart pour avoir examiné ma thèse,
- la DGA pour le financement de ce travail de thèse.

Mes remerciements vont tout particulièrement à Patrick Rives, mon directeur de thèse, pour m'avoir donné beaucoup de liberté dans mon travail tout en me faisant profiter de sa grande expérience et de sa sagesse. Un grand merci aussi à Ezio Malis pour son travail collaboratif, ses discussions enrichissantes et son amitié ainsi qu'à Selim Benhimane. Merci à Claude Samson et Pascal Morin pour leurs conseils avisés. Je ne saurais oublier tous mes amis, les "anciens" : Alessandro, Nicolas, Guillaume, Matthieu, Mauro, Geraldo, Andrew et Omar et la "vieille troupe" : Benoît, Marie-Claude, Damiana, Alberto, Baptiste et Céline. Merci aussi à Patricia pour sa gentillesse. La relève pleine d'idées fraîches (comme des tubes de dentifrice) promet de garder une excellente ambiance au travail avec Adan, Benoît, Gabi, Hicham, Minh-Duc et Youssef. Un merci tout particulier à Cyril pour ses relectures et pour avoir relié mes derniers manuscrits... Enfin je remercie mes parents, ma sœur et Shalima pour leur amour et soutien de tous les instants.

Synthèse en Français

Contenu

Objectif

Estimer le mouvement d'un robot et construire en même temps une représentation de l'environnement (problème connu sous le nom de SLAM : Simultaneous Localisation And Mapping) est souvent considéré comme un problème essentiel pour développer des robots pleinement autonomes qui ne nécessitent pas de connaissances a priori de l'environnement pour réaliser leurs tâches.

L'évolution du SLAM est très liée aux capteurs utilisés. Les sonars avec l'odométrie sont souvent présentés comme les premiers capteurs ayant fourni des résultats convaincants. Depuis, les lasers 2D ont souvent remplacé ces capteurs pour des raisons de précision et de rapport signal/bruit. Néanmoins les lasers 2D permettent uniquement d'estimer des mouvements planaires et ne donnent pas des informations perceptuelles suffisantes pour identifier de manière fiable des régions précédemment explorées.

Pour répondre à ces enjeux, les chercheurs se sont penchés sur d'autres capteurs permettant d'estimer le mouvement 3D d'un robot et de reconnaître des lieux. Les caméras perspectives classiques répondent à ces deux critères mais souffrent du risque d'occlusion à cause d'un angle de vue faible et nécessitent d'estimer la profondeur des amers pour la construction de cartes métriques. Les télémètres laser 3D sont une autre alternative mais sont peu discriminants et ont actuellement une vitesse d'acquisition faible.

Ces observations nous ont amenés à explorer à travers cette thèse comment combiner un capteur omnidirectionnel à grand angle de vue avec un télémètre laser pour effectuer de la localisation et de la cartographie simultanée dans des environnements complexes et de grandes tailles.

Les contributions de cette thèse concernent l'étalonnage des capteurs centraux catadioptriques [Mei and Rives, 2006a, 2007] (avec le développement d'un logiciel open-source disponible sur le site internet de l'auteur) et la recherche de la position relative entre un capteur omnidirectionnel et un télémètre laser [Mei and Rives, 2006b]. Des approches efficaces pour estimer le mouvement 3D du capteur en utilisant des droites [Mei and Malis, 2006] et des plans [Mei et al., 2006a,b] sont détaillées avec notamment l'utilisation des algèbres de Lie pour obtenir une représentation minimale. Enfin deux méthodes sont proposées combinant laser et vision pour effectuer du SLAM planaire mais aussi pour estimer la position 3D du robot ainsi que la structure de l'environnement.

1 La vision omnidirectionnelle : introduction, modèle de projection et étalonnage

1.1 La vision omnidirectionnelle

1.1.1 Obtenir un grand angle de vue

Un grand angle de vue peut être obtenu par différents moyens, un état de l'art est présenté dans [Yagi, 1999] :

- reconstitution à partir de plusieurs images,
- utilisation de lentilles grands-angles,
- utilisation de miroirs convexes.

Contraintes temps-réel La reconstitution d'images omnidirectionnelles à partir de plusieurs images peut être réalisée avec plusieurs caméras ou une caméra rotative. Néanmoins l'acquisition est coûteuse en calcul et en temps et ne répond donc pas aux contraintes temps-réel présentes en robotique.

Objectif grand-angle Les objectifs grands-angulaires (*fish-eye lenses*) permettent d'obtenir des images omnidirectionnelles en temps-réel. Ils présentent cependant l'inconvénient d'avoir une résolution qui n'est bonne qu'au centre de l'image, la résolution en périphérie restant faible. De plus, les lentilles ne vérifient pas strictement la contrainte de centre optique. Le champ de vue est souvent inférieur à 180° .

Caméras multiples regardant vers l'extérieur ou caméras multiples regardant vers des miroirs plans Ces deux configurations présentent l'avantage d'avoir une très bonne résolution à l'horizontal avec une acquisition d'images en temps réel. Cependant la reconstruction d'images panoramiques génèrent des calculs importants. Il est aussi difficile de positionner les caméras avec suffisamment de précision pour être sûr d'obtenir un centre de projection unique pour le système optique ainsi formé. L'encombrement est aussi un défaut supplémentaire de ce type de système.

Miroirs convexes Le principe des miroirs convexes est de placer une caméra en position verticale pointant vers un miroir convexe qui renvoie une image de 360° de l'espace environnant. Cette approche permet à la fois de conserver les propriétés d'acquisition en temps-réel et aussi d'obtenir des images de bonne résolution en périphérie (figures 1 et 2). Ces capteurs nécessitent d'adapter les algorithmes de traitement classiques au modèle de projection non-linéaire et à la résolution non-uniforme du capteur.

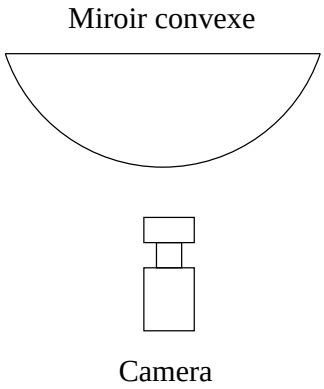


Figure 1: Capteur catadioptrique

Figure 2: Capteur à miroir parabolique

Pour ces raisons, les capteurs omnidirectionnels formés de miroirs convexes appelés couramment capteurs catadioptriques [1] sont des capteurs grands-angles de plus en plus utilisés en robotique. Nous nous intéresserons par la suite principalement à ce type de capteurs.

1.1.2 Miroirs à centre de projection unique

Les miroirs à centre de projection unique sont les capteurs qui relient un point de l'image à un unique rayon projectif. Ils présentent l'avantage par exemple de permettre d'effectuer des transformations pour obtenir des images panoramiques sans distorsions.

[1] *dioptrique* étant la partie de la physique portant sur la réfraction de la lumière (lentille) et *catoptrique* celle portant sur les surfaces réfléchissantes (miroirs)

Dans [Baker and Nayar, 1998], il est démontré, en utilisant le modèle de sténopé de la caméra qu'il n'existe que certaines formes de miroirs qui permettent d'obtenir des capteurs catadioptriques centraux. Ces formes correspondent à trois quadriques de révolution "classiques" : ellipsoïdes, paraboloïdes et hyperboloïdes ainsi que le cas planaire[2] (voir figure 3 et table 1 adaptés de [Barreto, 2003, p. 10]). Dans le cas d'une paraboloïde, les rayons qui arrivent au centre focal du capteur ressortent parallèlement à l'axe optique. Il est alors nécessaire d'utiliser une caméra orthographique. Pour le cas des ellipses et hyperboles, un rayon qui arrive vers le premier point focal est réfléchi vers le deuxième point focal où l'on peut placer une caméra perspective.

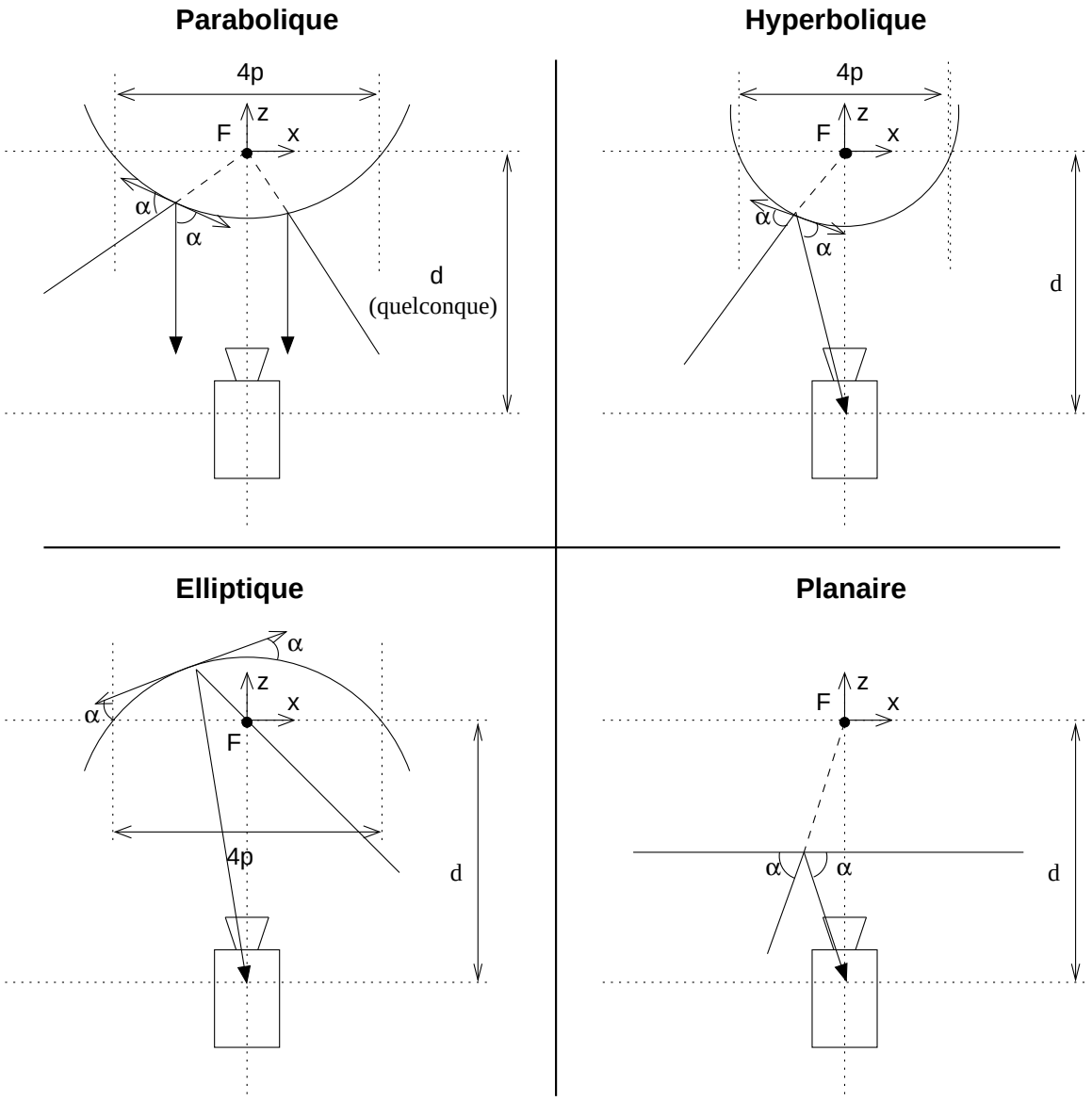


Figure 3: Ensemble des capteurs omnidirectionnels avec un centre de projection unique

[2]par abus de langage nous utiliserons par la suite les termes d'ellipse, de parabole et d'hyperbole

Table 1: Équation correspondant aux coniques

Paraboloïde	$\sqrt{x^2+y^2+z^2} = z + 2p$
Hyperboloïde	$\frac{(z+\frac{d}{2})^2}{a^2} - \frac{x^2}{b^2} - \frac{y^2}{b^2} = 1$
Ellipsoïde	$\frac{(z+\frac{d}{2})^2}{a^2} + \frac{x^2}{b^2} + \frac{y^2}{b^2} = 1$
Plan	$z = -\frac{d}{2}$

$a = 1/2(\sqrt{d^2+4p^2} \pm 2p)$, '$-$' pour l'hyperbole, '$+$' pour l'ellipse

$b = \sqrt{p(\sqrt{d^2+4p^2} \pm 2p)}$, '$-$' pour l'hyperbole, '$+$' pour l'ellipse

Geyer et Daniilidis [Geyer and Daniilidis, 2000] ainsi que Barreto [Barreto, 2003, Chap. 2] ont développé une théorie qui unifie l'étude de tels miroirs. Une extension de ce modèle ainsi qu'une méthode d'étalonnage est présentée dans la Section 1.2.2.

Un exemple de miroir parfois utilisé et qui apporte des distorsions est le miroir sphérique (qui est un cas particulier de l'ellipsoïde de révolution). En effet, une sphère ne contient qu'un seul point focal qui est situé au centre de la sphère. En décalant la caméra, une caustique apparaît [Swaminathan et al., 2001]. En d'autres termes, ce système ne vérifie plus la contrainte d'un centre de projection unique. Néanmoins, ce type de capteur est simple à réaliser et le montage est aisé car tout rayon de la sphère est un axe de révolution et donc un axe optique. Le modèle projection proposé a permis de calibrer un capteur de ce type.

1.1.3 Choisir un capteur catadioptrique

Nous verrons par la suite que le modèle de projection d'un capteur parabolique est théoriquement plus simple que celui d'un miroir hyperbolique. Néanmoins, dans la pratique, pour simuler une caméra orthographique, une lentille télécentrique est rajoutée entre le miroir et la caméra. Cette lentille doit avoir le même diamètre que le miroir. Cette taille importante rend difficile la production de lentilles de bonnes qualités optiques et introduit de la distorsion. Cette distorsion doit être prise en compte dans le modèle de projection qui n'est alors pas plus simple que dans le cas hyperbolique.

1.2 Modèle de projection

Nous allons détailler dans cette partie le modèle de projection choisi. Nous ne donnerons pas les étapes pour obtenir le modèle unifié sur la sphère à partir de la projection sur les miroirs. Le lecteur intéressé peut se référer aux thèses de Geyer [Geyer, 2003] ou Barreto [Barreto, 2003].

1.2.1 Projections perspectives planaires et sphériques

L'approche la plus classique pour modéliser la projection d'un point 3D dans le plan image pour une caméra perspective est l'utilisation du plan normalisé. Les étapes de la projection (figure 4) sont :

1. soit $(\boldsymbol{\mathcal{X}})_{\mathcal{R}_c} = (X, Y, Z)$ un point 3D dans le repère de la caméra, le point est projeté dans le plan normalisé, π_m :

$$(\boldsymbol{\mathcal{X}})_{\mathcal{R}_c} \longrightarrow \mathbf{m} = (x, y, 1) = (\frac{X}{Z}, \frac{Y}{Z}, 1)$$

2. avec f_1 la distance focale horizontale, f_2 la distance focale verticale, s le facteur d'obliquité et (u_0, v_0) le point principal, nous obtenons linéairement la projection de $\mathbf{m}$ dans le plan image, π_p :

$$\mathbf{p} = (u, v, 1) = \mathbf{K}\mathbf{m} = \begin{bmatrix} f_1 & f_1 s & u_0 \\ 0 & f_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{m} = k(\mathbf{m})$$

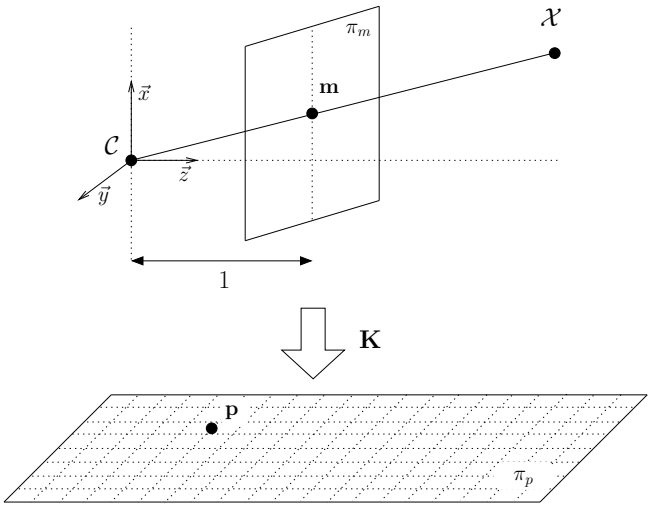


Figure 4: Projection perspective planaire

Dans le cas d'un grand angle de vue (>180°), ce modèle n'est plus adapté. En effet, avec un plan unique nous avons une ambiguïté entre "l'arrière" et "l'avant" de la caméra. L'utilisation de la projection sur une sphère (que nous prendrons unité pour simplifier les calculs) résout ces problèmes. Figure 6 illustre la projection perspective sphérique où la projection à partir de la sphère est suivie d'une projection non-linéaire par une fonction Π. Dans la section suivante nous détaillerons une fonction Π adaptée aux capteurs centraux catadioptriques.

1.2.2 Modèle unifié

Geyer [Geyer, 2003] et Barreto [Barreto, 2003] ont proposé un modèle de projection valable pour tous les capteurs centraux catadioptriques. Il peut être montré que ce modèle est aussi valable pour certaines lentilles fisheye (Section 2.2.2.3). Par contre le modèle ne permet pas de prendre en compte des distorsions radiales souvent présentes dans le cas de capteurs paraboliques où une lentille télécentrique est rajoutée pour pouvoir utiliser une caméra perspective (et non orthographique). Un modèle de distorsion radial a été rajouté pour prendre en compte ce cas. Des paramètres de distorsion tangentielle ont aussi été rajoutés pour modéliser des erreurs d'alignement faibles (le modèle de distorsion est détaillé dans la Section 2.3.1). Les étapes de projection proposées sont illustrées dans la figure 7 avec les paramètres en relation avec la forme du miroir dans la table 2.

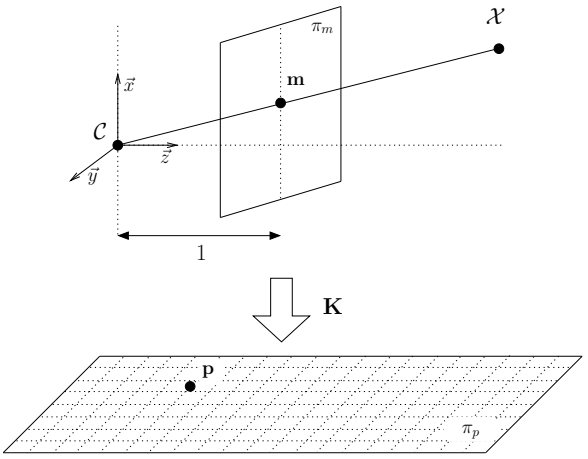


Figure 5: Projection perspective planaire

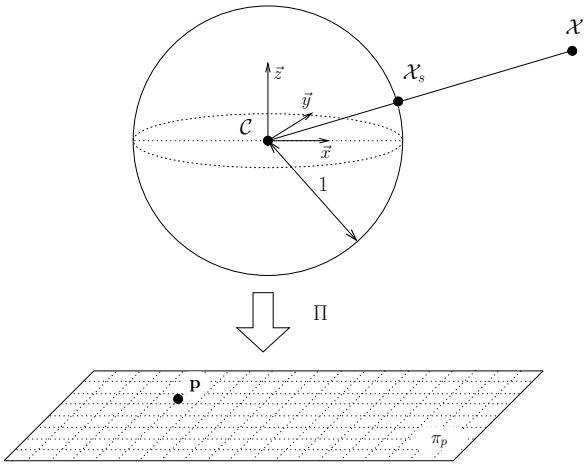


Figure 6: Projection perspective sphérique

Table 2: Paramètres du modèle unifié

	ξ	γ
Parabole	1	$-2pf$
Hyperbole	$\frac{df}{\sqrt{d^2+4p^2}}$	$\frac{-2pf}{\sqrt{d^2+4p^2}}$
Ellipse	$\frac{df}{\sqrt{d^2+4p^2}}$	$\frac{2pf}{\sqrt{d^2+4p^2}}$
Planaire	0	-f
Perspectif	0	f
d : distance entre points focaux $4p$: latus rectum		

1.3 Étalonnage de capteurs à centre de projection unique

1.3.1 Paramètres du modèle

Les paramètres de la fonction Π que nous souhaitons estimer sont au nombre de 10 :

1. ξ qui dépend de la géométrie du miroir,
2. k_1, k_2, p_1 et p_2 qui sont les paramètres modélisant la distorsion du miroir et les problèmes d'alignement,
3. γ_1, γ_2, s, u_0 et v_0 les paramètres de la caméra généralisée (en effet γ_1 et γ_2 contiennent aussi des informations sur la forme du miroir)

Pour pouvoir estimer les paramètres nous allons utiliser des mires planaires de dimension connue et minimiser l'erreur entre la reprojection des points de la grille et l'extraction de ces points dans l'image. La fonction de projection est non-linéaire et nous devons donc initialiser les paramètres avant d'effectuer la minimisation.

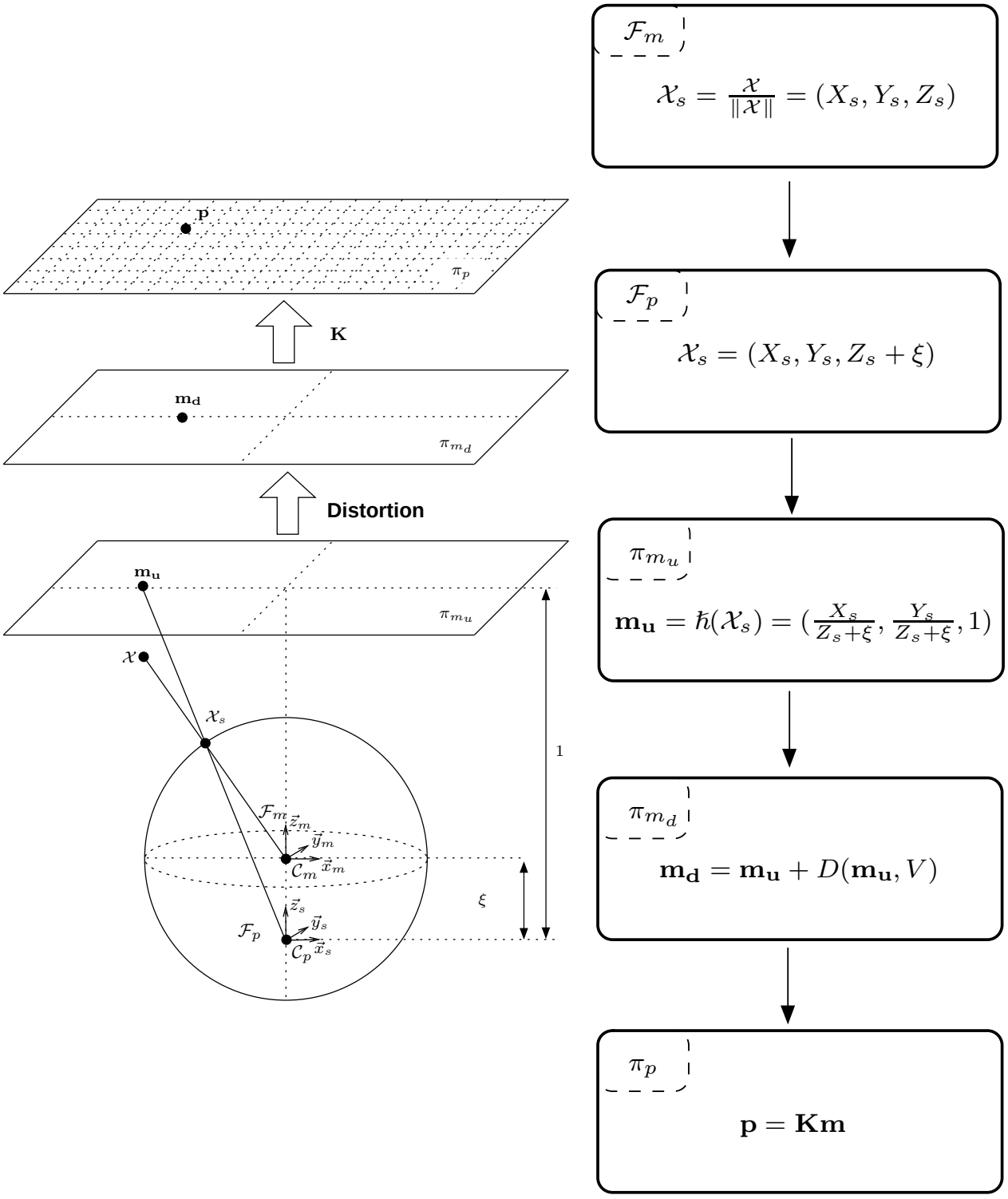


Figure 7: Modèle de projection complet

Expérimentalement, nous constatons que ξ n'a pas une influence très importante sur l'erreur de reprojection, nous utiliserons comme valeur initiale $\xi = 1$. Les valeurs de k_1, k_2, p_1 et p_2 correspondent à des erreurs de modèle que nous supposons faibles et initialiserons par la valeur nulle. De même, $s \approx 0$ et $\gamma_1 \approx \gamma_2$.

Par contre les valeurs de γ, u_0 et v_0 doivent être estimées ainsi que les paramètres extrinsèques correspondant à la position relative (rotation et translation) entre les grilles planaires et le miroir. La rotation sera représentée par un quaternion $\mathbf{Q} = [q_0\ q_1\ q_2\ q_3]^\top$ et une translation $\mathbf{t} = [t_x\ t_y\ t_z]^\top$.

Notons V la matrice des paramètres :

$$V_{17\times1} = [q_0\ q_1\ q_2\ q_3\ t_x\ t_y\ t_z\ \xi\ k_1\ k_2\ p_1\ p_2\ s\ \gamma_1\ \gamma_2\ u_0\ v_0]^\top$$

$$V^1_{7\times1} = [q_0\ q_1\ q_2\ q_3\ t_x\ t_y\ t_z]^\top,\ V^2_{1\times1} = \xi,\ V^3_{4\times1} = [k_1\ k_2\ p_1\ p_2]^\top,\ V^4_{5\times1} = [s\ \gamma_1\ \gamma_2\ u_0\ v_0]^\top$$

1.3.2 Méthodologie pour l'étalonnage

Les étapes suivantes permettent d'extraire les points des mires et d'initialiser les paramètres :

1. initialisation du point principal (u_0, v_0) grâce à la bordure du miroir (figure 8),
2. estimation de la distance focale généralisée γ (en supposant que $\gamma = \gamma_1 = \gamma_2$) grâce à trois points (ou plus) alignés sur l'image d'une droite non-radiale de la scène (figure 9),
3. pour chaque image de la mire, nous sélectionnons les quatre coins de la grille (figure 10), estimons les paramètres extrinsèques puis extrayons les points restants par reprojection (figure 11),
4. nous terminons l'étalonnage par la minimisation globale de l'erreur de reprojection (par exemple en utilisant l'algorithme de Levenberg-Marquardt).

Figure 8: Extraction de la bordure du miroir pour estimer le point principal

Les différentes étapes sont détaillées dans le Chapitre 3. Un des points importants est l'initialisation de la distance focale grâce à un modèle simplifié. C'est cette étape qui va permettre par la suite de mettre en correspondance les points 3D de la grille avec leur images.

Figure 9: Estimation de la distance focale généralisée grâce à des points alignés dans la scène

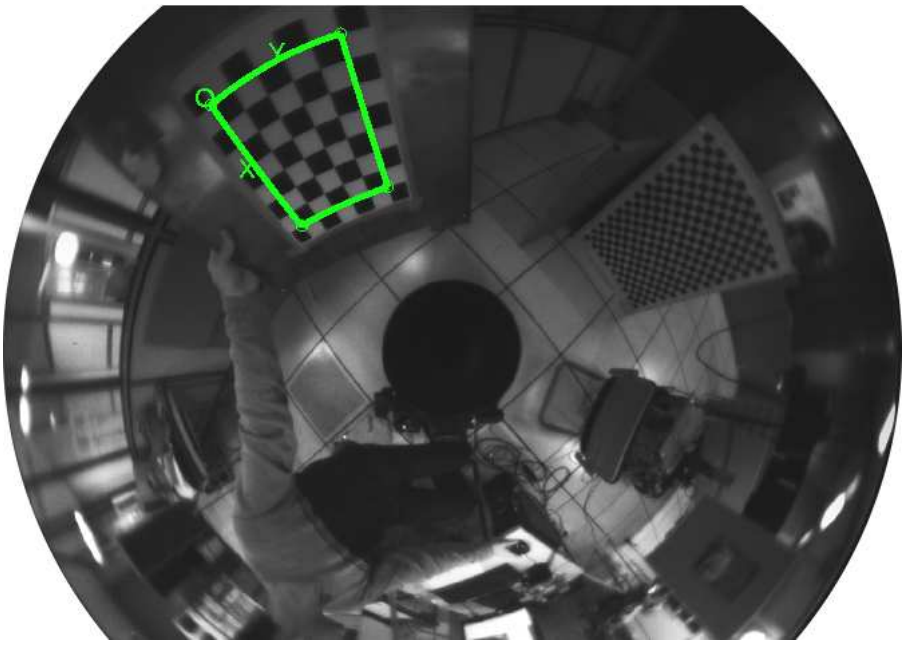


Figure 10: Extraction de quatre coins de la grille

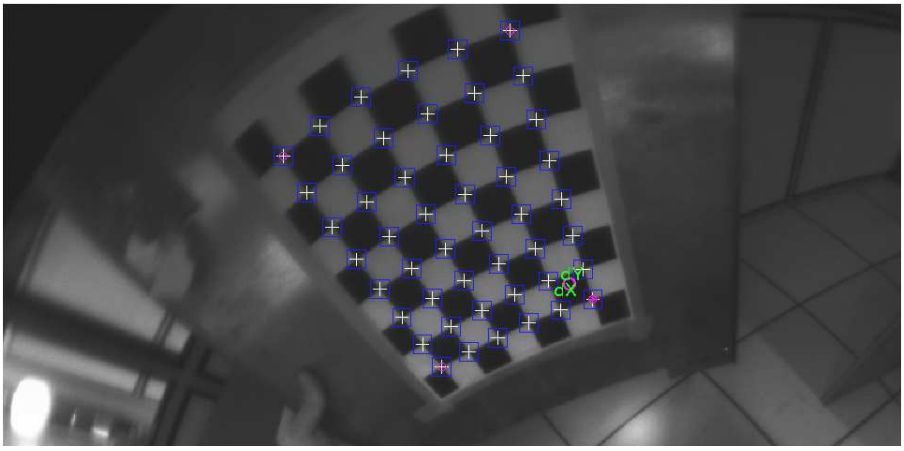


Figure 11: Extraction sub-pixellique des points

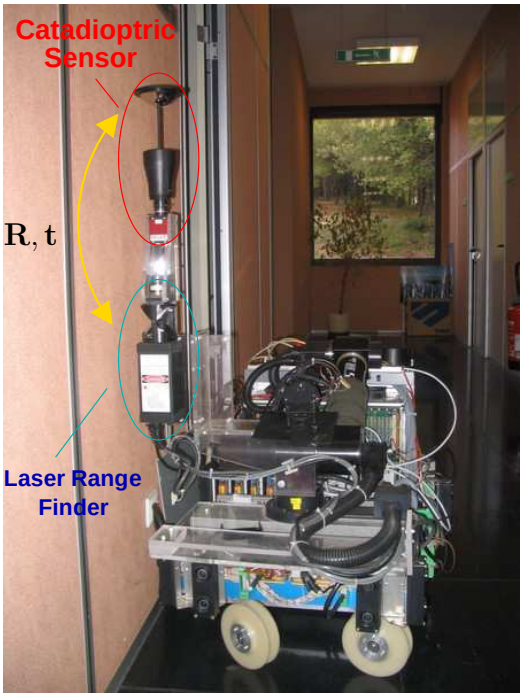


Figure 12: Étalonnage entre un télémètre laser et un miroir omnidirectionnel

1.4 Étalonnage entre un télémètre laser et un miroir omnidirectionnel

Nous nous intéressons ici à l'étalonnage entre un télémètre laser et un capteur omnidirectionnel, c'est-à-dire trouver la position relative (rotation $\mathbf{R}$ et translation $\mathbf{t}$) entre les deux capteurs (figure 12). Nous supposons que chaque capteur a été étalonné de manière séparée.

Nous envisagerons deux cas distincts : le cas où le laser est visible dans l'image et le cas où il est invisible (proche infrarouge).

Dans le premier cas, nous étudierons l'estimation de la position relative lorsque les points laser

3D peuvent être associés directement à des points dans l'image et le cas où nous pouvons associer des droites extraites dans la coupe laser avec leurs images dans le capteur omnidirectionnel.

Dans le deuxième cas, nous nous poserons la question de savoir si l'association entre des points de rupture dans les données laser avec des droites omnidirectionnelles sont suffisantes pour étalonner le capteur. Nous verrons que ce cas n'est pas favorable et proposerons une alternative en utilisant la position de plans 3D visibles dans la coupe laser.

Dans cette synthèse, nous montrerons quelques résultats, plus de détails peuvent être trouvés au Chapitre 4.

1.4.1 Laser visible

Points laser Lorsque le faisceau laser est visible dans l'image, nous nous trouvons en présence du problème classique d'estimation de pose en vision par ordinateur.

Figure 13 illustre le cas de l'association entre des points laser 3D et des points dans l'image. Ce cas est analogue aux problèmes PnP [Fischler and Bolles, 1981] (Perspective from n Points) ou "position relative avec modèle 3D". Il faut néanmoins prendre soin de travailler sur la sphère et non dans un plan (comme expliqué dans la Section 2.1.2).

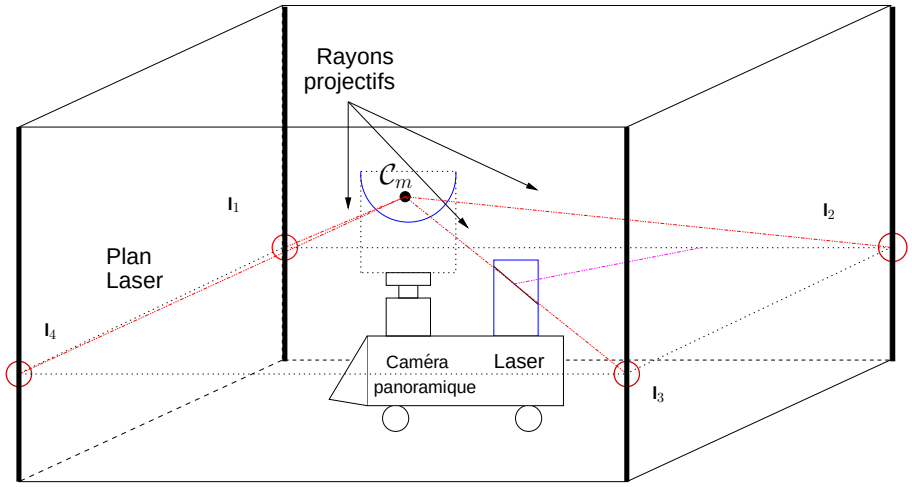


Figure 13: Association entre points laser 3D et points dans l'image

Figure 14 montre des mesures laser effectuées dans l'environnement et figure 15 le résultat final après extraction des points dans l'image et étalonnage.

Segments laser Certains télémètres laser ne permettent pas d'effectuer des mesures à des angles donnés mais renvoient directement des coupes lasers complètes. Dans ce cas, nous pouvons associer des segments extraits de la coupe laser à leur images dans le capteur omnidirectionnel (figure 16).

Cette approche est plus difficile à mettre en œuvre et à automatiser que dans le cas des points. En effet, il est nécessaire de mettre en correspondance les droites du plan laser avec les droites de l'image qui est un problème de complexité exponentielle. Figure 17 montre des segments extraits de la coupe laser et figure 18 leur reprojection dans l'image.

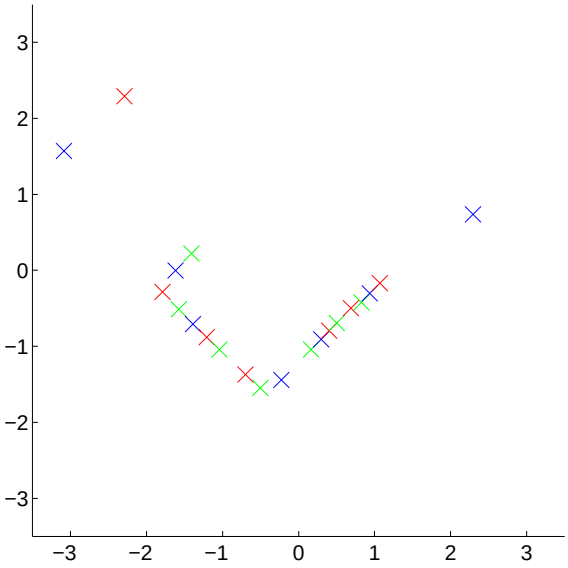


Figure 14: Mesures laser effectuées de l'environnement

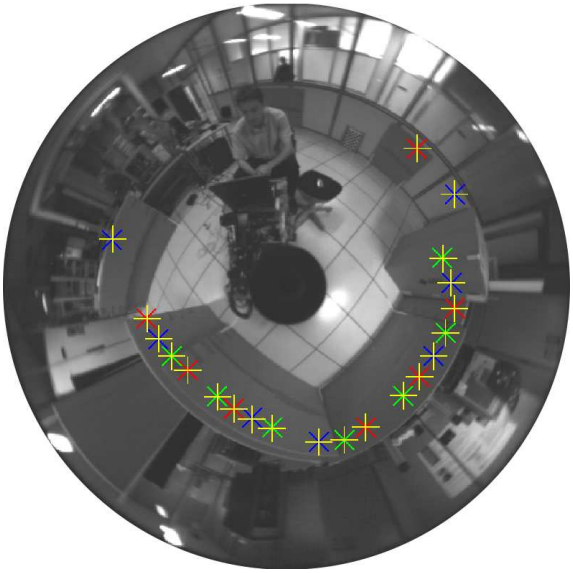


Figure 15: Points laser extraits (×) et points laser 3D reprojetés après étalonnage (+)

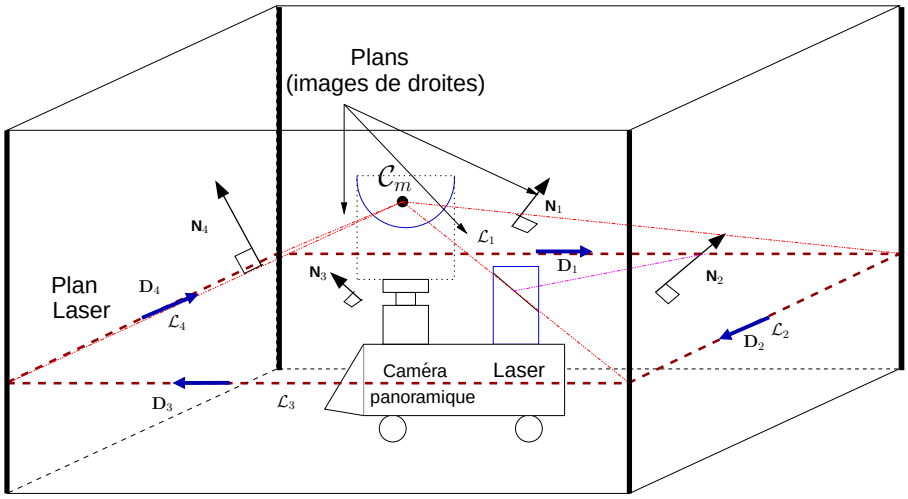


Figure 16: Association entre segments extraits de la coupe laser et leur projections dans l'image

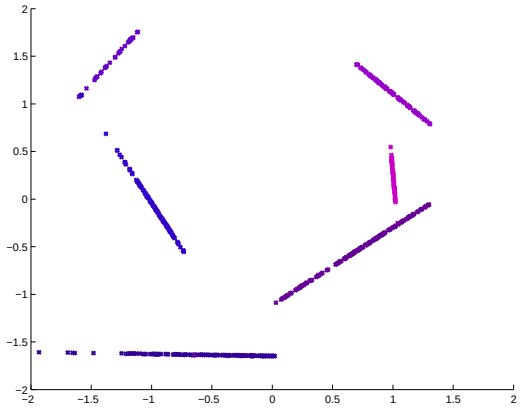


Figure 17: Extraction de segments de la coupe laser

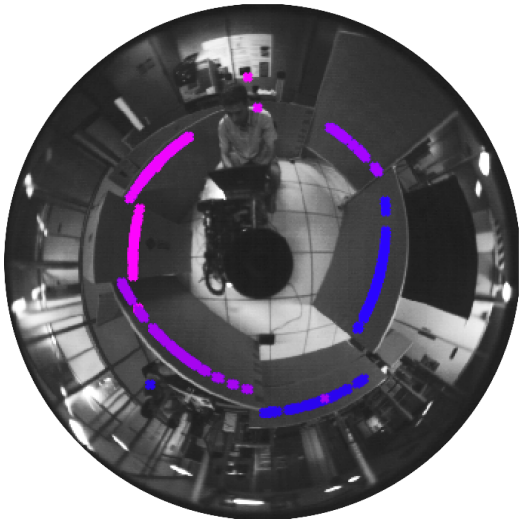


Figure 18: Extraction des images des segments

1.4.2 Laser invisible

De nombreux lasers émettent des faisceaux dans le proche infrarouge qui ne peuvent être vus par la caméra. Les approches précédentes ne sont donc pas applicables.

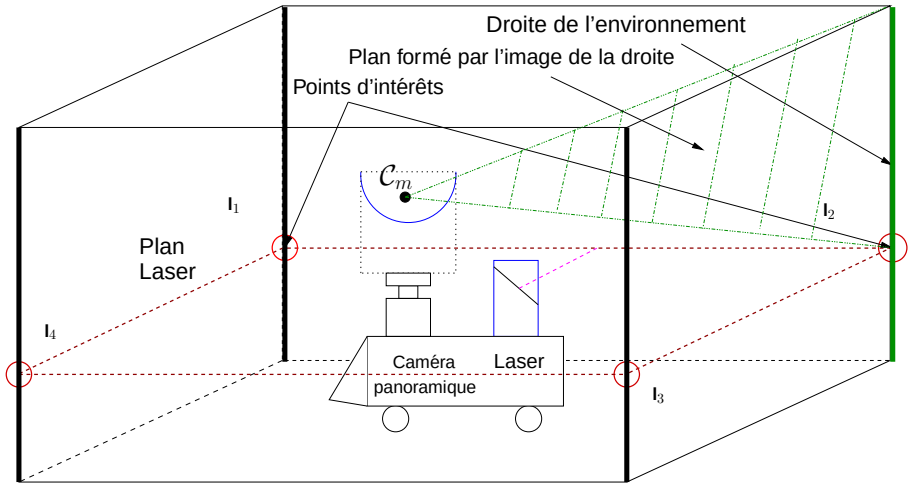


Figure 19: Association entre points de rupture laser et droites dans l'image

Points de rupture Si nous associons des points de rupture dans les données laser avec des droites omnidirectionnelles (figure 19) nous avons $6 + 3n$ $(\mathbf{R}, \mathbf{t}, \boldsymbol{\mathcal{X}}_1, ..., \boldsymbol{\mathcal{X}}_n)$ inconnues et $3n + n$ équations. 6 points sont donc nécessaires pour minimiser le système. En réalité 6 points ne sont pas suffisants et quelque soit le nombre de points, le système est sous-contraint car les équations ne sont pas indépendantes (Section 4.3). Une autre méthode d'étalonnage utilisant des informations 3D est

nécessaire.

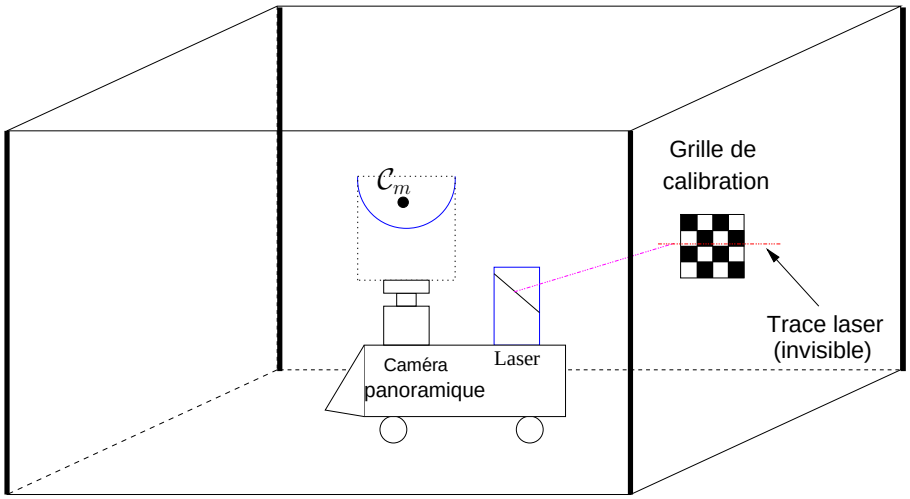


Figure 20: Association entre points laser et plans 3D

Plans 3D Pour étalonner les capteurs, nous pouvons utiliser des informations 3D comme la position de plan illustrées par la figure 20).

Le système d'équations obtenu permet alors de trouver la position relative (figure 21).

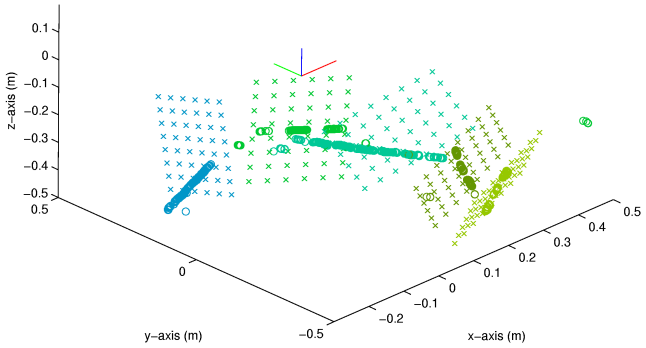


Figure 21: Vue 3D des plans utilisés pour l'étalonnage avec la reprojection des points laser

1.5 Conclusion

Les capteurs centraux catadioptriques sont des systèmes de vision qui présentent l'avantage d'un grand angle de vue obtenu en temps réel et sans parallaxe.

Pour raisonner avec ces types de capteurs, la représentation perspective planaire utilisée classiquement pour les capteurs à angle de vue faible n'est pas adaptée. Nous avons montré par contre qu'une représentation perspective sphérique permet d'exprimer correctement les points devant et derrière le capteur.

Le modèle de projection des capteurs centraux catadioptriques a ensuite été étendu pour prendre en compte les problèmes d'alignement et de distorsion souvent présent en pratique. Une méthodologie a été présentée qui permet d'estimer les paramètres du capteur et rend aisée la mise en correspondance de points. Un logiciel d'étalonnage a été développé et permet d'étalonner efficacement un grand nombre de capteurs grands-angles présents en robotique.

Nous avons ensuite présenté plusieurs approches pour trouver la position relative entre un capteur laser et une caméra omnidirectionnelle. Dans le cas où le capteur laser est visible et nous pouvons effectuer des mesures télémétriques ponctuelles, l'étalonnage est facilement automatisable. Pour des lasers avec des faisceaux invisibles, il est nécessaire d'utiliser de l'information 3D comme la position de plans.

2 Estimation du mouvement à partir d'une caméra centrale catadioptrique

2.1 Représentation minimale

De nombreuses transformations (rotations, homographies) et objets géométriques (plans, droites 3D) utilisés en vision par ordinateur et en robotique peuvent être paramétrés en utilisant des groupes de Lie dont une représentation minimale - au moins locale - existe. Ces paramétrisations permettent de s'assurer que dans des problèmes de minimisation les objets manipulés restent dans les groupes étudiés. Le Chapitre 5 résume certaines propriétés des algèbres de Lie et des matrices exponentielles.

Un point important est la représentation de ces entités dans les problèmes d'optimisations.

Soit G un groupe de Lie matriciel de dimension n, soit :

$$\begin{array}{rccc} f: & G & \longrightarrow & \mathbb{R} \\ & \mathbf{g} & \longmapsto & f(\mathbf{g}) \end{array}$$

Considérons le problème de minimisation suivant, avec d une distance différentiable (typiquement une norme L_2) et $\overline{\mathbf{f}} \in \mathbb{R}$:

$$\overline{\mathbf{g}} = \min_{\mathbf{g}} d(f(\mathbf{g}), \overline{\mathbf{f}})$$

Si f est une fonction non-linéaire, le problème n'a souvent pas de solution explicite et une méthode de descente de gradient est couramment employée. Nous partons d'une solution initiale de valeur $\widehat{\mathbf{g}}$ et à chaque étapes nous rajoutons une valeur $\mathbf{g}_k$ calculée à partir du jacobien, par exemple : $\widehat{\mathbf{g}} \leftarrow \widehat{\mathbf{g}} + \mathbf{g}_k$. Le problème d'une telle approche est que nous ne pouvons garantir que la nouvelle valeur $\widehat{\mathbf{g}}$ va appartenir au groupe G. Pour résoudre ce problème, la nouvelle valeur $\widehat{\mathbf{g}}$ est souvent projetée sur la variété mais ceci peu dégrader à la fois la vitesse mais aussi la région de convergence.

Une alternative est de définir une nouvelle fonction h. Avec $\mathfrak{g}$ l'algèbre de Lie de G et $+$ l'opération de groupe.

$$\begin{array}{rccccc} h: & \mathbb{R}^n & \longrightarrow & \mathfrak{g} & \longrightarrow & \mathbb{R} \\ & \mathbf{x} & \longmapsto & G(\mathbf{x}) & \longmapsto & f(\widehat{\mathbf{g}} + e^{G(\mathbf{x})}) \end{array}$$

h est seulement défini *localement* par la paramétrisation de l'algèbre de Lie de G. Si nous appliquons une méthode de descente de gradient à h en partant de $\mathbf{x} = 0$ (qui correspond à la valeur initiale f), la mise à jour s'écrit : $\widehat{\mathbf{g}} \leftarrow \widehat{\mathbf{g}} + e^{G(\mathbf{x}_k)}$. Nous sommes maintenant assurés qu'à chaque étape la nouvelle valeur de $\widehat{\mathbf{g}}$ appartient au groupe de Lie G.

Notons qu'il faut aussi s'assurer qu'il existe un chemin reliant $\widehat{\mathbf{g}}$ à $\overline{\mathbf{g}}$ (voir Chapitre 5).

Cette technique de minimisation est très importante pour la localisation et la cartographie simultanée et apparaît dans les problèmes étudiés par la suite dans cette thèse lors de la cartographie à partir de plans et de droites.

2.2 Suivi basé vision avec une caméra omnidirectionnelle

Dans cette thèse, nous nous sommes intéressés aux techniques de suivi dense incrémental illustrées par la figure 22. A partir d'une région extraite d'une image de référence notée $\mathcal{I}^*$ nous calculons de manière séquentielle la transformation entre la position de référence et la position courante $\mathcal{I}_k$. L'avantage d'une telle approche est la possibilité d'estimer le mouvement sans dérive et uniquement à partir de données images. Dans cette thèse, nous avons exploré le suivi de région de l'image correspondant à des plans de la scène. La transformation de points appartenant à un même plan suit une homographie planaire.

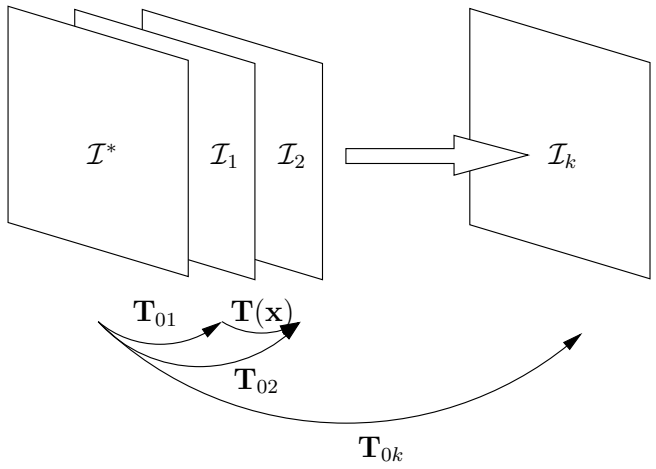


Figure 22: Calcul incrémental de la transformation

2.2.1 Homographies planaires

Soit $\mathbf{R} \in \mathbb{SO}(3)$ la matrice correspondant à la rotation de la caméra et soit $\mathbf{t} \in \mathbb{R}^3$ la translation. Une homographie planaire $\mathbf{H}$ est définie à un facteur d'échelle près :

$$\mathbf{H} \sim \mathbf{R} + \mathbf{t}\mathbf{n}_d^{*\top} \tag{1}$$

où $\mathbf{n}_d^* = \mathbf{n}^*/d^*$ est le rapport entre la normale au plan $\mathbf{n}^*$ (vecteur unitaire) et la distance d^* du plan à l'origine du repère de référence. Par la suite, par abus de langage, nous appellerons $\mathbf{n}_d^*$ la normale au plan. Les homographies sont des propriétés projectives et restent donc valables pour tous les capteurs centraux catadioptriques. Figure 23 illustre la transformation engendrée par une homographie planaire en utilisant le modèle perspectif sphérique. Les points $\boldsymbol{\mathcal{X}}_s^*$ et $\boldsymbol{\mathcal{X}}_s$ sont reliés par:

$$\exists(\rho, \rho^*) \in \mathbb{R}^2, \;\; \boldsymbol{\mathcal{P}} = \rho\boldsymbol{\mathcal{X}}_s = \rho^*\mathbf{H}\boldsymbol{\mathcal{X}}_s^*$$

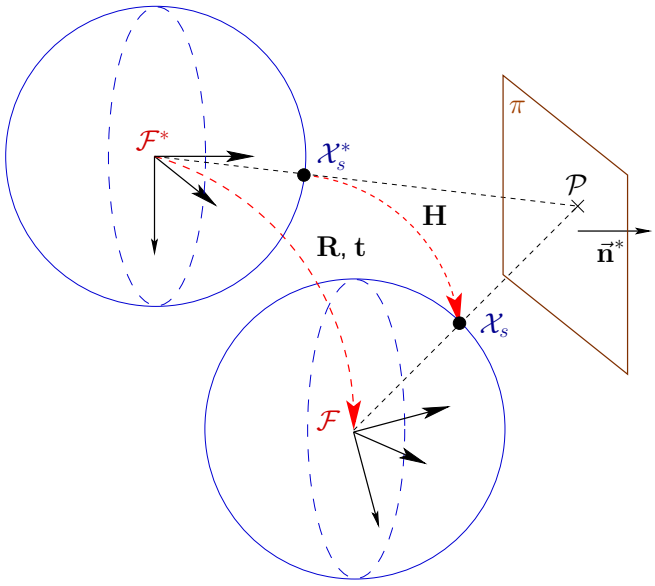


Figure 23: Homographie planaire avec un modèle perspectif sphérique

2.2.2 Suivi basé vision

Soit $\mathbf{W}$ le warping (transformation entre différentes vues) engendré par une homographie (avec S la normalisation sur la sphère) :

$$\mathbf{W}(\mathbf{H}, \boldsymbol{\mathcal{X}}^*) = S(\mathbf{H}\boldsymbol{\mathcal{X}}^*)$$

Le suivi basé vision consiste à trouver les transformations $\overline{\mathbf{H}}$ et et $\mathbf{H}(\overline{\mathbf{T}}, \overline{\mathbf{n}}_d^j)$ optimales telles que :

- pour un plan : $\forall i, \quad \mathcal{I}\left(\Pi(\mathbf{W}(\overline{\mathbf{H}}, \Pi^{-1}(\mathbf{p}_i)))\right) = \mathcal{I}^*(\mathbf{p}_i)$ avec $\overline{\mathbf{H}} \in \mathbb{SL}(3)$
- pour n $(n > 1)$ plans : $\forall (i,j), \quad \mathcal{I}\left(\Pi(\mathbf{W}(\mathbf{H}(\overline{\mathbf{T}}, \overline{\mathbf{n}}_d^j), \Pi^{-1}(\mathbf{p}_{ij})))\right) = \mathcal{I}^*(\mathbf{p}_{ij})$ avec $\overline{\mathbf{T}} \in \mathbb{SE}(3)$, $\overline{\mathbf{n}}_d \in \mathbb{R}^3$

Dans cette thèse, nous avons exploré comment minimiser efficacement ces équations dans un cadre valable pour tous les capteurs centraux (Chapitre 6).

Les contributions dans ce domaine concernent la technique de minimisation employée ESM (Efficient Second-order Minimisation) avec des variantes avec de meilleures propriétés en terme de complexité (αESM; iESM). L'estimation en ligne des normales dans le cas de plusieurs plans n'avait pas été étudiée au préalable. Les résultats expérimentaux présentés dans la section 6.5.2 montrent la validité de l'approche pour l'estimation du mouvement et de la structure de l'environnement.

2.3 Droites omnidirectionnelles

Dans des environnements structurés, les droites sont des amers visuels qui peuvent aider dans l'estimation du mouvement. Les algorithmes classiques utilisés avec des caméras perspectives doivent néanmoins être adaptés.

Le Chapitre 7 explore les différentes étapes pour extraire des droites de la scène projetées dans des images omnidirectionnelles, estimer leurs paramètres et permettre d'effectuer un suivi d'image à

image. Nous proposons aussi une représentation non-singulière des droites à partir de leur normales et les segments à partir d'un point, un angle et une normale (à travers la formule de Rodrigues). Figure 24 illustre la représentation "projective" de la projection d'une droite de la scène choisie pour cette étude.

Dans ce chapitre, nous étudions aussi le problème de minimisation qui apparaît dans le problème de cartographie à partir de droites dans la scène. Nous détaillons en particulier plusieurs distances possibles adaptées aux capteurs centraux catadioptriques.

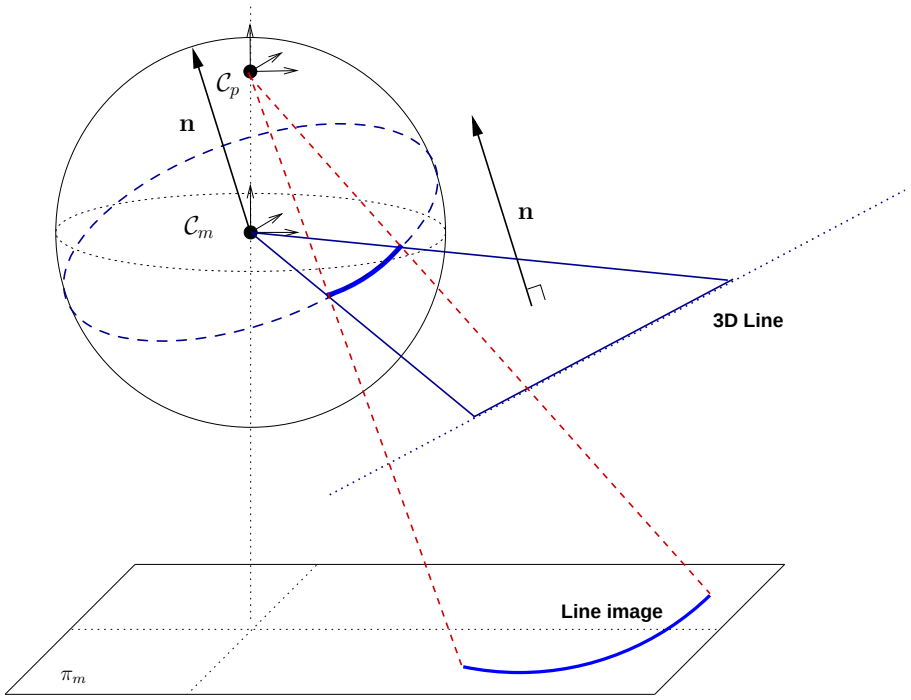


Figure 24: Représentation d'une droite de la scène par une normale

3 Couplage vision omnidirectionnelle et télémétrie laser

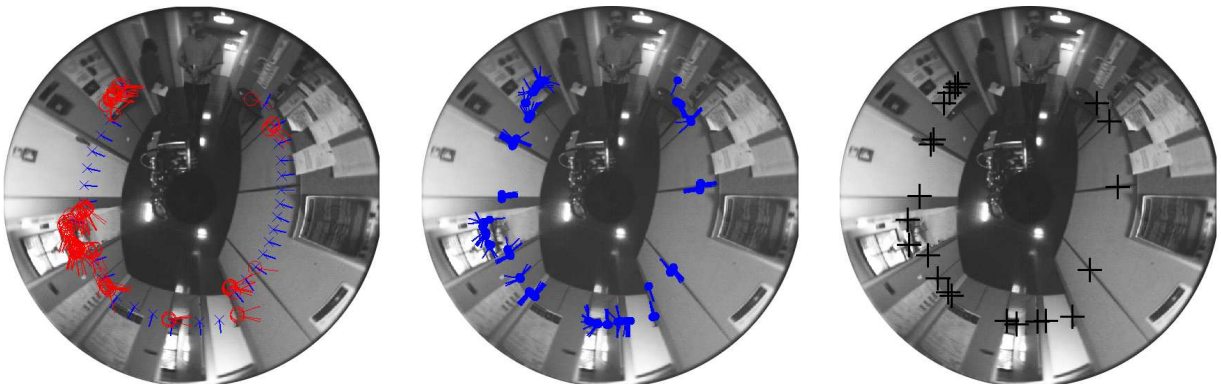
La dernière partie de cette thèse a été consacrée au couplage entre une caméra omnidirectionnelle et un télémètre laser. La vision permet de faciliter la reconnaissance des lieux et l'estimation du mouvement et le laser permet d'obtenir des mesures directes de la distance métrique de points de la scène.

Dans une première partie, nous avons exploré le SLAM avec trois degrés de liberté (3-DOF SLAM soit pour un mouvement planaire du robot). Nous avons ensuite proposé une direction possible pour effectuer la cartographie lors d'un mouvement avec six degrés de liberté (6-DOF SLAM).

3.1 Couplage vision omnidirectionnelle et laser pour le 3-DOF SLAM

La méthode proposée pour la cartographie planaire a les caractéristiques suivantes :

- carte métrique : l'environnement est représenté par un ensemble de points facilement identifiables,



Critère sur l'orientation | Critère sur l'amplitude | Points caractéristiques

Figure 25: Critères pour obtenir des points discriminants

- carte topologique : une carte topologique est utilisée pour gérer de manière explicite l'association entre données,
- fermeture de boucle : pour corriger la dérive, une approche de reconnaissance de lieux est proposée,
- l'état (pose+carte) est mise-à-jour grâce à un filtre de Kalman étendu.

Une approche combinant contours dans l'image et trace laser a été proposée (Section 9.3) pour obtenir des points saillants qui constituent la carte de l'environnement. Figure 25 illustre les différents critères.

En plus d'une carte métrique, nous réalisons une carte topologique dont les sommets constituent un ensemble de lieux identifiables grâce aux images omnidirectionnelles. La carte topologique permet de gérer l'association entre les données. Une distance prise sur la carte topologique permet de décider si nous acceptons l'association de données obtenue par l'incertitude du filtre. En effet, le filtre peut devenir inconsistant (erreurs de linéarisation, mauvaises associations de données, ...) et la distance sur la carte topologique permet de limiter le risque d'association incorrecte.

3.1.1 Fermeture de boucle

La méthode de fermeture de boucle décrite au Chapitre 9 consiste en quatre étapes distinctes :

1. déclenchement d'une vérification de fermeture de boucle grâce à des signatures de lieux (corrélogrammes),
2. estimation de la rotation entre les vues grâce aux images omnidirectionnelles,
3. estimation de la transformation en recalant les coupes laser,
4. association des données et mise à jour du filtre.

Les images omnidirectionnelles sont bien adaptées à la reconnaissance de lieux (invariance en rotation pour des mouvements planaires). Les signatures images utilisées reposent sur les auto-corrélogrammes qui ont des propriétés intéressantes en terme de temps de calcul et de discriminance. La construction est illustrée par la figure 26.

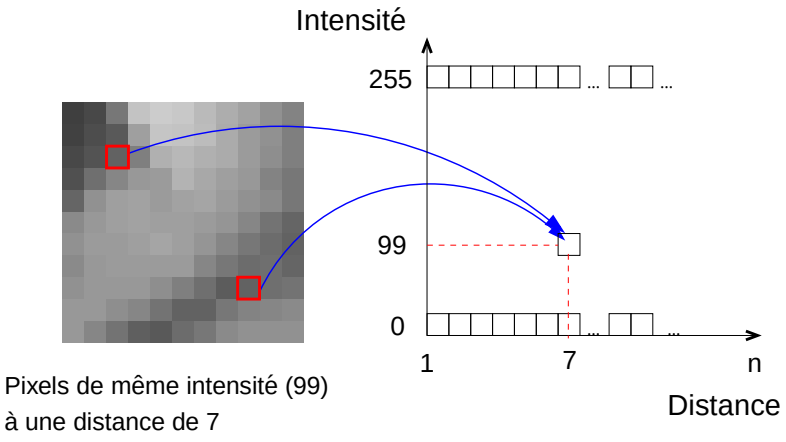


Figure 26: Construction d'auto-corrélogrammes

3.1.2 Résultats expérimentaux

L'approche de cartographie combinant télémétrie laser et vision omnidirectionnelle a été expérimentée sur le robot mobile Anis du projet ICARE. Figure 3.1.2 montre la carte métrique avant la détection d'une fermeture de boucle. Grâce aux signatures images, une fermeture de boucle est déclenchée. L'estimation de la rotation et le recalage des coupes permet d'associer les données et de mettre à jour le filtre. La figure 3.1.2 montre les résultats obtenus après la fermeture de boucle.

3.1.3 Conclusion

L'approche de cartographie 3-DOF proposée a les caractéristiques suivantes :

- aucun *a priori* n'est considéré sur le type d'environnement (en particulier nous ne supposons pas que l'environnement est linéaire par morceaux),
- le capteur de vision permet d'associer les données de manière robuste et de reconnaître des cas de fermeture de boucle,
- une carte topologique permet de gérer l'association de données pour plus de robustesse.

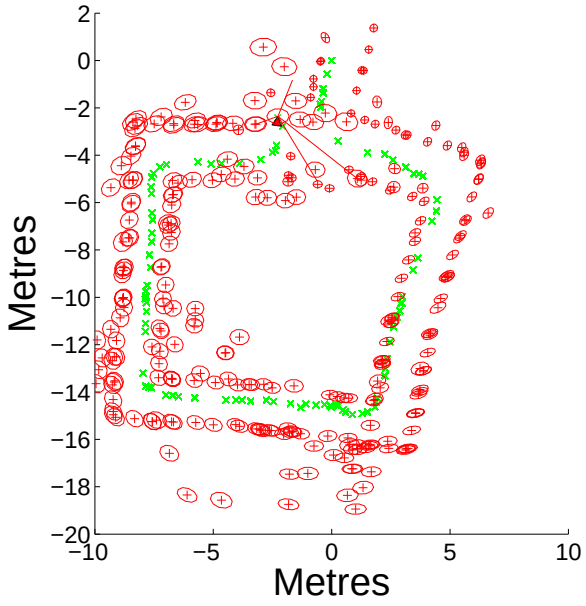
Les limitations de l'approche sont la création d'une carte uniquement planaire et non-dense (ce qui ne permet pas par exemple d'effectuer de l'évitement d'obstacle ou de la planification). Ces limitations nous ont amenés à explorer la cartographie 6-DOF basée vision au Chapitre 10.

3.2 Couplage vision omnidirectionnelle et laser pour le 6-DOF SLAM

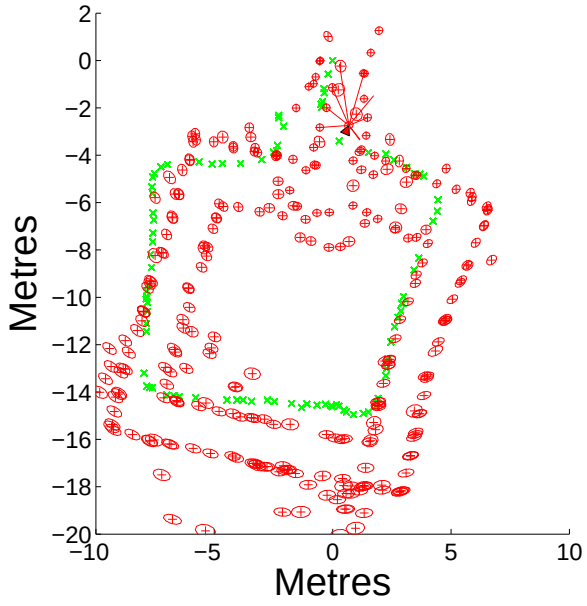
Au Chapitre 10, nous explorons comment combiner données vision et information laser pour obtenir une approche pour la cartographie 6-DOF avec les caractéristiques suivantes :

- 6-DOF SLAM,
- extraction et initialisation automatique de plans grâce au laser 2D,
- rapide, en limitant le nombre de variables à estimer.

La technique repose sur le suivi basé vision développé au Chapitre 6. Le laser permet d'imposer des contraintes sur l'homographie du plan à suivre.



Carte métrique *avant* la fermeture de boucle



Carte métrique *après* la fermeture de boucle

3.2.1 Modélisation du problème

En extrayant des segments de la coupe laser (figure 27), il est possible de contraindre les homographies entre vues comme illustré par les figures 28 et 29. L'homographie peut alors s'écrire :

$$\mathbf{H} \sim \mathbf{R} + \mathbf{t} \left(\mathbf{n}_b + \lambda \mathbf{n}_{Ker}\right)^\top$$

Ainsi le nombre d'inconnus passe de $6 + 3 \times m - 1$ dans le cas de la vision seule à $6 + m$ grâce au segment laser.

Initialisation des plans Plusieurs valeurs peuvent être données à λ lors de l'initialisation comme illustré par les figures 30 et 31.

3.2.2 Résultats expérimentaux

L'estimation du mouvement à partir de plans en combinant télémétrie laser et vision omnidirectionnelle a été validée sur une séquence dans un couloir avec le robot ANIS. La séquence comporte des occlusions et des spécularités. L'utilisation d'estimateurs robustes a permis néanmoins d'obtenir des résultats précis sur cette séquence. Figure 32 montre une des occlusions dans la séquence et figure 33 montre l'estimation du mouvement (lignes avec symboles) comparée à la vérité terrain (en ligne continue).

3.2.3 Conclusion

L'approche proposée permet d'effectuer une cartographie de l'environnement avec 6 degrés de liberté en combinant un laser 2D avec une caméra omnidirectionnelle. Le mouvement et la structure sont estimés de manière efficace grâce au suivi visuel. L'information laser apporte des avantages pour l'initialisation de la structure à suivre. Les limitations de la méthode concernent principalement le type de structure suivi (planaire).

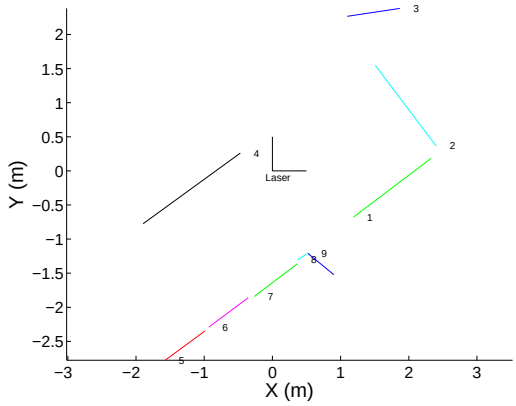


Figure 27: Segments extraits d'une coupe laser

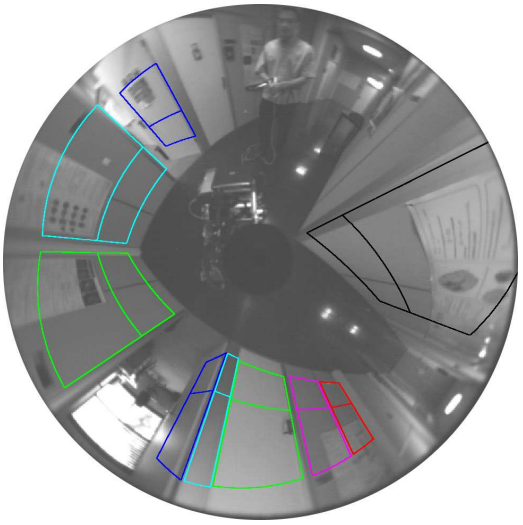


Figure 28: Plans choisis à partir des segments

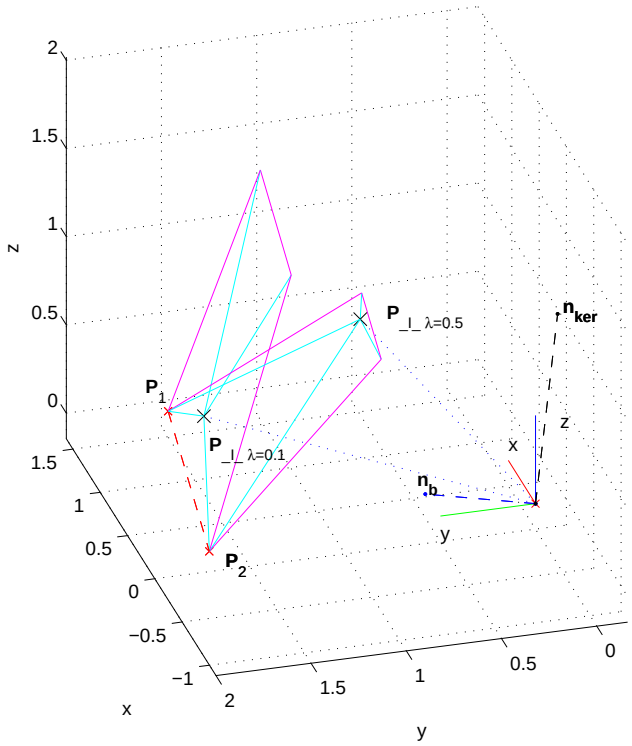


Figure 29: Segment laser : contraint 2 degrés de liberté de l'homographie planaire

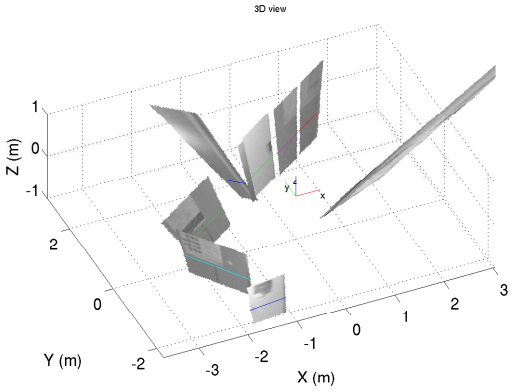


Figure 30: Initialisation pour maximiser l'angle de vue

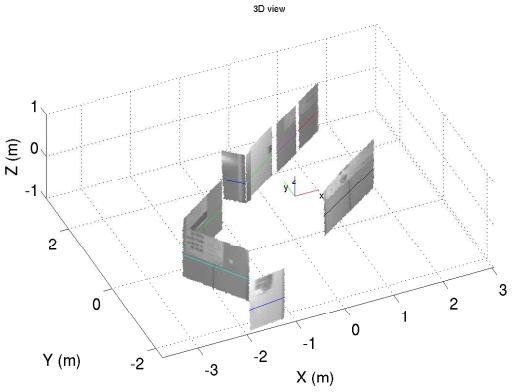


Figure 31: Initialisation sous hypothèse de verticalité

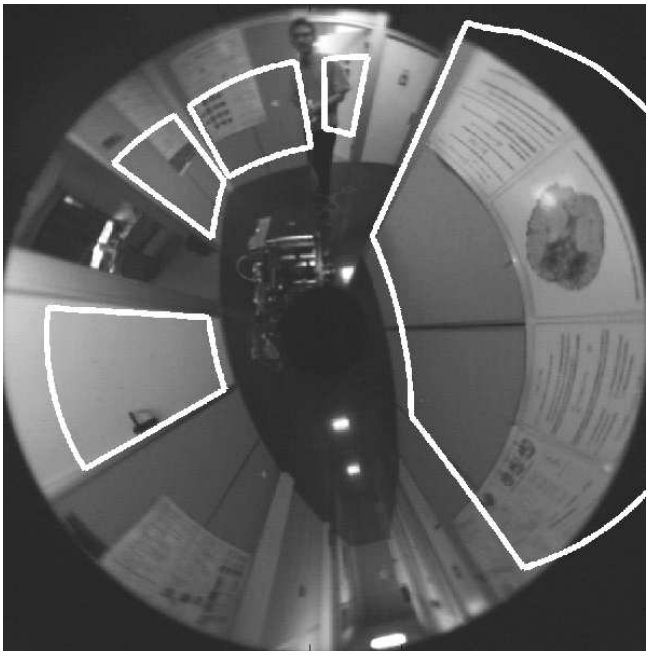
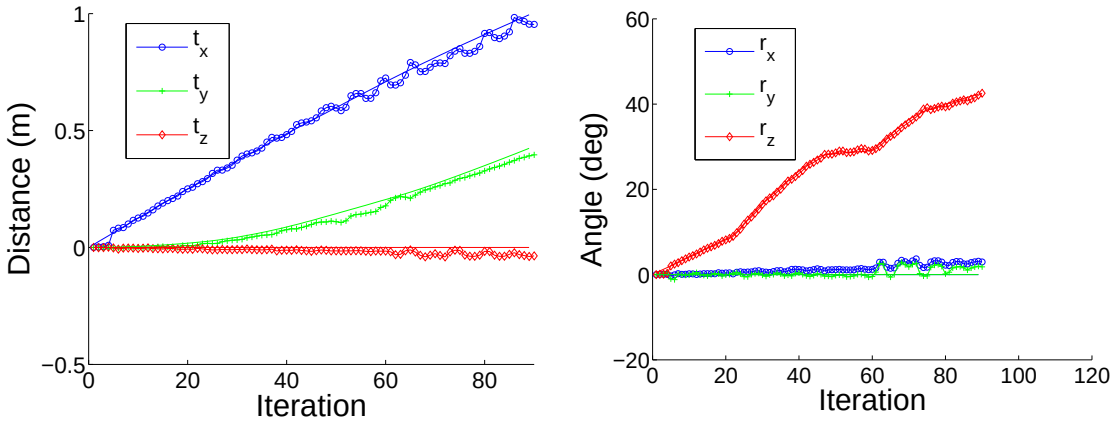


Figure 32: Occlusion (image 90)



Estimation de la translation Estimation de la rotation

Figure 33: Estimation du mouvement

Conclusion et perspectives

Ce travail de thèse a contribuer à faciliter l'utilisation des caméras catadioptriques grâce au développement d'une technique d'étalonnage simple à mettre en œuvre avec un logiciel open-source. Les techniques de suivi et d'estimation du mouvement développées pour les droites et les plans dans les images omnidirectionnelles sont des outils efficaces à sa disposition pour les problèmes de localisation et de cartographie. Les exemples de cartographie ont aussi permis de mettre en avant l'avantage des capteurs omnidirectionnels notamment pour la reconnaissance de lieux et la fermeture de boucle, deux problèmes importants du SLAM.

Ces travaux ouvrent des perspectives dans le domaine de la reconstruction 3D temps-réel soit avec la vision seule soit en combinant vision et laser. La précision et la robustesse des méthodes proposées permettent d'envisager la cartographie dans des environnements complexes et de grandes tailles.

Introduction

Objective

Roboticians and researchers in computer vision have a common goal: to estimate the motion of a robot or camera and simultaneously build a representation of the environment. The first call it simultaneous localisation and mapping (SLAM) and the latter structure from motion (SFM). This problem is considered as essential to build fully autonomous systems that do not require any prior knowledge of the environment to fulfill their tasks. Examples of applications range from the exploration of Mars, to helping firemen locate victims in a building, to guarding a warehouse or aiding soldiers in guerrilla warfare. In all these situations, the environment can be complex and the robot must adapt to fulfill its exploration or surveillance tasks.

Ways of solving the SLAM problem vary between communities. The computer vision literature focuses more on uncalibrated bundle adjustment whereas the robotics community generally favours iterative techniques such as the Kalman filter or particle filtering and often combines calibrated cameras with other sensors (odometry, gyroscopes, laser range finders, ...).

The work in this thesis was done from a robotics perspective. We wish to find a combination of sensors that will help solve some of the challenges of SLAM in large-scale complex environments. The evolution of SLAM is closely linked to the sensors used. Sonars with odometry are often considered as the first sensors having led to convincing results. Since then, 2D laser range finders have often replaced sonars when possible because of the higher precision and better signal to noise ratio. However 2D lasers alone limit SLAM to planar motion estimation and do not provide sufficiently rich information to reliably identify previously explored regions.

Well aware of the limitations of 2D lasers, researchers are increasingly using 3D laser scanners and estimate full 3D trajectories. These sensors however only mildly improve the problem of place recognition and the data acquisition process is not currently compatible with real-time. An obvious alternative is to use vision sensors. Vision sensors enable 6 degree of freedom motion estimation and are rich in perceptual information. However standard cameras only have a small field of view ($\sim$30° -40°). Place recognition or motion estimation can be easily affected by occlusion. Omnidirectional vision, in other words, how to generate and work on large field of view cameras can solve some of these issues. Vision alone does not provide range-bearing measurements and, as such, lasers are better adapted for SLAM.

Combining omnidirectional vision and 2D laser range finders provides many advantages for SLAM that we will explore throughout this thesis.

Contributions

Before this work was undertaken, there were no simple ways to calibrate omnidirectional sensors. We devised a calibration approach and identified the essential parameters that should be taken into account. This lead to two publications [Mei and Rives, 2006a, 2007] and an opensource toolbox made available on the author's website.

Another essential step before fusing the laser data and the vision data, was to establish the relative pose between the sensors. In [Mei and Rives, 2006b], we studied different approaches and worked on the two cases of visible and invisible laser beams.

An effort has also been made to clarify the use of Lie algebras as incremental local parameterisations for minimisation. This technique is essential to the efficient second order approach (ESM) and was used extensively for central catadioptric homography-based tracking [Mei et al., 2006a,b] and structure and motion from line images [Mei and Malis, 2006].

The final part of this thesis relies heavily on the previous chapters to build fast and efficient approaches to motion estimation and map building by fusing the information from the laser range finder and omnidirectional camera.

Outline

Part I

In the first part of this thesis, we will describe the sensors used and how to calibrate them. This is an essential step before using the sensors.

Chapter 1

We will explore the world of omnidirectional vision through biology and human history. Closer to computer vision and robotics, we will describe different methods to acquire large field of views and explain the advantages of central catadioptric sensors for robotics.

Chapter 2

Calibrating a vision sensor consists in finding a mathematical function that links 3D points to their 2D image projections. This function should be sufficiently general to encompass a wide variety of sensors but also sufficiently constrained to avoid sensitivity to noise. Our approach consists in assuming small errors with respect to a theoretical projection model.

Chapter 3

To calibrate, we also need a *method* to estimate the parameters of the function. We will describe the different steps in the calibration process. Results on a wide variety of sensors will be given.

Chapter 4

The problem of finding the relative pose between an omnidirectional sensor and a laser range is studied. We analyse the two distinct cases where the laser beam is visible in the image and when it is invisible.

Part II

The second part of this thesis presents computer vision algorithms adapted to omnidirectional cameras in view of their integration into a SLAM framework.

Chapter 5

Minimal parameterisations are important to ensure robustness and optimality. This chapter describes how Lie algebras associated to Lie groups can impose the required constraints on the parameters. We also detail explicit formulas for the exponential map for an efficient implementation.

Chapter 6

Visual tracking often forms the basis of structure and motion algorithms. We detail how to extend SSD tracking to omnidirectional cameras (and more generally to all single viewpoint sensors) and improve the computational efficiency over standard methods.

Chapter 7

Lines are often used in structure and motion algorithms. We describe how they can be extracted and tracked in omnidirectional images. We also analyse the structure and motion problem using Lie algebras.

Part III

The final part of this thesis describes how to combine the laser and vision information.

Chapter 8

We give here a short overview of SLAM algorithms and the some of the current challenges.

Chapter 9

We describe ways of combining the visual information of an omnidirectional sensor with the metric information from the laser for 3-DOF SLAM.

Chapter 10

6-DOF SLAM has mainly been studied using only computer vision or with 3D lasers. However we will see in this chapter that there are many advantages in combining the precise 2D metric information from the laser with the natural 6-DOF estimation obtained from the vision sensor.

Notations and acronyms

General

$\mathbf{M}$	:	matrix $\mathbf{M}$
$\mathbf{v}$	:	vector $\mathbf{v}$
$\mathbf{R}$	:	rotation matrix
$\mathbf{t}$	:	translation vector
$\mathbf{T}$	:	Euclidean transformation
$\mathbf{M}^\top$	:	the transpose of the matrix $\mathbf{M}$
$\|\mathbf{x}\|$	:	L_2 norm of $\mathbf{x}$
$\mathcal{F}$	:	a reference frame
$\mathfrak{g}$	:	Lie algebra of the Lie group G
$\mathbf{x} \times \mathbf{y}$	:	cross product between $\mathbf{x}$ and $\mathbf{y}$
$[\mathbf{x}]_\times$	:	skew-symmetric matrix associated to $\mathbf{x}$, $[\mathbf{x}]_\times \mathbf{y} = \mathbf{x} \times \mathbf{y}$
$\mathbf{M}^+$	:	pseudo-inverse of $\mathbf{M}$, $\mathbf{M}^+ = (\mathbf{M}^\top \mathbf{M})^{-1} \mathbf{M}^\top$
$\widehat{\mathbf{x}}$	:	estimate of the true value $\mathbf{x}$
$\mathbf{0}_{m \times n}$	:	a matrix with m lines and n columns with zero values
$\mathbf{I}_n$	:	the identity matrix of size $n \times n$

Projective geometry

$\boldsymbol{\mathcal{X}} = (X, Y, Z)$	:	a 3D point
$\boldsymbol{\mathcal{X}}_s = (X_s, Y_s, Z_s)$	:	a point belonging to the unit sphere ($\|\boldsymbol{\mathcal{X}}_s\|$=1)
$\mathbf{p} = (u, v)$	:	coordinate of a point in the image
$\mathbf{m} = (x, y)$	:	coordinate of a point on the normalised plane
Π	:	function that projects a 3D point to the image plane
$\mathbf{K}$	:	camera projection matrix
$\hbar$	:	function projecting a 3D point to the normalised plane for an omnidirectional sensor
$\mathcal{L}$	:	2D or 3D line
$\mathcal{I}$	:	an image

Simultaneous Localisation and Mapping

See Section 8.2.1 for specific notations.

$\mathcal{S}$	:	a laser scan

Acronyms

SLAM	:	Simultaneous Localisation and Mapping
CML	:	Concurrent Map-building and Localisation
ELS	:	Enriched Laser Scan
NNG	:	Nearest Neighbour Gating
ICP	:	Iterative Closest Point
MRP	:	Matching-Range-Point rule
DOF	:	Degree Of Freedom

Contents

Part I

Omnidirectional vision : projection models and calibration

Chapter 1

An introduction to omnidirectional vision

Contents

1.1 An insight on different aspects of large field of views

1.1.1 Importance of wide field of views in nature

Biomimetics is the study of nature as a way to develop or improve design and engineering of machines. Panoramic vision often appears in the biorobotics literature and we give here a short insight into this work.

When we wish to build a fast and robust robot for localisation and mapping, we start by asking the question: are there any existing systems with such capabilities? Nature offers a wide range of examples in particular in the insect kingdom. Despite having relatively simple nervous systems and restricted processing capabilities, insects show effective solutions for autonomous navigation. Many animals are known to find their way back home reliably after foraging for food [Graham and Collett, 2002].

The Saharan desert ant, for example, can forage up to 200 m away from its nest and return in a straight line. It does not use pheromones, as most ants do, as they would evaporate quickly because of the high ground temperature. Instead it uses a combination of compass information (absolute orientation) from polarised patterns of the sun and visual landmarks (relative orientation). By using panoramic vision, roboticians were able to mimic simple navigation strategies [Weber et al., 1998; Lambrinos et al., 1999]. The wide-angle field of view gave more discriminate results and more robustness to changes in the environment.

Similarly research on the human brain, shows that regions (called "place cells") in the hippocampus are dedicated to visual navigation [Giovannangeli et al., 2006]. An implementation of the algorithms on a mobile robot underlying the image processing gave satisfying results for navigation.

If we look at larger scale animals, we see that the field of view is also adapted to the type of animals and the environment. Herbivores (rabbits, horses, cows, ...) have a large field of view (more than 300°) but a small binocular field (around 50°). Carnivores or primates on the contrary have smaller field of views with bigger binocular regions.

For migratory birds, a large field of view is essential for localisation through landmarks and for following the horizon [bir, 2005]. Recently, robust attitude estimation for aerial robot navigation has been proposed [Demonceaux et al., 2006].

1.1.2 Panoramas in history

Panoramas started as an art form. It was a way of immersing the viewer in the world created or reproduced by the artist. The first patent on the subject was filed by Robert Barker in 1767. The first detailed book on the subject was written in 1794 and gives an insight on the works produced at the time. In the 1820s, viewing realistic 360° paintings became very popular in Paris. The techniques evolved with better acquisition methods and enhancement of the viewer's experience through added visual motion effects on the 2D surface. The paintings were usually of historical battles or important political events. In Britain, for example, viewers would have seen "The Battle of Waterloo" (1815) or "The Coronation of George IV" (1822). There were also depictions of landscapes and popular places such as "The Panorama of Toulon" in Germany or "Palace and Gardens of Versailles" in the United States.

Improvements in photography by the end of 19^{th} made it possible to assemble photographs and create photographic panoramas. Cineoramas by Raoul Grimoin-Sanson followed at the Universal Exposition held in Paris in 1900. In fact it is unclear today if there ever was any public projections because of the fire hazard created by the movie equipment at the time. However it did start a popular

entertainment industry. Many amusement parks today have panoramic cinemas. The term "O-rama" (or "A-rama") itself came to signify any expensive entertainment spectacular and even trendy products.

More references can be found in Stephan Oettermann's book "The Panorama: History of a Mass Medium" or in the historical perspective of "Panoramic Vision" [Benosman and Kang, 2001].

1.1.3 Catadioptric sensors in history

Panoramic cameras appeared in the mid-19$^{\text{th}}$ century. They were composed of lenses rotating around a given axis (*swing lens cameras*). Because the camera was stationary, the acquired field of view was limited between 120° and 150° . *Rotating cameras*, created shortly after, do not have this limitation and make it possible to create 360° views of the environment. An alternative to capture a large field of view is to combine a camera with a convex mirror. The image then needs to be processed with a computer. Sensors with convex mirrors are called catadioptric cameras, from *dioptric*, the science of light refraction (lenses) and *catoptric*, the science of reflective surfaces (mirrors).

Rees [Rees, 1970] was the first in 1970 to patent the combination of a perspective camera and a convex mirror (in this case a hyperbolic mirror). In his US patent, he describes how to capture omnidirectional images that can be transformed to correct perspective views (no parallax).

It was only much later, in the 90's, that omnidirectional vision became an active research topic in computer and robot vision.

Different sensors for capturing wide field of views

There are several possible methods for obtaining a wide angle field of view. The choice should be made according to the task we wish to solve. A state of the art is presented in [Yagi, 1999]. The techniques can be classified in three categories:

- reconstruction from several images (mosaicing),
- use of wide angle lenses (fish-eye),
- use of convex mirrors.

A desirable property of these systems is the single viewpoint constraint. This indicates that images are produced without parallax and that we can recreate perspective images without distortion. Under this constraint, the results of projective geometry can be used as such.

Mosaicing Reconstruction of panoramas from several images can be obtain from multiple cameras or a camera rotating around a given axis. With these systems, acquisition and the data association is computationally expensive and rarely real-time. It is also difficult to obtain images without parallax. Systems with several cameras are generally not very compact but do have the advantage of high resolution.

Wide-angle lenses Wide angle lenses are an obvious way of obtaining compact real-time wide field of view images. However they do not verify exactly the single viewpoint constraint and the field of view is often less than 180° .

Catadioptric cameras By placing a perspective camera in front of a convex mirror (or several mirrors for *folded* sensors), we can obtain a 360° view of the environment. The acquisition is real-time with a good resolution around the mirror border (Figure 1.1). Under certain conditions on the mirror shape, the single viewpoint constraint can be fulfilled. The main disadvantage of these systems is the non-uniform resolution and a low resolution compared to the use of multiple cameras.
These sensors are becoming popular choices for applications in robotics. Experiments in this thesis will mainly concern catadioptric cameras even though the results are often more general.

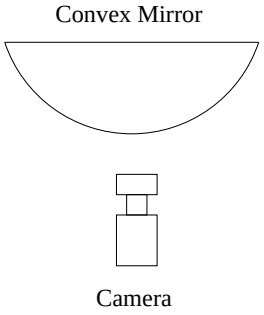


Figure 1.1: A Catadioptric camera

1.2 Omnidirectional vision in robotics and computer vision

The interest for omnidirectional vision in the robotics community is very pragmatic: it is easier to recognise previously observed places whatever the orientation with a 360° field of view, it is also less likely that the robot will "get stuck" when facing a wall or obstacle.

The sensors used at first did not satisfy the single viewpoint constraint and were based on a combination of a perspective camera with a conic [Yagi and Kawato, 1990] or a spherical [Hong et al., 1991] mirror. Hyperbolic sensors with their single viewpoint property were rediscovered in [Yamazawa et al., 1993]. The work by Yagi *et al* with their conic sensor COPIS explored the problems of motion estimation and obstacle avoidance. Several other projects like the SYCLOP project from the University of Picardie Jules Verne in France also explored the optical properties of the sensor and its application to mobile robotics.

In [Baker and Nayar, 1998], an answer was given to the question "What are all the combinations of cameras with a single mirror that satisfy the single viewpoint constraint?". This helped clarify how to avoid parallax in catadioptric design. (We may note however that folded catadioptric cameras combining several mirror that are more compact and cheaper to manufacture can also follow the single viewpoint constraint [Nayar and Peri, 2001].) This is often considered as a turning point in omnidirectional vision. Since then a lot of research in computer vision has been dedicated to the projective properties of omnidirectional sensors [Svoboda et al., 1998] and the problem of calibration. In robotics, the initial hypothesis of planar motion is being slowly abandoned for full 6-DOF motion estimation algorithms. The CAVIAR project (Catadioptric Vision for Aerial Robots) for example is dedicated to estimating the motion of aerial robots.

There has recently been a renewed interest for non single viewpoint sensors that are often cheaper to manufacture and present interesting theoretical challenges [Pajdla, 2002].

1.3 Conclusion

We have seen that in the animal kingdom, panoramic vision plays a key role for localisation but also for survival by helping herbivores anticipate possible attacks from predators. For humans, panoramas appeared at first as an art form and in the entertainment industry by immersing the viewer in past battles and exotic places. Nowadays panoramic vision is becoming used increasingly to improve robots achieve their tasks. Within the choice of wide field of view imaging devices, convex mirrors are well adapted for robotic tasks with a large field of view acquired in real-time and - under certain conditions - a single viewpoint.

Chapter 2

Projection Models

Contents

In this chapter, we will describe how to link the image we see in our sensor to the light rays emitted by a region of space. This is the basic step to using vision sensors but it is not a straight forward task: sensors are not perfect and we need to model small errors of design. We will also see that wide-angle sensors impose a different view of the world through spherical perspective projection.

2.1 Definitions and notations

Let I be an image of the world obtained through an optical device. We will consider I to be a two-dimensional finite array containing intensity values (irradiance). I can be seen as a function:

$$\begin{array}{rccc} I & : & \Omega \subset \mathbb{R}^2 & \longrightarrow & \mathbb{R}_+ \\ & & (u,v) & \longmapsto & I(u,v) \end{array}$$

The irradiance at an image point $\mathbf{p} = (u, v)$ is due to the energy emitted from a region of space determined by the optical properties of the device. In the case of a *central* device with a unique viewpoint, the direction of the energy source is represented by a projective ray (a half-line) with initial point the optical center (or focus) of the device noted C. If we cannot consider a single viewpoint C, the device will be said to be *non-central*. Two characteristics are then of interest:

1. the viewpoint is a continuous region called a caustic [Swaminathan et al., 2006],
2. there are several viewpoints and the system is a *general imaging device* (GID). A stereo head for example is a GID if considered as a single camera.

A sensor can of course have a combination of the these properties.

Caustics are sometimes seen as defaults in a perspective imaging device. However, they can also be considered as part of the sensor's properties and have predefined shapes. For example, the viewpoint for a linear pushbroom camera will be a line and for an omnivergent camera it will be a sphere or circle. Examples of such devices can be found in [Bakstein and Padjla, 2001; Sturm, 2005].

In this thesis, we will consider only central catadioptric devices or cameras than can be approximated by the projection model proposed in the following section (fisheye lenses, spheres).

We will now present two perspective projection models: the planar perspective projection that is the standard approach to define perspective cameras and the spherical perspective projection that is better adapted to wide angle views. The aim of the perspective projection is to separate what is common to most visual imaging systems (projective geometry) from what is specific to a given device (intrinsic parameters). A perspective projection removes the depth information of a 3D point. The goal of structure from motion (SFM) is to recover this information.

2.1.1 Planar perspective projection

The standard projection model of a 3D point onto the image plane uses the *normalised* image plane. The steps are (figure 2.1):

1. let $(\boldsymbol{\mathcal{X}})_{\mathcal{F}_c} = (X, Y, Z)$ be a 3D point in the camera reference frame, it is projected to the normalised plane π_m by the following equation:

$$(\boldsymbol{\mathcal{X}})_{\mathcal{F}_c} \longrightarrow \mathbf{m} = (x, y, 1) = (\frac{X}{Z}, \frac{Y}{Z}, 1)$$

2. with f_1 the horizontal focal length, f_2 the vertical focal length, s the skew factor and (u_0, v_0) the principal point, the projection of $\mathbf{m}$ in homogeneous coordinates to the image plane π_p is obtained linearly by:

$$\mathbf{p} = (u, v, 1) = \mathbf{K}\mathbf{m} = \begin{bmatrix} f_1 & f_1 s & u_0 \\ 0 & f_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{m} = k(\mathbf{m})$$

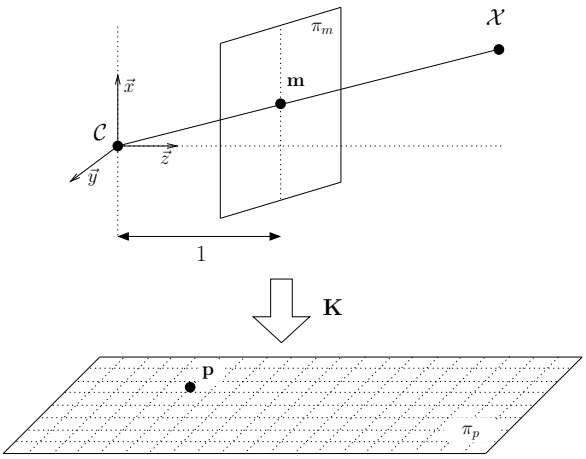


Figure 2.1: Planar perspective projection

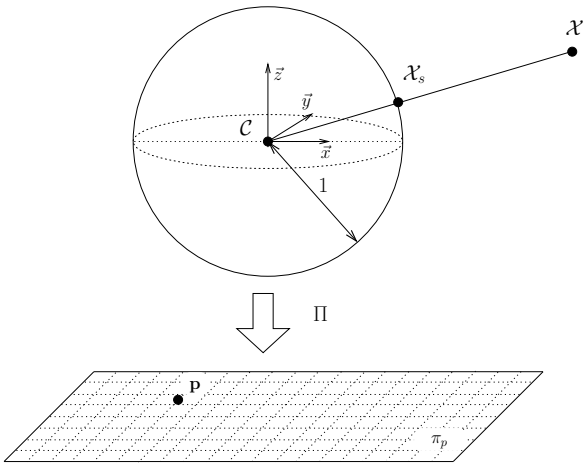


Figure 2.2: Spherical perspective projection

For a field of view greater than 180° , this model is not adapted. With a unique plane, there is an ambiguity between the front and the back of the camera which makes metric reconstruction impossible.

2.1.2 Spherical perspective projection

Instead of projecting the points to the unit plane π_m, we can project the points to the unit sphere $\mathbb{S}^2 = \{\boldsymbol{\mathcal{X}} \in \mathbb{R}^3 | \|\boldsymbol{\mathcal{X}}\| = 1\}$. We will note $\boldsymbol{\mathcal{X}}_s$ the points on $\mathbb{S}^2$. The cheirality constraint[1] can be imposed even for a field of view greater than 180° . The scale factor λ relating points on the sphere to the 3D points must be positive:

$$\exists \boldsymbol{\mathcal{X}}_s \in \mathbb{S}^2 \implies \exists \lambda > 0 | \boldsymbol{\mathcal{X}} = \lambda \boldsymbol{\mathcal{X}}_s$$

From the unit sphere, we can then apply the projection function noted Π that depends on the intrinsic parameters of the sensor:

$$\Pi : \Upsilon \subsetneq \mathbb{S}^2 \to \Omega \subset \mathbb{R}^2$$

Π is not defined on *all* of $\mathbb{S}^2$ because we wish Π to be bijective which cannot be the case between $\mathbb{S}^2$ and $\mathbb{R}^2$ as they do not share the same topology. If Π is bijective, Π^{-1} will relate points from the image plane to their projective rays (lifting).

[1] constraint that the scene points are in front of the camera

2.2 Central catadioptric projection model

As explained in the previous chapter, a single viewpoint is a desirable property as it enables the creation of perspective images without parallax. Baker and Nayar [Baker and Nayar, 1998] derived the class of central catadioptric cameras with this property under the assumption of the pinhole camera model. The four configurations that have this property are an orthographic camera associated to a parabolic mirror or a perspective camera associated to a hyperbolic, elliptical or planar mirror. Figure 2.4 adapted from the work of Barreto [Barreto, 2003] depicts these cases. We choose the unconventional axis representation shown in figure 2.3.

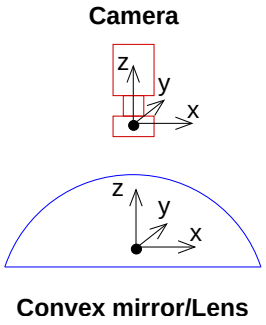


Figure 2.3: Axis

2.2.1 Degenerate configurations

The spherical mirror and the cone are two degenerate configurations. The sphere can be seen as the limit of an ellipse when the two focal points coincide. To obtain a single viewpoint, we would need to place the camera in the center of the sphere. We would then only see the camera itself. By putting the camera in another position, we obtain a caustic. A spherical mirror has the advantage of being less expensive to manufacture than central catadioptric systems and easier to calibrate: any diameter can be chosen as an optical axis for the sensor.

The cone is an interesting example of the limit of the pinhole camera model. The single viewpoint constraint imposes that the cone be situated in front of the camera with the vertex at the focal point. For a pinhole camera, this would mean no light could be seen by the imager. However under the Gaussian optics model, light is visible and we keep the single viewpoint property [Lin and Bajcsy, 2006]. Cones are cheaper to manufacture than hyperbolic or parabolic surfaces. This sensor could be use to enrich 2D laser range scans with visual data. The narrow horizontal field of view makes it however less attractive for structure from motion with 6 degrees of freedom (DOF).

Geyer [Geyer, 2003] and Barreto [Barreto, 2003] developed a unified model to study all central catadioptric cameras. We will present this model in the next section and see that it encompasses a larger range of devices including fisheye lenses.

2.2.2 Unified projection model

In figure 2.4 is presented the entire class of central catadioptric sensors. Table 2.1 details the equations of the surfaces and the relation between the standard (a, b) parameterisation and the (p, d) parameters used in Barreto's model.

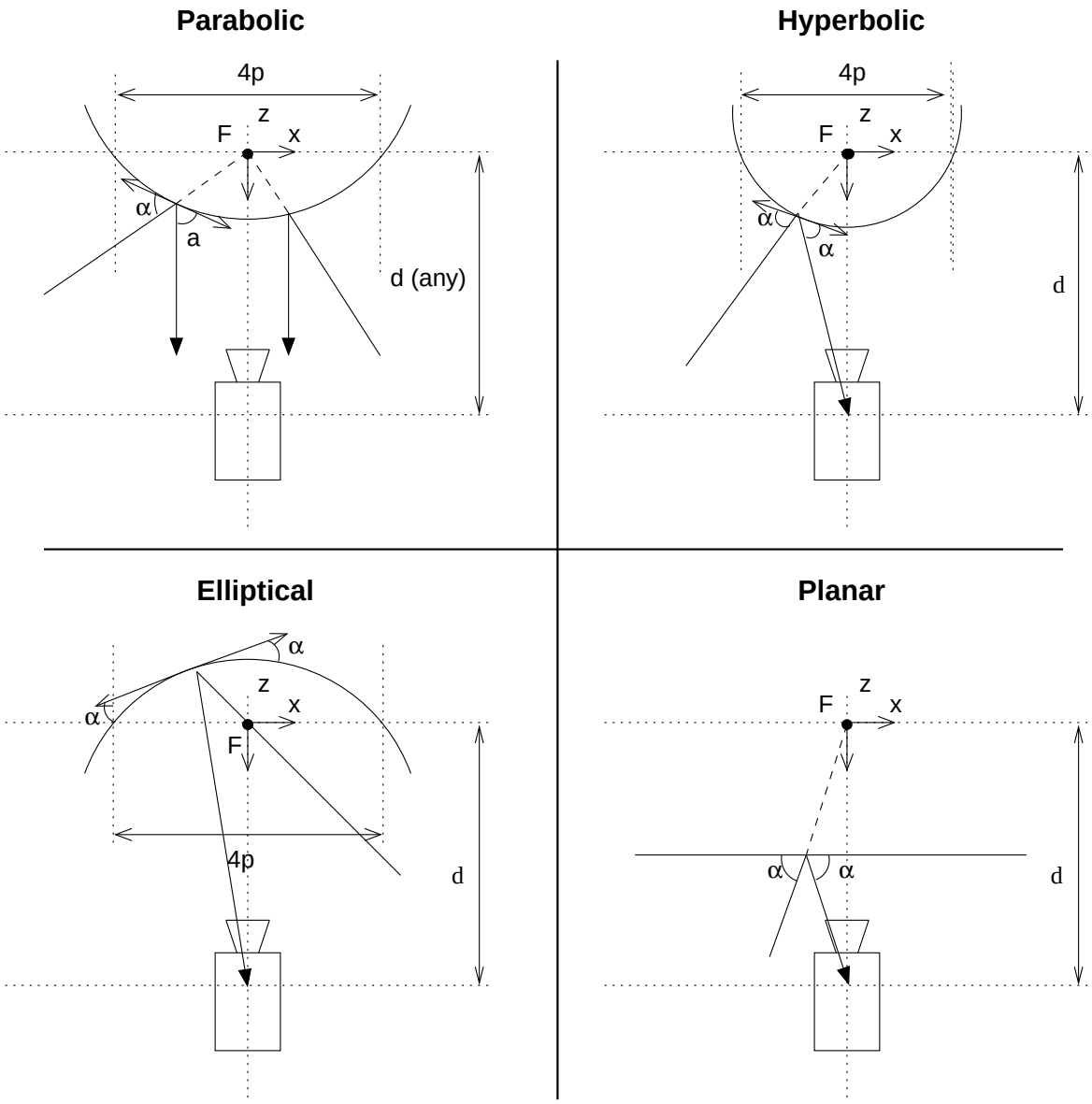


Figure 2.4: The class of catadioptric sensors with a single viewpoint

Table 2.1: Conic equations

Parabola	$\sqrt{x^2+y^2+z^2} = 2p - z$
Hyperbola	$\frac{(z-\frac{d}{2})^2}{a^2} - \frac{x^2}{b^2} - \frac{y^2}{b^2} = 1$
Ellipse	$\frac{(z-\frac{d}{2})^2}{a^2} + \frac{x^2}{b^2} + \frac{y^2}{b^2} = 1$
Plane	$z = \frac{d}{2}$

$a = 1/2(\sqrt{d^2+4p^2} \pm 2p)$, '$-$' for a hyperbola, '$+$' for an ellipse

$b = \sqrt{p(\sqrt{d^2+4p^2} \pm 2p)}$, '$-$' for a hyperbola, '$+$' for an ellipse

The projection model induced by these mirrors can be unified using the spherical perspective projection. A geometrical proof was proposed by Geyer [Geyer, 2003] and gives an intuitive explanation of the equivalence of the projection on a quadric surface and the projection on the sphere.

2.2.2.1 Geometric explanation

Consider figure 2.5 that represents the projection of a 3D point in the parabolic case.

On the left, we see the standard projection model. The ray FP between the 3D point P and the focal point F intersects the mirror in K_1. It is then reflected parallel to the optical axis and intersects the image plane in Q.

On the right, the same projection can be obtain by using a sphere centered in F and of radius d. The point P is first projected to K_2. K_2 is then projected from the North pole N to the image plane in the same point Q.

In the hyperbolic case, a similar result can be obtained but the center of projection is no longer the North pole N but a point between N and F, the position depending on the shape of the mirror.

By algebraic manipulation, the *unit* sphere can be used as the projective surface (Barreto [Barreto, 2003]) and the spherical perspective projection becomes a natural representation separating extrinsic and intrinsic parameters.

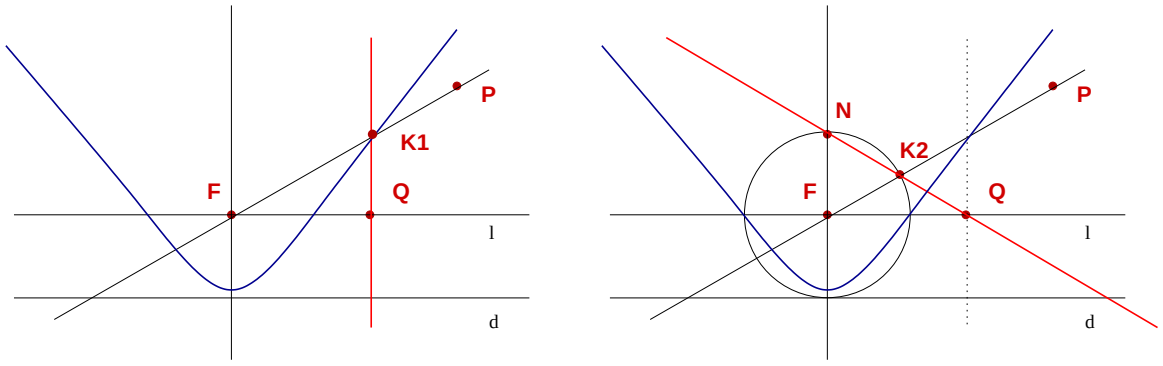


Figure 2.5: Equivalence between the projection on a quadric and the projection on a sphere

2.2.2.2 Projection model

We present here a slightly modified version of the projection model of Geyer and Barreto (Fig. 2.6). We choose the convention that the z axis is the optical axis and points *outwards* (this simplifies the formulas for unifying fisheye and catadioptric sensors). The projection of 3D points can be done in the following steps (the values for ξ and γ, in the ideal case where $\gamma_1 = \gamma_2 = \gamma$, are related to the mirror parameters. The equations can be found in Table 2.2):

1. world points in the mirror frame are projected onto the unit sphere,
$$(\boldsymbol{\mathcal{X}})_{\mathcal{F}_m} \rightarrow (\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_m} = \frac{\boldsymbol{\mathcal{X}}}{\|\boldsymbol{\mathcal{X}}\|} = (X_s, Y_s, Z_s)$$
2. the points are then changed to a new reference frame centered in $\boldsymbol{\mathcal{C}}_p = (0, 0, \xi)$,
$$(\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_m} \rightarrow (\boldsymbol{\mathcal{X}}_s)_{\mathcal{F}_p} = (X_s, Y_s, Z_s + \xi)$$
3. they are then projected onto the normalized image plane,
$$\mathbf{m} = (\frac{X_s}{Z_s + \xi}, \frac{Y_s}{Z_s + \xi}, 1) = \hbar(\boldsymbol{\mathcal{X}}_s)$$
4. the final projection involves a generalized camera projection matrix $\mathbf{K}$ (with (γ_1, γ_2) the generalized focal lengths, (u_0, v_0) the principal point and s the skew)
$$\mathbf{p} = \mathbf{K}\mathbf{m} = \begin{bmatrix} \gamma_1 & \gamma_1 s & u_0 \\ 0 & \gamma_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{m} = k(\mathbf{m})$$

The function $\hbar$ is bijective from $\{\boldsymbol{\mathcal{X}}_s | Z_s > -\xi\}$ to $\mathbb{R}^2$ and:

$$\hbar^{-1}(\mathbf{m}) = \begin{bmatrix} \frac{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} x \\ \frac{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} y \\ \frac{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{bmatrix} \tag{2.1}$$

This last value constrains the points to be on the sphere, but we also have the simpler projective equivalence:

$$\hbar^{-1}(\mathbf{m}) \sim \begin{bmatrix} x \\ y \\ 1 - \xi \frac{x^2+y^2+1}{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}} \end{bmatrix} \tag{2.2}$$

We will call *lifting* the calculation of the point $\boldsymbol{\mathcal{X}}_s$ corresponding to a given point $\mathbf{m}$ (or $\mathbf{p}$ according to the context). We may note that in the perspective case, there is no mirror and only points with $Z > 0$ are considered (we thus fall back to the standard projection model with an extra normalization to the sphere).

The difference with the model from Barreto, is the use of a generalised focal length that depends on the focal length of the camera and on the mirror shape. This is a conceptual change: we consider the sensor to be a single imaging device and not the combination between a camera and a mirror. In the rest of this thesis, we will refer to this model as the unified projection model (**UPM**). This model is interesting for the study of the theoretical properties of central catadioptric sensors but has limitations as we will see in the following section.

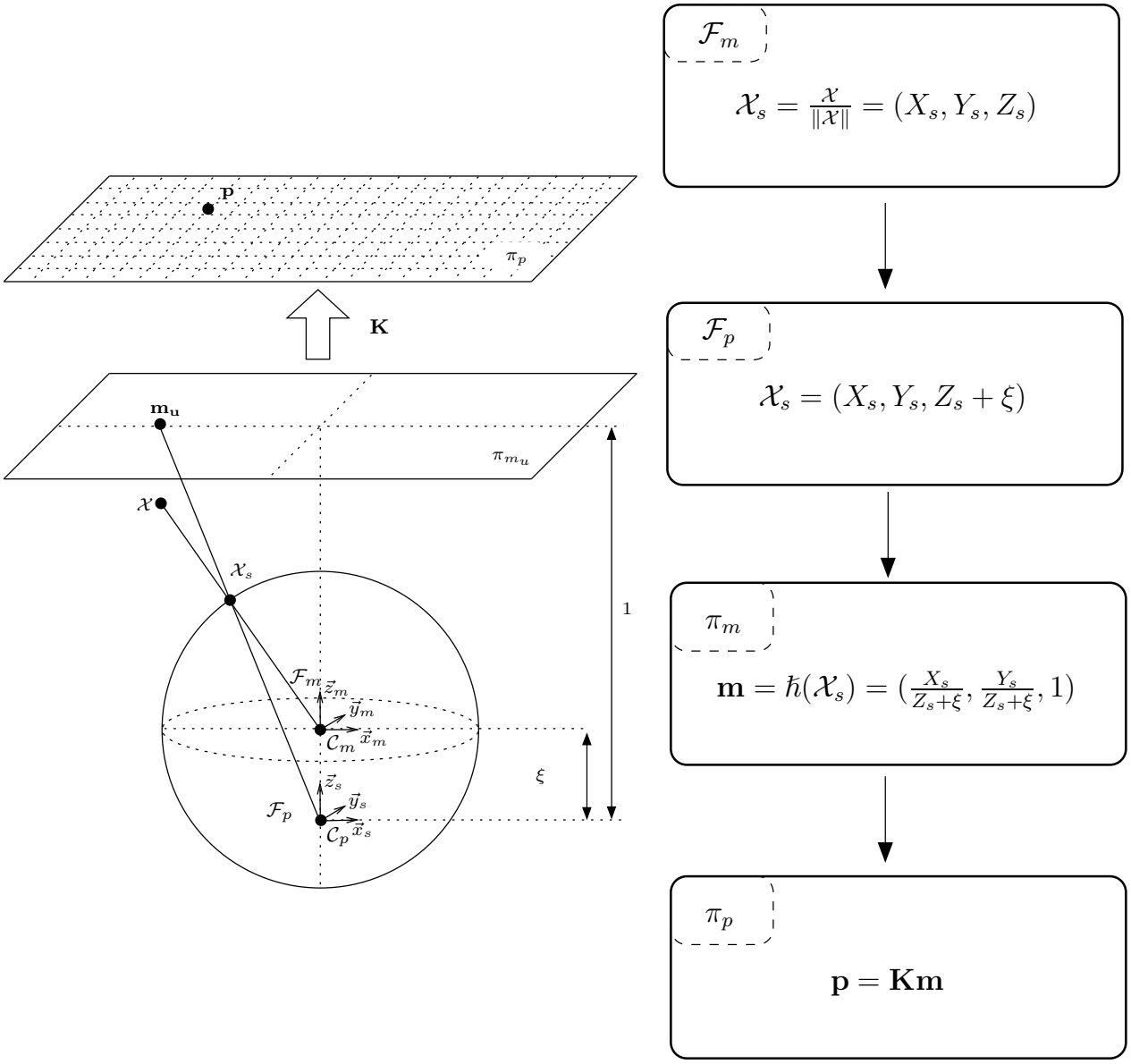


Figure 2.6: Unified projection model

Table 2.2: Unified model parameters

	ξ	γ
Parabola	1	$-2pf$
Hyperbola	$\frac{df}{\sqrt{d^2+4p^2}}$	$\frac{-2pf}{\sqrt{d^2+4p^2}}$
Ellipse	$\frac{df}{\sqrt{d^2+4p^2}}$	$\frac{2pf}{\sqrt{d^2+4p^2}}$
Planar	0	-f
Perspective	0	f
d: distance between focal points $4p$: latus rectum		

2.2.2.3 Fisheye lenses

The unified projection model has been shown to be valid for some fisheye lenses [Ying and Hu, 2004] under the approximation of the division model [Brauer-Burchardt and Voss, 2001; Fitzgibbon, 2001]. This model will be presented here as it has implications in the linear estimation of motion and intrinsic parameters in omnidirectional computer vision.

Division model Let $\mathbf{m}_u = \begin{bmatrix} x_u \\ y_u \end{bmatrix}$ be a point before distortion and $\mathbf{m}_d = \begin{bmatrix} x_d \\ y_d \end{bmatrix}$ after. With $\rho_u = \sqrt{x_u^2 + y_u^2}$ and $\rho_d = \sqrt{x_d^2 + y_d^2}$, the following rational function has been shown to model correctly some fisheye sensors [Brauer-Burchardt and Voss, 2001; Fitzgibbon, 2001] (figure 2.7):

$$\rho_u = k_1 \frac{\rho_d}{1 - k_2 \rho_d^2} \tag{2.3}$$

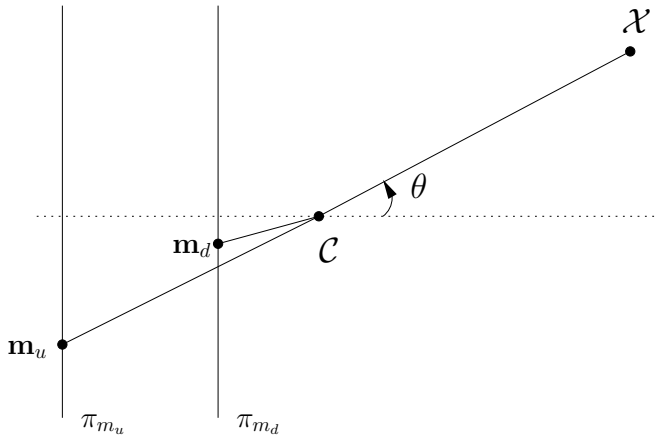


Figure 2.7: Projection model for fisheye sensors

Relationship with the unified projection model In [Ying and Hu, 2004], the authors show that the unified projection model can approximate fisheye projections. The planar perspective projection

to the normalised plane can be written:

$$\mathbf{m}_u = (x, y, 1) = (\frac{X}{Z}, \frac{Y}{Z}, 1)$$

with $\xi = 1$, the unified projection model becomes:

$$\mathbf{m}_d = (\frac{X}{Z + \|\boldsymbol{\mathcal{X}}\|}, \frac{Y}{Z + \|\boldsymbol{\mathcal{X}}\|}, 1) \tag{2.4}$$

By algebraic manipulation, we obtain the relation:

$$\rho_u = \frac{2\rho_d}{1 - \rho_d^2}$$

which has the same form as equation (2.3).

2.3 Compensating for telecentric distortion and misalignment

Equations (2.4) and (2.2) have a particularly simple form for parabolic sensors ($\xi = 1$). This hides a practical problem. To satisfy the property of a unique viewpoint, parabolic sensors need to be combined with orthographic cameras. An alternative is to combine a telecentric lens with a perspective camera. However the lens has to have a diameter of the same size as the mirror (figure 2.9). Large lenses are generally difficult to manufacture and introduce radial distortion in the model. In this section we describe a standard radial and tangential distortion model and ways of approximating the inverse of the function. This function will be applied after the projection of the points on the normalised plane. Figure 2.8 shows the full projection model used throughout this thesis. We will refer to this model as the complete projection model (**CPM**).

2.3.1 Distortion

We will consider two main sources of distortion [Weng et al., 1992]: imperfection of the lens shape that are modeled by radial distortion and improper lens and camera assembly (which can also include misalignment between the camera optical axis and the mirror rotational axis) that generate both radial and tangential errors. In the case of a paracatadioptric sensor, the radial model will compensate for the radial distortion induced by the telecentric lens.

Let $\mathbf{m}_u = \begin{bmatrix} x_u \\ y_u \end{bmatrix}$ be an undistorted point, the point $\mathbf{m}_d = \begin{bmatrix} x_d \\ y_d \end{bmatrix}$ after radial distortion can be obtained from the infinite series, with $\rho_u = \sqrt{x_u^2 + y_u^2}$:

$$\begin{cases} \mathbf{m}_d = \mathbf{m}_u + \mathbf{m}_u R(\rho_u) \\ R(\rho_u) = k_1\rho_u^2 + k_2\rho_u^4 + k_3\rho_u^6 + \cdots \end{cases} \tag{2.5}$$

Generally two parameters are sufficient to calibrate the sensor. We will only consider (k_1, k_2).

We can add tangential distortion to model the misalignment between the mirror axis and the camera optical axis (this is a combination between decentering distortion and thin prism distortion that arises from imperfection in lens design [Weng et al., 1992]):

$$T(\mathbf{m}_u) = \begin{bmatrix} 2p_1x_uy_u + p_2(\rho_u^2 + 2x_u^2) \\ p_1(\rho_u^2 + 2y_u^2) + 2p_2x_uy_u \end{bmatrix}$$

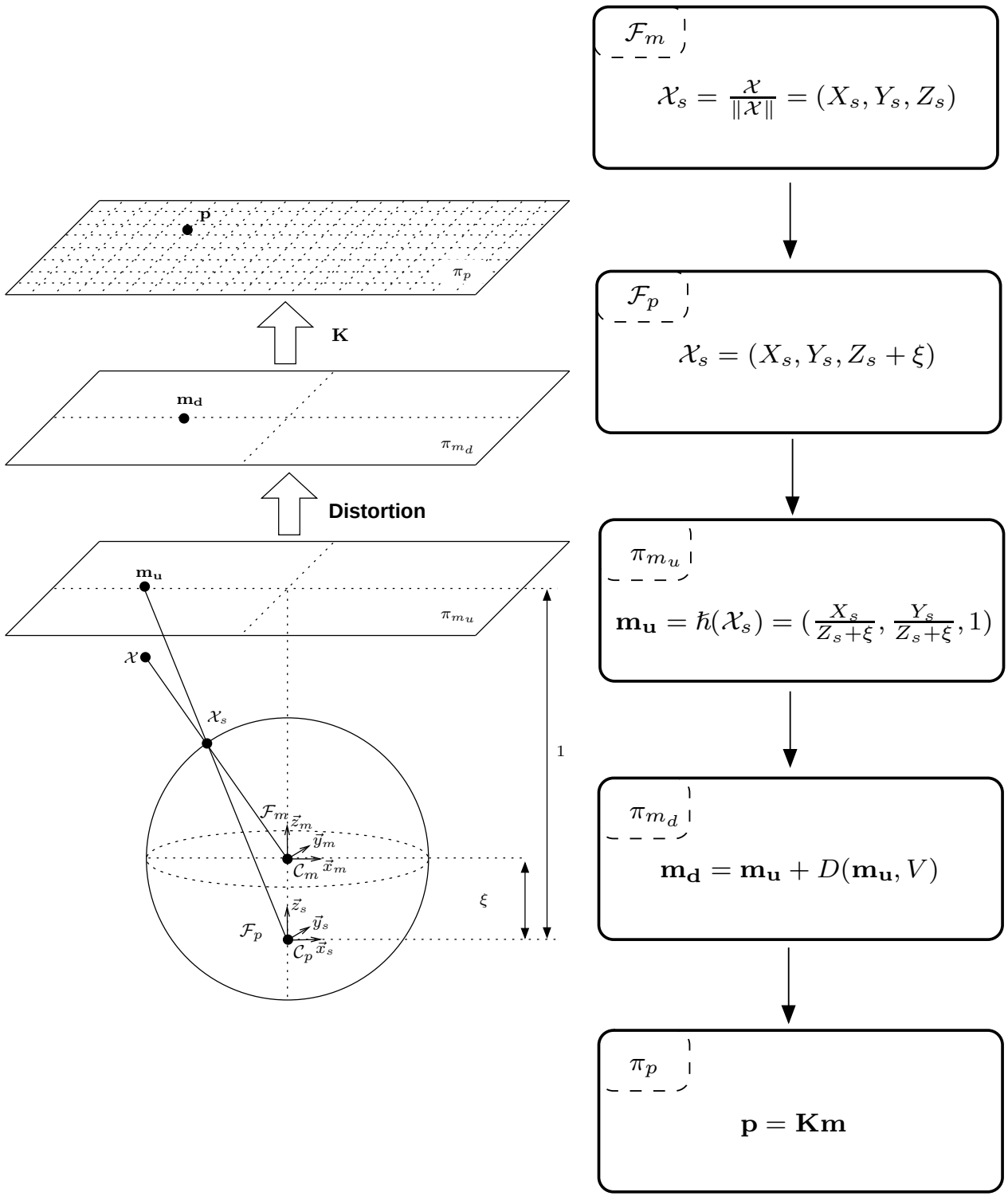


Figure 2.8: Full projection model

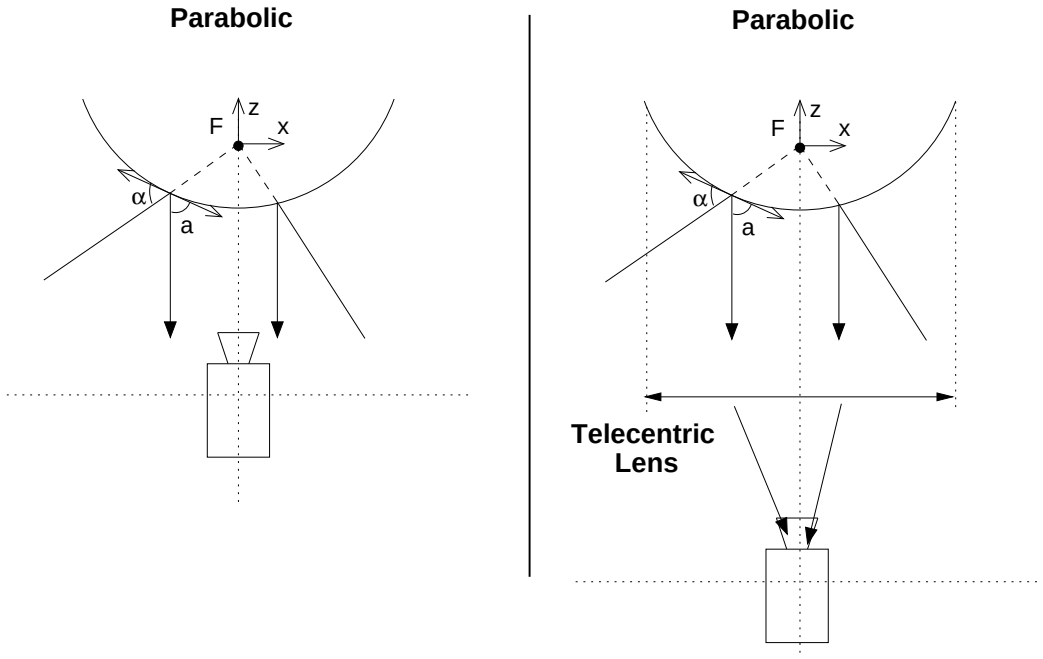


Figure 2.9: Telecentric lens added to a parabolic sensor to guarantee an orthographic projection model

Finally, the following function can be used to model the radial and tangential distortion of points on the normalised plane:

$$\begin{aligned} \mathbf{m}_d &= \mathbf{m}_u + \mathbf{m}_u R(\rho_u) + T(\mathbf{m}_u) \\ &= \mathbf{m}_u + D(\mathbf{m}_u, V) \\ &= \mathbf{m}_u + \begin{bmatrix} x_u(k_1\rho_u^2 + k_2\rho_u^4) + 2p_1x_uy_u + p_2(\rho_u^2 + 2x_u^2) \\ y_u(k_1\rho_u^2 + k_2\rho_u^4) + 2p_2x_uy_u + p_1(\rho_u^2 + 2y_u^2) \end{bmatrix} \end{aligned} \tag{2.6}$$

2.3.2 Inverse distortion model

Equation (2.6) is not analytically invertible. Had we only considered one radial parameter (which is often sufficient), we could solve the resulting 3rd order polynomial using Cardan's method. Here we wish to invert the function with the four parameters (k_1, k_2, p_1, p_2). To find the undistorted points $\mathbf{m}_u$ from $\mathbf{m}_d$, we could for example solve the non-linear least-square problem associated to the inversion but this would be costly in terms of computation and we are not sure to converge to the optimal solution. Several methods exist to obtain an approximation [Heikkilä, 2000].

We will start by taking the first order Taylor series expansion for D about $\mathbf{m}_u = \mathbf{m}_d$:

$$D(\mathbf{m}_u, V) \approx D(\mathbf{m}_d, V) + \underbrace{\left[\left.\frac{\partial D}{\partial x}\right|_{x_d} \quad \left.\frac{\partial D}{\partial y}\right|_{y_d} \right]}_{\mathbf{J}(\mathbf{m}_d)} (\mathbf{m}_u - \mathbf{m}_d) \tag{2.7}$$

Thus (from (2.6)):

$$\mathbf{m}_d \approx \mathbf{m}_u + D(\mathbf{m}_d, V) + \mathbf{J}(\mathbf{m}_d)(\mathbf{m}_u - \mathbf{m}_d) \tag{2.8}$$

Solving for $\mathbf{m}_u$, we obtain:

$$\mathbf{m}_u \approx \mathbf{m}_d - (\mathbf{I}_{2\times 2} + \mathbf{J}(\mathbf{m}_d))^{-1} D(\mathbf{m}_d, V) \tag{2.9}$$

$$\mathbf{J}(\mathbf{m}_d) = \begin{bmatrix} k_1(\rho_d^2 + 2x^2) + k_2\rho_d^2(\rho_d^2 + 4x^2) + p_1 2y + p_2 6x & k_1 2xy + k_2 4\rho_d^2 xy + p_1 2x + p_2 2y \\ k_1 2xy + k_2 4\rho_d^2 xy + p_1 2x + p_2 2y & k_1(\rho_d^2 + 2y^2) + k_2\rho_d^2(\rho_d^2 + 4y^2) + p_1 6y + p_2 2x \end{bmatrix} \tag{2.10}$$

$$(\mathbf{I}_{2\times 2} + \mathbf{J}(\mathbf{m}_d))^{-1} D(\mathbf{m}_d, V) = \begin{bmatrix} \frac{D_1 + D_1\mathbf{J}_{22} - \mathbf{J}_{12}D_2}{1+\mathbf{J}_{11}+\mathbf{J}_{12}+\mathbf{J}_{11}\mathbf{J}_{22}-\mathbf{J}_{12}\mathbf{J}_{21}} \\ \frac{D_2 + D_2\mathbf{J}_{11} - \mathbf{J}_{21}D_1}{1+\mathbf{J}_{11}+\mathbf{J}_{12}+\mathbf{J}_{11}\mathbf{J}_{22}-\mathbf{J}_{12}\mathbf{J}_{21}} \end{bmatrix} \tag{2.11}$$

Under the assumption that the distortion values and the jacobian values are small, we obtain the model proposed by Heikkilä [Heikkilä, 2000]:

$$\mathbf{m}_u \approx \mathbf{m}_d - \frac{D(\mathbf{m}_d, V)}{1 + D_{11}(\mathbf{m}_d, V) + D_{22}(\mathbf{m}_d, V)} \tag{2.12}$$

$$D_{11}(\mathbf{m}_d, V) + D_{22}(\mathbf{m}_d, V) = 4k_1\rho_d^2 + 6k_2\rho_d^4 + 8p_1 y_d + 8p_2 x_d \tag{2.13}$$

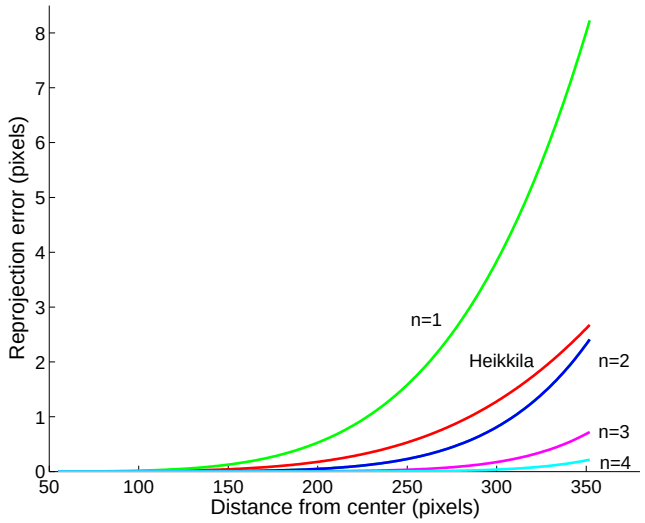


Figure 2.10: Reprojection error (n corresponds to the number of iterations of the recursive estimation model)

For the sensor used in our study, this approximation proved insufficient to correct the relatively strong distortion $k_1 = -0.07$ (Figure 2.10).

In [Mallon and Whelan, 2004], the authors propose a more precise model obtained by estimating independently the variables of the numerator and denominator. However this approach is made less attractive by imposing a minimisation step (using images from lines or planes) to estimate these variables.

Alternatively, the following sequence can be used:

$$\begin{cases} \mathbf{m}_u &= \mathbf{m}_d - D^n \\ D^n &= D(\mathbf{m}_d - D^{n-1}, V) \\ D^1 &= D(\mathbf{m}_d, V) \end{cases} \tag{2.14}$$

It converges towards the correct inverse if the distortion parameters (k_1, k_2, p_1, p_2) are strictly inferior to 1. For our case, $k_1 = -0.07$, four iterations were sufficient (Figure 2.10). Applying this method is computationally expensive, however once obtained, the lifting of each image point can be pre-computed and saved in a lookup table (LUT).

2.4 An overview of projection models

Several projection models have been proposed in the literature with the increasing popularity of omnidirectional vision. The motivation behind is to find a compromise between expressivity of the model and simplicity.

The model proposed by Sturm and Ramalingam [Sturm et al., 2006] for example is very general, it associates a projective ray to each pixel in the image. Thus it can model a very wide range of vision sensors (eg. non-single viewpoint sensors, general imaging devices). However it is difficult to obtain a stable calibration.

In [Tardif et al., 2006] the authors propose a radially symmetric distortion model that encompasses radial non-SVP sensors but stays sufficiently constrained to enable a stable calibration.

The work by Micusik [Micusik, 2004], on the other hand, uses different simplifying hypothesis to generate polyeigenvalue problems (PEPs) for central catadioptric, spherical and fisheye epipolar geometry. As shown by Fitzgibbon in [Fitzgibbon, 2001], the fundamental matrix eight-point problem (and homography estimation) can be extended to take into account distortion. This approach is particularly adapted to outlier rejection methods (RANSAC) and initialising iterative bundle adjustment methods. It does not provide however precise calibration and a final global iterative minimisation is generally required.

The model proposed in this chapter finds a compromise by assuming small errors from the ideal theoretical model. It has a clearly identifiable physical meaning. However it cannot model general non-single viewpoint sensors. Furthermore the non-linear projection function means it is not well suited to non-calibrated structure from motion (in particular for RANSAC type of approaches). This is not a strong limitation in robotics where sensors are generally calibrated before use.

2.5 Conclusion

In this chapter we presented a projection model based on the unified projection model from Geyer and Barreto. This model was shown to be valid for some fisheye lenses. To take into account the distortion introduced by telecentric lenses or by a misalignment, we added radial and tangential distortion. Compared to other models present in the literature, the proposed model has easily identifiable parameters and presents a compromise between genericity and over-parameterisation. In the following chapter, we will detail how to calibrate the sensor. It will appear that the model leads to a flexible calibration approach and is well adapted to central catadioptric calibration in view of precise robotic applications.

Chapter 3

Calibration from planar grids

Contents

The previous chapter described a projection model for a class of omnidirectional cameras. We now need a method to find the different parameters of the sensor and preferably in an efficient way (without having to select manually too many features). Precise calibration is a crucial step as it will later have an impact over the quality of the reconstruction and motion estimation.

3.1 Model parameters

The projection function described in the previous chapter and noted Π depends on 10 parameters:

1. ξ that is function of the mirror shape,
2. k_1, k_2, p_1 and p_2 that model the radial and tangential distortion,
3. γ_1, γ_2, s, u_0 and v_0 that describe the generalised camera model (γ_1 and γ_2 depend on the camera and the mirror shape)

The calibration approach proposed in this chapter relies on minimising the reprojection error of points of a planar grid of known dimension. We made the choice of using standard planar grids because they are commonly available and simple to make. Alternatively, we could have chosen grids adapted to the wide field of view of the sensor as in [Vasseur and Mouaddib, 2004].

The function we wish to minimise is non-linear so we need initial values to hope to converge towards the global minimum.

By assuming that the errors from the theoretical model are small, we have $k_1 \approx k_2 \approx p_1 \approx p_2 \approx s \approx 0$ and $\gamma_1 \approx \gamma_2 \approx \gamma$.

We still need to find the extrinsic parameters of the grids and values for $[\xi, \gamma, u_0, v_0]$. The image center can be used to initialise the principal point (u_0, v_0) or it can be approximated by the center of the mirror border (assumed to be a circle).

Experimentally, we will show that errors in the values of (ξ, γ) do not have a strong influence over the precision of the extraction process for parabolic and hyperbolic sensors (Section 3.4.6). We will start by assuming $\xi = 1$. This value is of course incorrect for non-parabolic sensors but simplifies the projection equations sufficiently to enable the estimation of the focal length from at least three image points that belong to a non-radial line image[1]. Once this step applied, the extrinsic parameters can be estimated from four points of a grid of known size.

The rotation will be represented by a unit quaternion $\mathbf{Q} = [q_0\ q_1\ q_2\ q_3]^\top$ and the translation by $\mathbf{t} = [t_x\ t_y\ t_z]^\top$. Let V be the matrix of parameters:

$$V_{17\times1} = [q_0\ q_1\ q_2\ q_3\ t_x\ t_y\ t_z\ \xi\ k_1\ k_2\ p_1\ p_2\ s\ \gamma_1\ \gamma_2\ u_0\ v_0]^\top$$

$$V^1_{7\times1} = [q_0\ q_1\ q_2\ q_3\ t_x\ t_y\ t_z]^\top,\ V^2_{1\times1} = \xi,\ V^3_{4\times1} = [k_1\ k_2\ p_1\ p_2]^\top,\ V^4_{5\times1} = [s\ \gamma_1\ \gamma_2\ u_0\ v_0]^\top$$

3.2 Calibration method

We suggest the following calibration steps to initialise the unknown parameters, make the associations between the grid points and their reprojection in the image and finally launch the minimisation:

[1] a "line image" is the projection of a 3D line on the image plane

1. initialisation of the principal point (u_0, v_0) thanks to the mirror border (or the center of the image if there is no border) (figure 3.1),

2. estimation of the generalised focal length γ (assuming $\gamma = \gamma_1 = \gamma_2$) thanks to at least three points belonging to a non-radial line image (figure 3.2),

3. for each image, we then select the four edge points of each grid (figure 3.3), estimate the extrinsic parameters and then extract the remaining points by reprojection (figure 3.4),

4. the final calibration step consists in the minimisation of the global reprojection error (using for example the Levenberg-Marquardt algorithm).

Figure 3.1: Extraction of the mirror border for the estimation of the principal point

Figure 3.2: Estimation of the generalised focal length from line image points

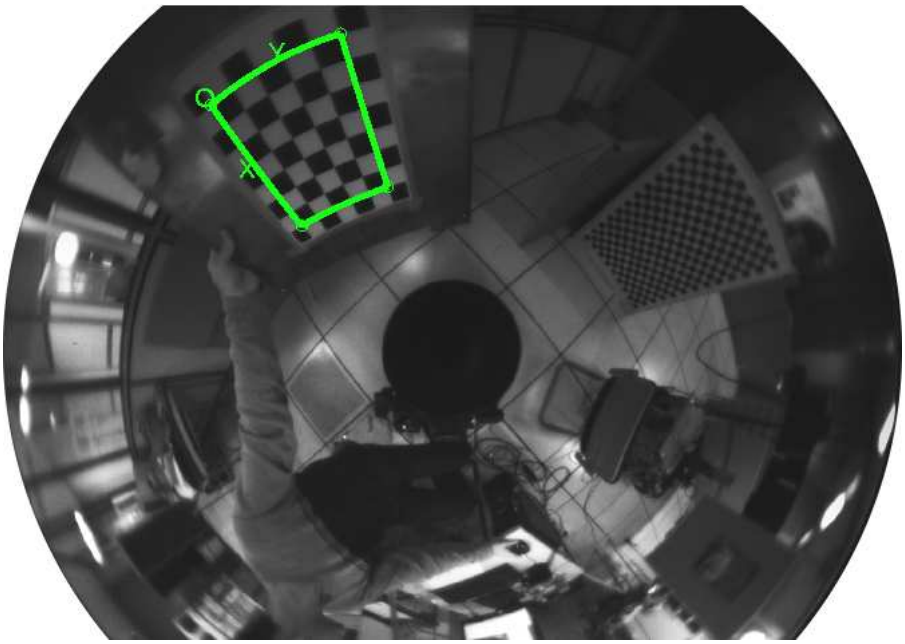


Figure 3.3: Extraction of the four corners belonging to the calibration grid

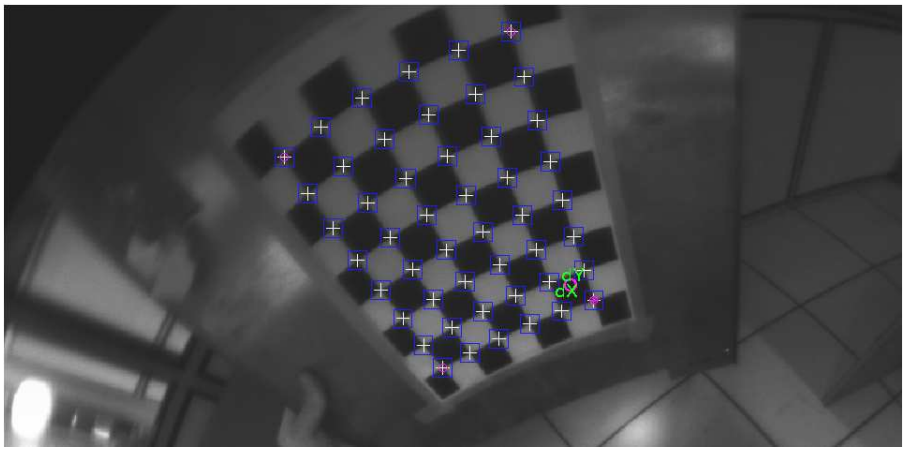


Figure 3.4: Sub-pixel point extraction

3.2.1 Initialisation of the principal point

Under the assumption of small assembly errors, we assume that the mirror border belongs to a plane that is parallel to the image plane, and that the pixels are square. The mirror border is then projected as a circle centered in the principal point (figure 3.1).

The border extraction is quite delicate because of the important quantity of information (and thus of edges) around the mirror border.

We propose a heuristic approach applied to the minimum of intensity of a set of images (the minimum avoids for example the lights on the ceiling creating zones of strong contrast that degrades the precision of the edge extraction). Figure 3.1 illustrates the different steps:

1. we ask the user to select the center of the image and the mirror border (the two '+' signs in image 3.1, the corresponding circle is illustrated by the blue dashed circle),
2. we then eliminate points that are too far or too close to the mirror border,
3. from the remaining points, we select random samples of minimum size to find potential circles. We then choose the median of the obtained values (the final circle is represented in a solid red line).

3.2.2 Estimation of the focal length

We will call "line image" the projection of a 3D line in the image plane. A more in depth study of lines will be undertaken in Chapter 7.

From equation (2.1), with $\xi = 1$, we obtain the following projective equation:

$$\begin{cases} \hbar^{-1}(\mathbf{m}) \sim \begin{bmatrix} x \\ y \\ f(x,y) \end{bmatrix} \\ f(x,y) = \frac{1}{2} - \frac{1}{2}(x^2 + y^2) \end{cases} \tag{3.1}$$

Let $\mathbf{p} = (u, v)$ be a point in the image plane. Thanks to the estimation of the principal point, we can center the points and calculate a corresponding point $\mathbf{p}_c = (u - u_0, v - v_0) = (u_c, v_c)$. This point follows the equation on the normalised plane that depends on γ: $\mathbf{p}_c = \gamma\mathbf{m}$:

$$\begin{cases} \hbar^{-1}(\mathbf{m}) \sim \begin{bmatrix} u_c \\ v_c \\ g(u_c,v_c) \end{bmatrix} \\ g(u_c,v_c) = \frac{\gamma}{2} - \frac{1}{2\gamma}(u_c^2 + v_c^2) \end{cases} \tag{3.2}$$

Let us assume the point belongs to a line image. The line image can be parameterised by the normal $\mathbf{n} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^\top$ of the plane spanned by the 3D line and the center $\boldsymbol{\mathcal{C}}_m$ of the mirror (see Chapter 7 for more details), we then obtain the projective property:

$$\hbar^{-1}(\mathbf{m})^\top \mathbf{n} = 0 \Longleftrightarrow \begin{cases} n_x u_c + n_y v_c + \frac{a}{2} - b\frac{u_c^2+v_c^2}{2} = 0 \\ a = \gamma n_z \\ b = \frac{n_z}{\gamma} \end{cases}$$

Let us assume, we have n points $\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n$ belonging to a same line image, they verify the system:

$$\mathbf{P}_{n\times 4}\mathbf{c}_{4\times 1} = 0$$

with:

$$\mathbf{P} = \begin{bmatrix} u_{c1} & v_{c1} & \frac{1}{2} & -\frac{u_{c1}^2+v_{c1}^2}{2} \\ u_{c2} & v_{c2} & \frac{1}{2} & -\frac{u_{c2}^2+v_{c2}^2}{2} \\ \vdots & \vdots & \vdots & \vdots \\ u_{cn} & v_{cn} & \frac{1}{2} & -\frac{u_{cn}^2+v_{cn}^2}{2} \end{bmatrix} \tag{3.3}$$

By singular value decomposition (SVD), $\mathbf{P} = \mathbf{USV}^\top$. The least square solution is obtained from the last column of $\mathbf{V}$ associated to the smallest singular value.

To obtain $\mathbf{n}$ and in particular γ from $\mathbf{c} = [c_1\ c_2\ c_3\ c_4]^\top$, the following steps can be applied:

1. Calculate $t = c_1^2 + c_2^2 + c_3 c_4$ and check that $t > 0$.
2. Let $d = \sqrt{1/t}$, $n_x = c_1 d$ and $n_y = c_2 d$.
3. We check that $n_x^2 + n_y^2 >$ threshold (for example threshold $= 0.95$) to be sure the line image is not radial,
4. If the line is not radial, $n_z = \sqrt{1 - n_x^2 - n_y^2}$.
5. Finally, $\gamma = \frac{c_3 d}{n_z}$

We may note that this process can in fact be applied to three randomly chosen points in the image to obtain an estimate of the focal length in a RANSAC fashion. This way we obtain an auto-calibration approach.

3.3 Cost function

In this section we will detail the calibration cost function and the Jacobians needed for the minimisation.

Let us assume we have m 3D points $\boldsymbol{\mathcal{X}}_i$ corresponding to the grid edges and their images $\mathbf{p}_i$:

$$\forall i \in [1..m],\ \boldsymbol{\mathcal{X}}_i \leftrightarrow \mathbf{p}_i$$

Let P be the projection function and V the parameters we are looking for. With W the function that transforms the grid points to the mirror frame and S the function that projects 3D points to the sphere $P = \Pi \circ S \circ W$. In practice, Π depends on the generalised projection matrix and on the distortion, so we will decompose P as $P = k \circ D \circ \hbar \circ S \circ W$.

The cost function is then:

$$F(V) = \frac{1}{2} \sum_{i=1}^{m} [P(V, \boldsymbol{\mathcal{X}}_i) - \mathbf{p}_i]^2$$

3.3.1 Changing frame

Let W be the first transformation:

$$W(\boldsymbol{\mathcal{X}}, V^1) = \mathbf{R}\boldsymbol{\mathcal{X}} + \mathbf{t}$$

A rotation by a quaternion $\mathbf{Q} = [q_0\ q_1\ q_2\ q_3]^\top$ can be written:

$$\mathbf{R}(\mathbf{Q}) = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(-q_0q_3+q_1q_2) & 2(q_0q_2+q_1q_3) \\ 2(q_0q_3+q_1q_2) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(-q_0q_1+q_2q_3) \\ 2(-q_0q_2+q_1q_3) & 2(q_0q_1+q_2q_3) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix}$$

For the quaternion to represent a rotation, it must be of unit length, we normalise it at each step: W can then be written with $\mathbf{R}_{\boldsymbol{\mathcal{X}}}(\mathbf{Q}') = \mathbf{R}(\mathbf{Q}')\boldsymbol{\mathcal{X}}$ with $\mathbf{Q}' = \frac{\mathbf{Q}}{\|\mathbf{Q}\|}$:

$$W(\boldsymbol{\mathcal{X}}, V^1) = \mathbf{R}_{\boldsymbol{\mathcal{X}}}(\mathbf{Q}') + \mathbf{t}$$

The final jacobian is then (the detailed values are given in Appendix A.1):

$$\frac{\partial W}{\partial V^1}_{3\times 7} = \begin{bmatrix} \left(\frac{\partial W}{\partial \mathbf{Q}'}\frac{\partial \mathbf{Q}'}{\partial \mathbf{Q}}\right)_{3\times 4} & \left(\frac{\partial W}{\partial \mathbf{t}}\right)_{3\times 3} \end{bmatrix}$$

3.3.2 Mirror transformation

Let $H = \hbar \circ S$, we will need to calculate $\frac{\partial H}{\partial \boldsymbol{\mathcal{X}}}_{2\times 3}$ and $\frac{\partial H}{\partial V^2}_{2\times 1}$ (the detailed values can be found in Appendix A.2).

3.3.3 Distortion

The distortion applies the function D, with Jacobians $\frac{\partial D}{\partial V^3}_{2\times 4}$ and $\frac{\partial D}{\partial \boldsymbol{\mathcal{X}}}_{2\times 2}$ (the detailed values can be found in Appendix A.3).

3.3.4 Generalised projection matrix

With k the projection function we have $\frac{\partial k}{\partial V^4}_{2\times 5}$ and $\frac{\partial k}{\partial \boldsymbol{\mathcal{X}}}_{2\times 2}$ (the detailed values can be found in Appendix A.4).

3.3.5 Final equation

By chain composition, we obtain the jacobian of $P = k \circ D \circ H \circ W$:

$$\frac{\partial P}{\partial V} = \left[\frac{\partial k}{\partial D}_{2\times 2}\left[\frac{\partial D}{\partial H}_{2\times 2}\left[\frac{\partial H}{\partial W}_{2\times 3}\frac{\partial W}{\partial V^1}_{3\times 7} \quad \frac{\partial H}{\partial V^2}_{2\times 1}\right] \quad \frac{\partial D}{\partial V^3}_{2\times 4}\right]_{2\times 12} \quad \frac{\partial P}{\partial V^4}_{2\times 5}\right]_{2\times 17}$$

We may note that we have calculated the Jacobian of $\Pi_S = \Pi \circ S$, noted $\mathbf{J}_{\Pi_S} = \frac{\partial k}{\partial D}_{2\times 2}\frac{\partial D}{\partial H}_{2\times 2}\frac{\partial H}{\partial \boldsymbol{\mathcal{X}}}_{2\times 3}$ that will appear when estimating the camera motion in Part II and III.

3.4 Experimental validation

The calibration approach was tested with five different configurations: parabolic, hyperbolic, folded mirror, wide-angle and spherical sensors. A different camera was used each time. For the experiments, we used the "Omnidirectional Calibration Toolbox", an opensource software freely available for download[2] and that we implemented following the described calibration method. The spherical sensor is not theoretically a single viewpoint sensor but we will see that it can be approximated well by the

[2] `http://www-sop.inria.fr/icare/personnel/Christopher.Mei/Toolbox.html`

proposed model. Previous studies have also shown the validity of single viewpoint approximations [Micusik, 2004].

To validate our model, we need to obtain a low residual error after minimisation and a uniform distribution of the error. It is not necessary do check the error distribution over the complete image as we assume the sensor is rotationally symmetric and that the tangential distortion was only needed for the folded and spherical catadioptric sensors (according to the covariance estimates). For each calibration experiment, we will show the radial distribution of the error with a curve representing the median value of the error for different intervals of ρ, the distance of a pixel from the principal point.

We may note that polynomial approximations are often valid only locally and badly approximate the projection around the edges. This bias will have a negative impact for example when estimating the motion of the camera using a maximum likelihood estimation under the assumption of a Gaussian distribution of the error.

The results will be summarised in a table containing the initial values after steps (0) and (1) of the calibration process and the results after the minimisation. e_x and e_y indicate the reprojection error in pixels. (b) means the mirror border was used to initialise the principal point.

Figure 3.5: S80 paracatadioptric sensor

3.4.1 Calibration of the parabolic sensor

The parabolic sensor (we fixed $\xi = 1$) used in this study consists of a S80 parabolic mirror from RemoteReality (Figure 3.5) with a telecentric lens and a perspective camera with an image resolution of 2048×1016. The calibration points were obtained from 8 images of a grid of size 6×8 with squares of 42 mm. Table 3.1 summarises the results. After minimisation, we can see that the error is correctly distributed over the image (Fig. 3.6).

Distortion If we do not take into account the distortion during the calibration, the error increases to $[0.74, 0.82]$ which confirms that the radial distortion function is needed to account for the error induced by the telecentric lens.

Initialisation	$\xi = 1$ was fixed	
$[u_0,v_0]$ (b) = [983.7, 545.2], $\gamma = 569$, $[e_x,e_y] = [1.92, 2.02]$		
Final	Values	3σ
$[u_0,v_0]$	[980.87, 545.02]	[3.73, 4.11]
$[\gamma_1,\gamma_2]$	[598.66, 597.69]	[6.52, 7.75]
$[e_x,e_y]$	[0.18, 0.31]	[0.48, 0.84]
$[k_1, k_2]$	[-0.088, 0.017]	[0.007, 0.004]

Table 3.1: Calibration results for the parabolic sensor

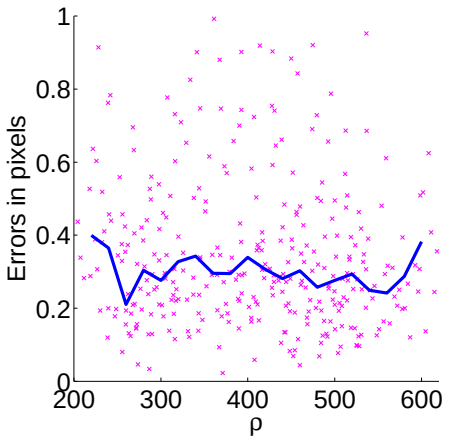


Figure 3.6: Pixel error versus distance to center for the parabolic sensor

3.4.2 Calibration of the hyperbolic sensor

In the hyperbolic case, the mirror is a HM-N15 from Accowle (Seiwapro) with a perspective camera with an image resolution of 800×600. 6 images of a grid of size 8×10 with squares of 30 mm were taken. Table 3.2 summarises the results. After minimisation, we can see a slight bias in the error that is more important in the center of the image (Fig. 3.7).

Initialisation		
$[u_0,v_0]$ (b) = [390.7,317.7], $\gamma = 270$, $[e_x,e_y] = [1.02, 1.24]$		
Final	Values	3σ
$[u_0,v_0]$	[386.54, 321.69]	[1.70, 1.61]
$[\gamma_1,\gamma_2]$	[242.11, 241.05]	[9.54, 9.32]
$[e_x,e_y]$	[0.29, 0.30]	[0.81, 0.78]
$[\xi]$	0.780	
$[k_1, k_2]$	[-0.101, 0.013]	[0.0120, 0.0013]

Table 3.2: Calibration results for the hyperbolic sensor

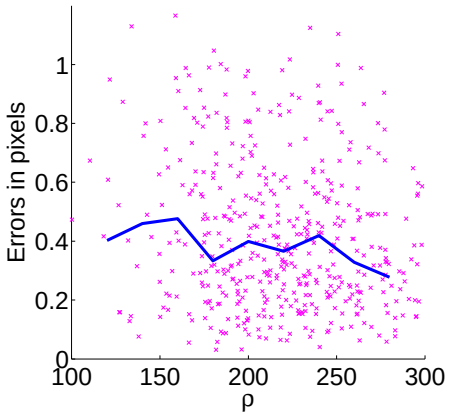


Figure 3.7: Pixel error versus distance to center for the hyperbolic sensor

3.4.3 Calibration of a folded catadioptric camera

Folded catadioptric sensors combine typically two mirrors and follow the single viewpoint constraint [Nayar and Peri, 2001]. They have the advantage of being compact and relatively cheap to manufacture. The 10 images used had a resolution of 640×480, the grids had a size of 6×8 with squares of 30 mm.

Table 3.3 summarises the results. We can see a very slight bias in the error that is stronger around the mirror border (Fig. 3.8).

Initialisation		
$[u_0,v_0]$ (b) = [298.5, 269.5], γ = 156.14, $[e_x,e_y]$ = [0.37, 0.58]		
Final	Values	3σ
$[u_0,v_0]$	[299.28, 267.42]	[1.31, 1.18]
$[\gamma_1,\gamma_2]$	[138.20, 134.78]	[3.35, 3.26]
$[e_x,e_y]$	[0.123, 0.193]	[0.293, 0.456]
$[\xi]$	0.72	
$[k_1, k_2, p_1, p_2]$ $\times 1e^{-3}$	[-104, 11.5, -2.51, 2.5]	[5.12, 0.78, 0.58, 0.5]

Table 3.3: Calibration results for the folded catadioptric camera

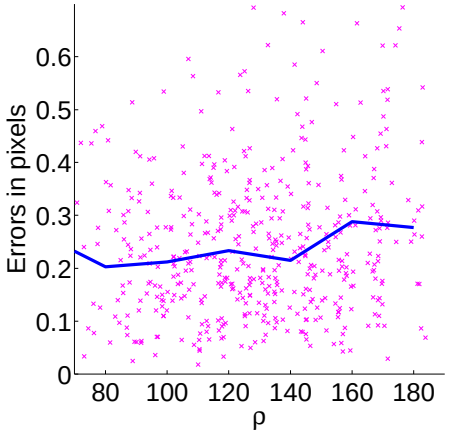


Figure 3.8: Pixel error versus distance to center for the folded catadioptric camera

3.4.4 Calibration of a wide-angle sensor

The calibration was also tested on a wide-angle sensor ($\sim 70^o$) on 21 images of resolution 320×240 . The grid used was the same as in the hyperbolic case. For the wide-angle sensor, there is no border so the center of the image was taken to initialise the principal point. Table 3.4 summarises the results. As before, we can see a very slight bias towards the edges in Figure 3.9.

Initialisation		
$[u_0,v_0]$ = [160, 120], γ = 448, $[e_x,e_y]$ = [0.73, 0.69]		
Final	Values	3σ
$[u_0,v_0]$	[166.40, 110.23]	[1.65, 1.19]
$[\gamma_1,\gamma_2]$	[635.91, 641.02]	[0.38, 0.38]
$[e_x,e_y]$	[0.13, 0.14]	[0.36, 0.42]
$[\xi]$	1.40	
$[k_1, k_2]$	[-0.88, 2.76]	[0.087, 1.28]

Table 3.4: Calibration results for the wide-angle sensor

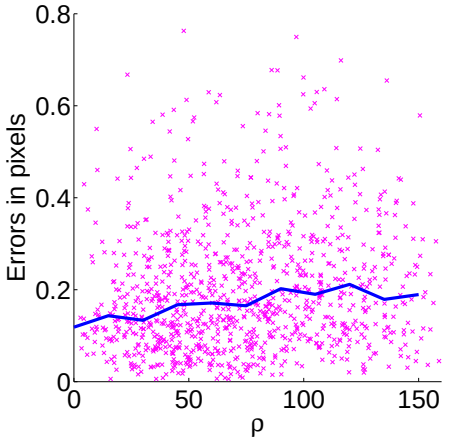


Figure 3.9: Pixel error versus distance - wide-angle sensor

The strong change in γ after minimisation is probably due to the radial distortion and the change in ξ. The value of ξ does not have a simple interpretation for wide-angle sensors.

3.4.5 Calibration of a camera with a spherical mirror

Finally we tested a low-quality camera consisting of a webcam in front of a spherical ball. The image resolution was of 352×264. 7 images were used with a similar grid as in the parabolic case.

The border extraction process did not prove very efficient for this sensor so the image center was used as an initial value for the principal point. Table 3.5 summarises the results.

Figure 3.10 shows the radial distribution of the error. The error seems to be distributed uniformly in the image (Fig. 3.10).

Initialisation		
$[u_0,v_0] = [184,\ 127]$, $\gamma = 137.7$, $[e_x,e_y] = [0.65, 0.62]$		
Final	Values	3σ
$[u_0,v_0]$	[183.10, 126.31]	[1.33, 0.30]
$[\gamma_1,\gamma_2]$	[164.79, 162.84]	[9.36, 9.31]
$[e_x,e_y]$	[0.16, 0.15]	[0.36, 0.33]
$[\xi]$	0.945	
$[k_1, k_2, p_2]$	[-0.322, 0.067, 0.0043]	$[7.6, 6.3, 2.0]\times 1e^{-3}$

Table 3.5: Calibration results for the spherical mirror

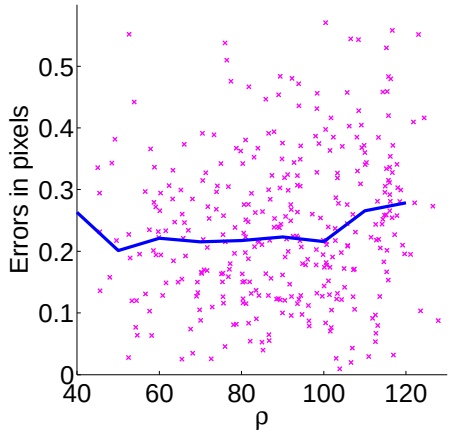


Figure 3.10: Pixel error versus distance - spherical sensor

3.4.6 Point extraction

Table 3.6: Influence of errors in (ξ, η) over the point extraction process

% error in (ξ, η)	0	10	20	30	40
% of correct points	99.7	88	81	83	76

To analyse the effect of errors on the mirror parameters over the point extraction process, we counted the amount of correctly extracted points obtained after the extrinsic parameters were estimated from four points, and the grid was reprojected followed by a subpixel precision extraction. The test was done for a parabolic sensor. However the fact that we could calibrate the sensors previously indicates that the observation is likely to be valid for the sensors that follow the chosen projection model. Table 3.6 summarises the results for errors in (ξ, η) ranging from 0 to 40 %. These values indicate that the extraction process presents a certain robustness to imprecise initial values. We still managed to calibrate the sensor with an error of 40 %.

3.5 Conclusion

An omnidirectional sensor has a non-linear projection function. To initialise the parameters, we devised simple calibration steps that do not require to know the mirror parameters. Thanks to the model based on small errors, we only need to select four grid corners per calibration grid and not all the points. In the previous chapter, we had justified theoretically that the projection model can be used for central catadioptric and fisheye sensors. These results were confirmed experimentally with the calibration of a wide range of sensors used in robotics.

Chapter 4

Calibration between an omnidirectional sensor and a laser range finder

Contents

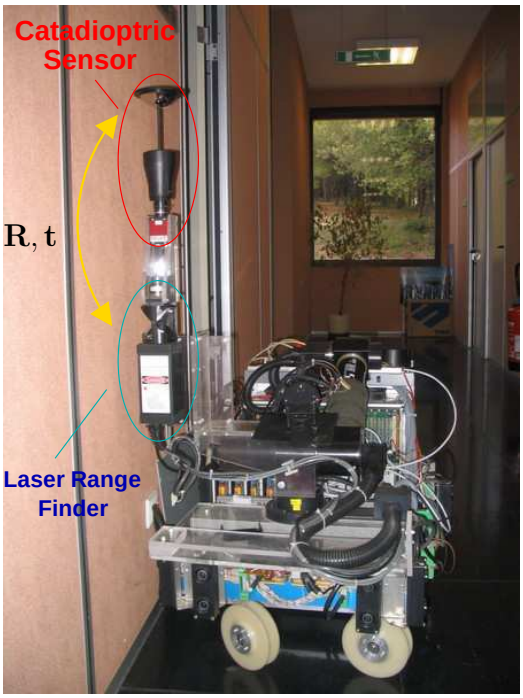


Figure 4.1: Calibration between an omnidirectional sensor and a laser range finder (Anis robot)

4.1 Calibration

In this chapter we will describe the calibration between an omnidirectional sensor and a laser range finder, in other words how to find the relative position (rotation $\mathbf{R}$ and translation $\mathbf{t}$) between the sensors (figure 4.1). We assume that each sensor has been calibrated separately. The calibration is an essential step to combine the data. Few studies have been done on the subject. Zhang and Pless [Zhang and Pless, 2004] propose a method for standard perspective cameras and a laser range finder with an invisible laser beam. The idea is to combine a plane with known position (calibration grid) with the calculated distances obtained from the laser range finder. The work in Section 4.3.2 was based on this method. In [Dupont et al., 2005], the authors improve the precision obtained by replacing the algebraic error by a distance with a geometric meaning.

We will analyse two distinct cases: the case where the laser beam is visible in the omnidirectional image and the case where it is invisible (close infrared).

In the first case, we consider the association between 3D laser points and points in the image but also the association between 3D lines extracted in the laser range scan and line images.

In the second case, we analyse if associating edges in the image to edges in the laser scan is sufficient to calibrate the sensor. We will see that this is not the case and will propose an alternative algorithm that uses the position of 3D planes visible in the image and which intersect the laser plane.

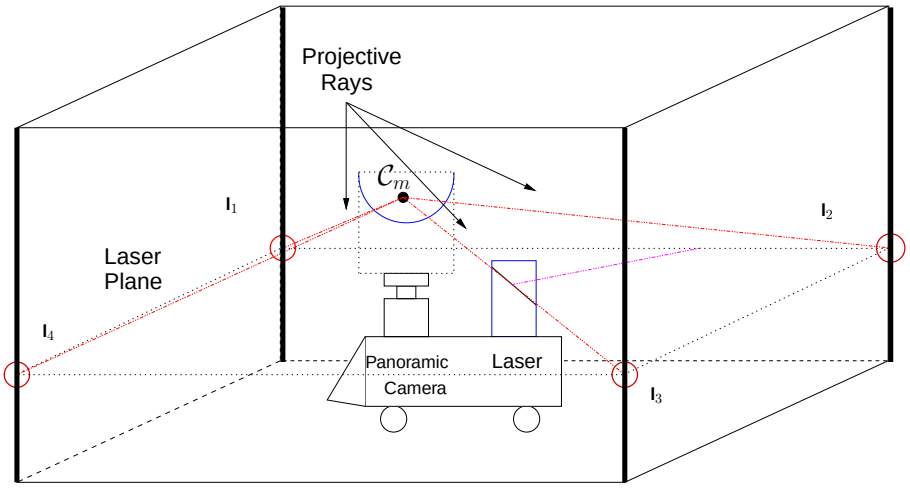


Figure 4.2: Association between 3D laser points and points in the image

4.2 Visible laser beam

4.2.1 Association between points

With some laser range finders, it is possible to make range measurements for different angles. If we simultaneously take an omnidirectional image, we obtain associations between 3D points and image points. More specifically, as we assume the omnidirectional sensor to be calibrated, we have the associations between projective rays and 3D points (figure 4.2).

Under these conditions, finding the relative position is the standard problem of photogrammetry called the "PnP problem" (Perspective from n Points) or "model-based pose estimation" [Fischler and Bolles, 1981]. To take into account the large field of view, the spherical perspective projection should be used as explained in Section 2.1.2.

4.2.1.1 Associated equations

Let $\mathbf{l}_i$ be the 3D points belonging to the laser scan and $\mathbf{p}_i$ the associated points in the omnidirectional image. The solution to the problem is obtained by solving the following non-linear equation:

$$\begin{cases} \min_{\mathbf{R},\mathbf{t}} \frac{1}{2} \sum_{i=1}^{n} \| f_i(\mathbf{R},\mathbf{t},\mathbf{l}_i,\mathbf{p}_i) \|^2 \\ f_i(\mathbf{R},\mathbf{t},\mathbf{l}_i,\mathbf{p}_i) = \Pi(\mathbf{R}\mathbf{l}_i + \mathbf{t}) - \mathbf{p}_i \end{cases}$$

We can initialise the minimisation with distances measured on the robot.

4.2.1.2 Experimental validation

Figure 4.3 shows different laser measurements made of the environment. By subtracting a reference image to an image taken while the laser was pointing in a specific direction, we associate the laser scan points to the image points (figure 4.4). (The laser range finder can also give the angle at which the measurement was taken but this extra information was not necessary.) After minimisation, we obtain the relative position between the sensors. Figure 4.4 shows the reprojection of the points by a + sign. The reprojection error in this case was of 1.37 ± 1.35 pixels.

To insure that the approach leads to repeatable and coherent results, the method was tested several times. The estimated translation (figure 4.5) and rotation (figure 4.6) obtained at each trial shows the coherence of the approach.

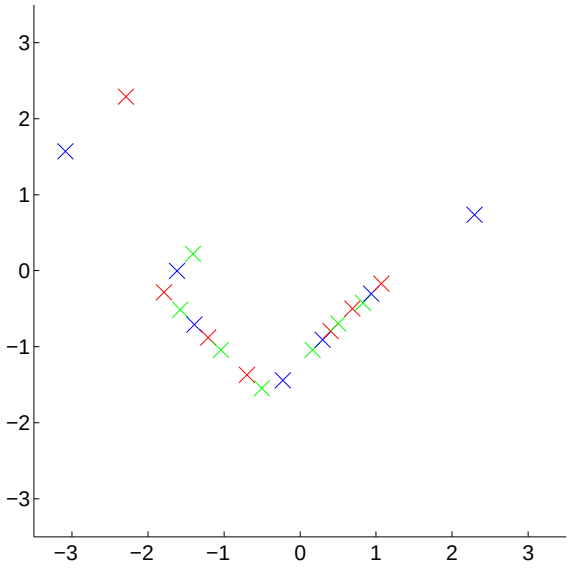


Figure 4.3: Laser measurements made of the environment

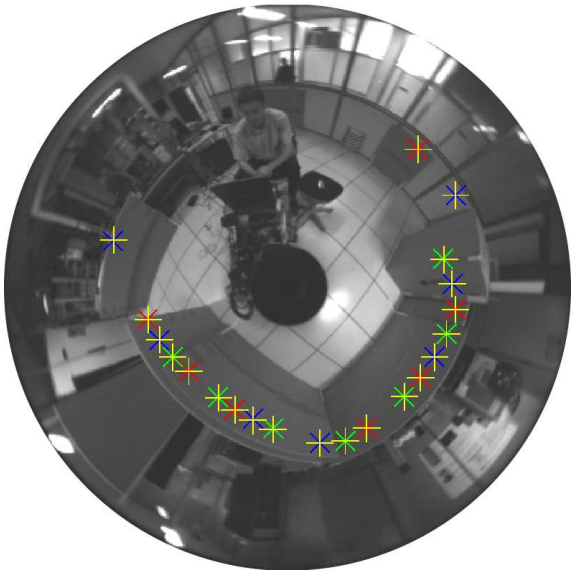


Figure 4.4: Extracted laser points (×) and reprojected points (+)

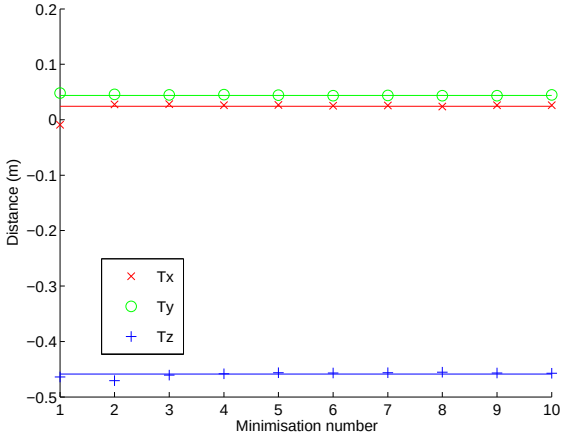


Figure 4.5: Estimation of the translation

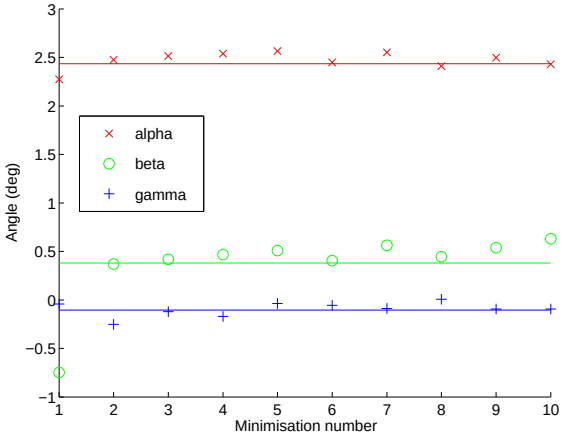


Figure 4.6: Estimation of the rotation

Table 4.1 summarises the results.

4.2.2 Association between lines

We will now study the case where we cannot make measurements at specific angles but we need to process the visible laser range scan directly. We will associate 3D lines extracted from the scan with

Table 4.1: Estimation of the parameters

t	σ	**R** (deg)	σ	Error in pixels	σ
0.026	0.0011	2.49	0.051		
0.044	0.00062	0.49	0.083	1.37	1.35
−0.46	0.0016	−0.089	0.050		

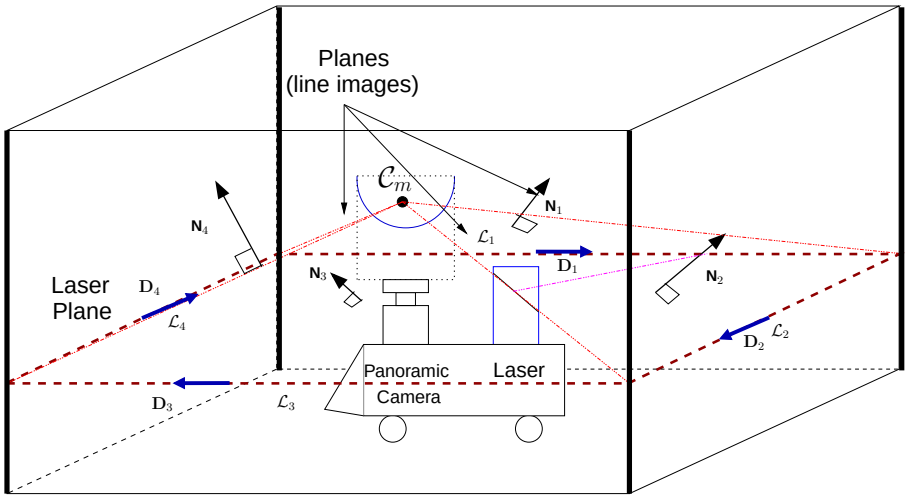


Figure 4.7: Association between 3D lines in the laser scan and line images

lines obtained in the omnidirectional image (figure 4.7).

4.2.2.1 Associated equations

Let $(\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n)_{\boldsymbol{\mathcal{F}}_l}$ be n lines extracted from the laser data. They will be imaged as curves by the catadioptric sensor (see Chapter 7 for more information on line images). These curves can be parameterised by the normals $(\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_n)_{\boldsymbol{\mathcal{F}}_m} \in (\mathbb{R}^3)^n$ to the planes formed by the mirror center $\boldsymbol{\mathcal{C}}_m$ and the laser lines.

A line $\mathcal{L}_i$ can be described by its direction vector $\mathbf{d}_i \in \mathbb{R}^3$ and a point $\boldsymbol{\mathcal{P}}_i \in \mathbb{R}^3$. If we change reference frame: $\{(\mathcal{L}_i)_{\boldsymbol{\mathcal{F}}_l} : (\mathbf{d}_i, \boldsymbol{\mathcal{P}}_i)\} \longrightarrow \{(\mathcal{L}_i)_{\boldsymbol{\mathcal{F}}_m} : (\mathbf{R}\mathbf{d}_i, \mathbf{R}\boldsymbol{\mathcal{P}}_i + \mathbf{t})\}$.

This leads to the constraints:

$$\begin{cases} \mathbf{n}_i^\top \mathbf{R}\mathbf{d}_i = 0 \\ \mathbf{n}_i^\top (\mathbf{R}\boldsymbol{\mathcal{P}}_i + \mathbf{t}) = 0 \end{cases} \tag{4.1}$$

Thus we have the following decoupled minimization problem (algebraic error):

$$\begin{cases} \min_{\mathbf{R}} \frac{1}{2} \sum_{i=1}^{n} \|t_i(\mathbf{R}, \mathbf{n}_i, \mathbf{d}_i)\|^2 \\ t_i(\mathbf{R}, \mathbf{n}_i, \mathbf{d}_i) = \mathbf{n}_i^\top \mathbf{R}\mathbf{d}_i \\ \mathbf{N}\mathbf{t} = -\mathbf{P} \end{cases} \tag{4.2}$$

with $\mathbf{N} = [\mathbf{n}_1^\top,\ \mathbf{n}_2^\top,\ \ldots\ \mathbf{n}_n^\top]^\top$ and $\mathbf{P} = [\mathbf{n}_1^\top \mathbf{R}\boldsymbol{\mathcal{P}}_1,\ \mathbf{n}_2^\top \mathbf{R}\boldsymbol{\mathcal{P}}_2,\ \ldots\ \mathbf{n}_n^\top \mathbf{R}\boldsymbol{\mathcal{P}}_n]^\top$

The second linear problem can be solved by using the pseudo-inverse:

$$\mathbf{t} = -(\mathbf{N}^\top \mathbf{N})^{-1} \mathbf{N}^\top \mathbf{P}$$

4.2.2.2 Validation

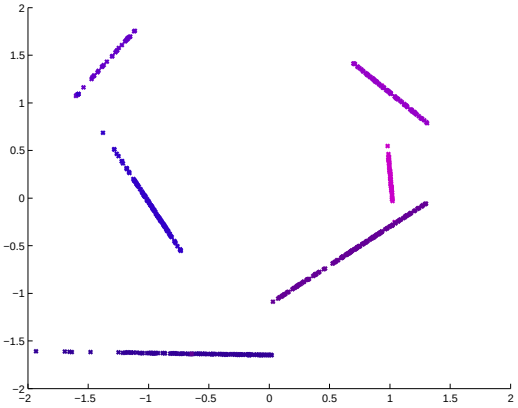


Figure 4.8: Line extraction in the laser plane

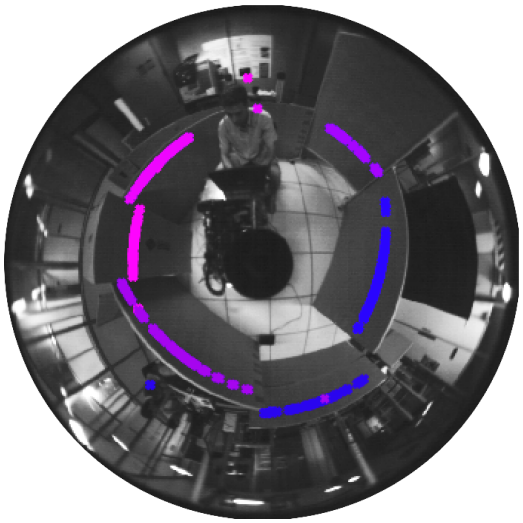


Figure 4.9: Line image extraction in the omnidirectional image

The case of lines is harder to solve than points. The difficulty comes from the data association problem that has an exponential complexity. Table 4.2 summarises the results.

Table 4.2: Parameter estimation

t	**R** (deg)
−0.032	4.26
0.051	−0.30
−0.44	0.86

4.3 Invisible laser

Some lasers emit infrared light that cannot be seen by the camera. (In reality, camera sensors are very sensitive to infrared light so filters are added to restrict the received wave length frequency to visible light...) This situation is more complex than before as we have two sets of data, the first from the camera and the second from the laser. The relationship is indirect and comes from the environment that is observed. In other words, we have to solve a data association problem and ensure that it is sufficient to find the relative position between the sensors.

4.3.1 Edge points

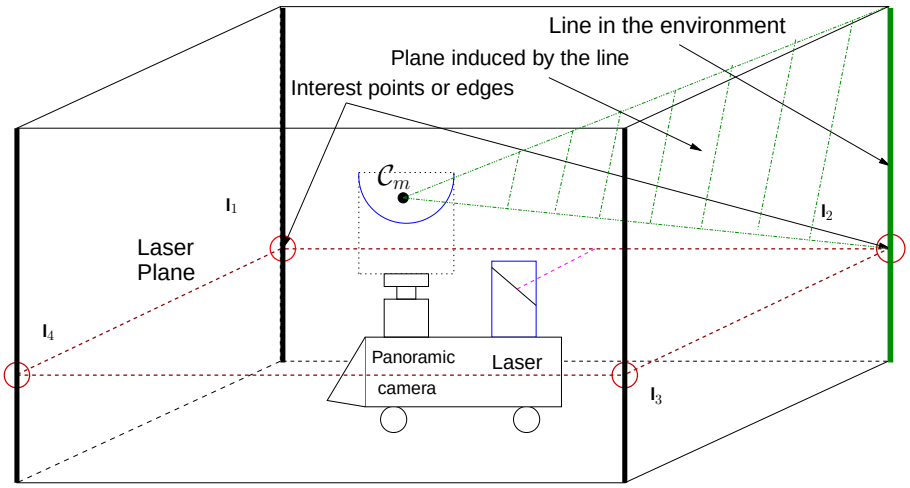


Figure 4.10: Association between laser edge points and lines in the image

Let us assume we can associate edge points in the laser scan to identifiable line images (figure 4.10). The minimization problem can be rewritten as the association between unknown 3D points $(\boldsymbol{\mathcal{X}}_1, \boldsymbol{\mathcal{X}}_2, \ldots, \boldsymbol{\mathcal{X}}_n)$ and the laser points $(\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_n)$ (without loss of generality, we can assume that the laser plane is $Z = 0$). The points $\boldsymbol{\mathcal{X}}_i$ belong to the planes parametrized by $\mathbf{n}_i$: $\mathbf{n}_i^\top \boldsymbol{\mathcal{X}}_i = 0$. This leads to the system of equations:

$$\left\{ \begin{array}{c} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{X}}_1 & \ldots & \boldsymbol{\mathcal{X}}_n \\ 1 & \ldots & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{l}_1 & \ldots & \mathbf{l}_n \\ 0 & \ldots & 0 \\ 1 & \ldots & 1 \end{bmatrix} \\ \mathbf{n}_i^\top \boldsymbol{\mathcal{X}}_i = 0 \end{array} \right. \tag{4.3}$$

In this case, there are $6 + 3 \times n$ unknowns and $3 \times n + n$ equations so at least 6 points and corresponding planes are needed to solve the calibration problem. The condition $rank(\mathbf{N}) = 3$ must also be satisfied or a translation direction is unsatisfied. For example, in Fig. 4.10, the four vertical lines are parallel, $rank(\mathbf{N}) = 2$ and the translation along these lines is not constrained.

Are these two conditions sufficient ? The answer is no and worse than that, however many points and plane associations, three parameters are always missing. The reason comes from the coplanarity of the $\mathbf{l}_i$ points.

Auto-calibration between a central catadioptric sensor and a laser range finder is impossible from a single image in the general case (without 3D point associations).

Proof: Equation (4.3) aims at finding the isometry $(\mathbf{R}, \mathbf{t})$ that transforms a n-tuple $(\boldsymbol{\mathcal{X}}_1, \ldots, \boldsymbol{\mathcal{X}}_n)$ into the polygon defined by $(\mathbf{l}_1, \ldots, \mathbf{l}_n)$. Are the constraints defined on the $\boldsymbol{\mathcal{X}}$ values through the planes with normals $(\mathbf{n}_1, \ldots, \mathbf{n}_n)$ sufficient to define uniquely the $(\boldsymbol{\mathcal{X}}_1, \ldots, \boldsymbol{\mathcal{X}}_n)$ polygon ? The proof established here uses an iterative geometric construction. In brackets are indicated the difference between the number of equations and the number of unknowns.

With 1 point, we have 1 equation but 3 unknowns (-2).

With 2 points, if $rank(\mathbf{n}_1, \mathbf{n}_2) = 2$, we have 2 equations from the normals and 1 equation from the distance but 6 unknowns (-3).

With 3 points, if $rank(\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3) = 3$, we have 3 equations from the normals and 3 distances - these equations are all independent - and 9 unknowns (-3) (see Fig. 4.11 drawn in the plane defined by the three points).

With an extra point $\boldsymbol{\mathcal{X}}_4$, if we add three distance constraints (see Fig. 4.12) two possibilities occur: either they are sufficient to define $\boldsymbol{\mathcal{X}}_4$ uniquely which is the case if the solution is planar, or there are two possible points which are at the intersection of the three spheres centered at $\boldsymbol{\mathcal{X}}_1$, $\boldsymbol{\mathcal{X}}_2$ and $\boldsymbol{\mathcal{X}}_3$. A plane defined by $\boldsymbol{\mathcal{X}}_4$ which does not contain the two points will define uniquely $\boldsymbol{\mathcal{X}}_4$ (-2).

If the solution is not planar, this reasoning can be applied recursively and for $n = 6$ with specific normals $\mathbf{n}$, the system will have a unique solution.

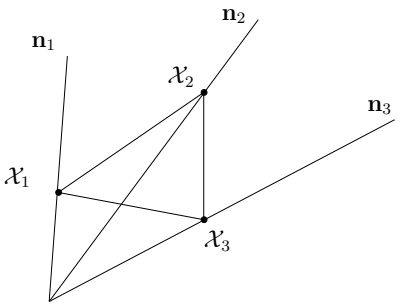


Figure 4.11: Constraints on three points in a plane

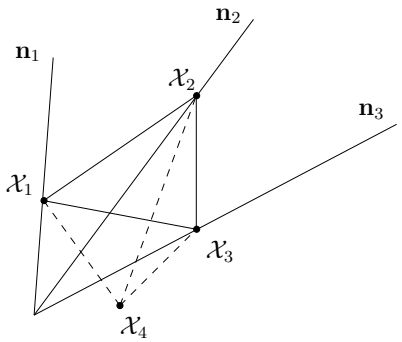


Figure 4.12: Distance constraints on a fourth point

In the case of the laser data, the solution $(\mathbf{l}_1, \ldots, \mathbf{l}_n)$ is planar so 3 extra constraints are missing to solve the system.

□

To summarise, we need a calibration method that uses 3D information.

4.3.2 3D planes

We will now study the case where we know the position of 3D planes with respect to the image sensor and that they intersect with the laser plane, forming lines parameterised by $(\mathbf{d}_i, \boldsymbol{\mathcal{P}}_i)$ (figure 4.13). A simple way of having 3D planes with known equations is to use calibration grids.

We obtain similar constraints as with lines (equation (4.1)) but here the distance d_i to the planes is no longer zero:

$$\begin{cases} \mathbf{n}_i^\top \mathbf{R}\mathbf{d}_i = 0 \\ \mathbf{n}_i^\top (\mathbf{R}\mathbf{p}_i + \mathbf{t}) = d_i \end{cases}$$

This leads to the following decoupled system of equations (algebraic error):

$$\begin{cases} \min_{\mathbf{R}} \frac{1}{2} \sum_{i=1}^{n} \|t_i(\mathbf{R}, \mathbf{n}_i, \mathbf{d}_i)\|^2 \\ t_i(\mathbf{R}, \mathbf{n}_i, \mathbf{d}_i) = \mathbf{n}_i^\top \mathbf{R}\mathbf{d}_i \\ \mathbf{N}\mathbf{t} = -\tilde{\mathbf{P}} \end{cases} \tag{4.4}$$

The same approach as before can be used for the minimisation with:

$$\tilde{\mathbf{P}} = [(\mathbf{n}_1^\top \mathbf{R}\boldsymbol{\mathcal{P}}_1 - d_1) \quad (\mathbf{n}_2^\top \mathbf{R}\boldsymbol{\mathcal{P}}_2 - d_2) \quad \dots \quad (\mathbf{n}_n^\top \mathbf{R}\boldsymbol{\mathcal{P}}_n - d_n)]^\top$$

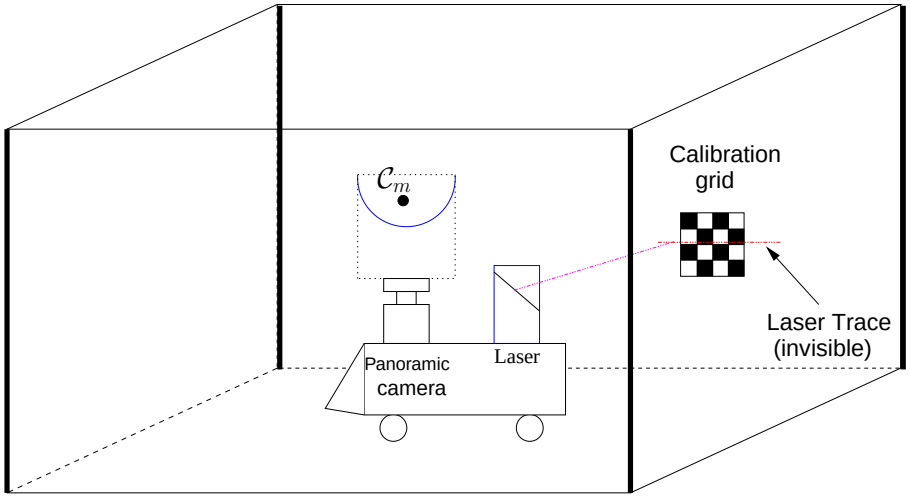


Figure 4.13: Association between laser points and 3D planes

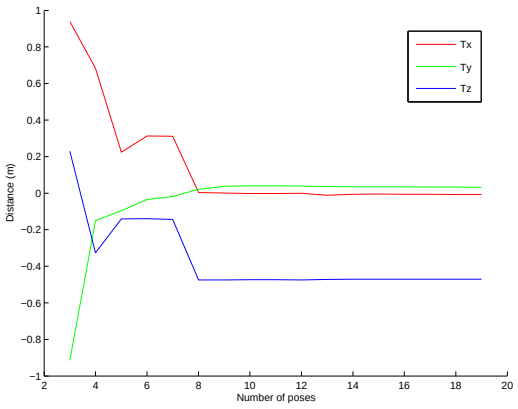


Figure 4.14: Estimation of the translation

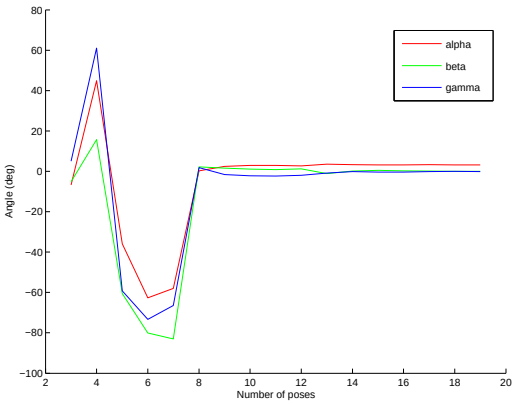


Figure 4.15: Estimation of the rotation

4.3.2.1 Validation

To validate the approach, 19 associations were made between planes and the laser trace. As we can see from figures 4.14 and 4.15, the estimates stabilise after 8 associations and the subsequent information does not have a strong influence.

The final result was $\mathbf{t} = [-0.0074, 0.032, -0.47]$ and $\mathbf{R} = [3.15, 0.048, -0.099]$ which is coherent with previous results.

Figure 4.16 shows some of the associations between planes and laser points.

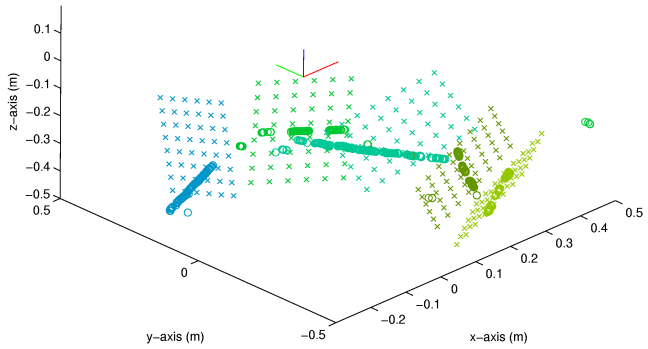


Figure 4.16: 3D view of the calibration planes and reprojected laser points

4.3.2.2 Autocalibration from planes

We will see in Chapter 6 that we can estimate the position of 3D planes by tracking them in the image. An obvious question is then: is it possible to autocalibrate the sensor through visual tracking?

If we are tracking a plane in general configuration, the answer is yes. However in the case of a laser range finder with a laser plane perpendicular to the optical axis of the omnidirectional sensor as for the Anis robot (figure 4.1), the vertical displacement is unobservable.

4.4 Conclusion

Several methods were proposed to find the relative position between a laser range finder and an omnidirectional camera.

When the laser is visible and that we can obtain measurements at different angles, the calibration can be automatised easily.

With invisible laser range finders, the situation is more difficult as we need 3D information. By obtaining the equation of planes (for example calibration grids), the calibration is possible and gives coherent results.

Part II

Real-time structure from motion

Chapter 5

Minimal parameterisation through Lie algebras

Contents

Iterative minimisation is an important step in many computer vision and robotic problems (eg. bundle adjustment, pose estimation with a geometric distance, calibration [Hartley and Zisserman, 2000], etc.). In this thesis, it appears naturally in visual tracking or structure and motion from lines.

Over-parameterisation, in other words using more elements than necessary, can have a detrimental effect on the convergence speed, region of convergence and quality of the results. A typical example are rotations. Parameterising a rotation by the 9 elements of the rotation matrix would be a poor choice. In presence of noise, after minimisation, we would no longer have a matrix representing a rotation. Projecting the matrix on the space of rotation matrices (using for example the Frobenius norm) could lead to a very bad approximation of the optimal solution. Other examples would be Plücker coordinates (represented by 6 parameters but with only 4 degrees of freedom) or planar homography matrices (represented by 9 parameters with only 8 degrees of freedom). Often there is no *global* minimal parameterisation but this is not a problem as we only want to parameterise the increment when minimising, which is a *local* parameterisation. We may note that in certain cases minimal parameterisation can lead to a more complex cost function with more local minima. The choice should be made according to the problem. In computer vision, it is common practice to remove constraints to obtain a linear (fast) set of equations that can be easily solved and help initialise a more precise non-linear minimisation approach [Hartley and Zisserman, 2000].

In this chapter, we will present how to parameterise motion but also the special linear group (later shown to be related to planar homographies) with the minimal amount of parameters through Lie algebras. We will also concentrate on explicit formulas that are of importance for real-time applications. This is of course a very restrictive use of Lie algebras. However it is close to the historical use where these algebras were considered as tools to represent Lie groups. For a more general study of Lie algebras, you can refer to [Varadarajan, 1974]. This chapter is partly based on [Hall, 2000; Gallier, 2001; Ma et al., 2003; Smith, 2001]. The proofs of general Lie group and algebra properties will not be given and can be found in [Hall, 2000]. A convincing approach of constraining a minimisation to a manifold can be found in [Lee and Moore, 2004]. Minimal approaches are also related to the orthonormal representation used by Bartoli and Sturm [Bartoli and Sturm, 2004].

5.1 A short introduction to Lie groups and Lie algebras

The use of Lie algebras in visual tracking and servoing has been popularised by the work of Drummond and Cipolla [Drummond and Cipolla, 1999, 2000]. The motivations behind using this representation is minimal local parameterisation. We will also discuss a property that will have important consequences for developing fast visual tracking as we will see in Chapter 6.

5.1.1 Matrix exponential and logarithm

Definition 1 *Given an $n \times n$ (real or complex) matrix $\mathbf{A}$, the exponential of $\mathbf{A}$ noted $e^{\mathbf{A}}$ is defined as:*

$$e^{\mathbf{A}} = \mathbf{I}_n + \sum_{p \geq 1} \frac{\mathbf{A}^p}{p!} = \sum_{p \geq 0} \frac{\mathbf{A}^p}{p!}$$

This series is absolutely convergent and thus well-defined.

For the logarithm, extra constraints must be imposed on $\mathbf{A}$.

Definition 2 *Under the condition* $\|\mathbf{A}-\mathbf{I}\|<1$*, the logarithm of* $\mathbf{A}$ *is defined as:*

$$\log \mathbf{A} = \sum_{p\geq 0}(-1)^{p+1}\frac{(\mathbf{A}-\mathbf{I})^p}{p}$$

We may also link the exponential to the logarithm with the following proposition.

Property 1

$$\|\mathbf{A}-\mathbf{I}\|<1 \Longrightarrow e^{\log \mathbf{A}} = \mathbf{A}$$
$$\|\mathbf{A}\|<\log 2 \Longrightarrow \left(\|e^{\mathbf{A}}-1\|<1\right)\wedge\left(\log e^{\mathbf{A}} = \mathbf{A}\right)$$

The following properties can also be useful.

Property 2 *For all* $\mathbf{X}$ *and* $\mathbf{Y}$ $n\times n$ *matrices:*

- $e^{\mathbf{0}} = \mathbf{I}$
- $(e^{\mathbf{X}})^{-1} = e^{-\mathbf{X}}$
- $\det(e^{\mathbf{X}}) = e^{trace(\mathbf{X})}$
- $\forall(\alpha,\beta)\in\mathbb{R}^2, e^{(\alpha+\beta)\mathbf{X}} = e^{\alpha\mathbf{X}}e^{\beta\mathbf{X}}$
- $\mathbf{XY}=\mathbf{YX}\Longrightarrow e^{\mathbf{X}+\mathbf{Y}} = e^{\mathbf{X}}e^{\mathbf{Y}} = e^{\mathbf{Y}}e^{\mathbf{X}}$
- $\mathbf{P}\in\mathbb{GL}(n,\mathbb{R})\Longrightarrow e^{\mathbf{PXP}^{-1}} = \mathbf{P}e^{\mathbf{X}}\mathbf{P}^{-1}$
- $\|e^{\mathbf{X}}\|\leq e^{\|\mathbf{X}\|}$

An essential property for defining Lie algebras of matrix Lie groups is the smoothness of $t\rightarrow e^{t\mathbf{X}}$.

Property 3 *Let* $\mathbf{X}$ *be a* $n\times n$ *real (or complex) matrix,* $t\rightarrow e^{t\mathbf{X}}$ *is a smooth curve in the space of real (or complex)* $n\times n$ *matrices. Furthermore:*

$$\frac{d}{dt}e^{t\mathbf{X}} = \mathbf{X}e^{t\mathbf{X}} = e^{t\mathbf{X}}\mathbf{X}$$

In particular,

$$\left.\frac{d}{dt}e^{t\mathbf{X}}\right|_{t=0} = \mathbf{X}$$

5.1.2 Lie algebras of matrix Lie groups

A Lie group is a group that locally has the topology of $\mathbb{R}^n$ everywhere (i.e. it is a smooth manifold). All closed subgroups of the general linear group (group of all invertible $n\times n$ matrices under multiplication) $\mathbb{GL}(n,\mathbb{C})$ are Lie groups[1]. Projective transformation groups and their subgroups that are of interest to us for this study are thus matrix Lie groups.

Definition 3 *A finite-dimensional real or complex Lie algebra is a finite-dimensional real or complex vector space* $\mathfrak{g}$*, together with a binary operation* $[\ ]$*, called Lie bracket, which satisfies the following properties:*

- $[\ ]$ *is bilinear,*
- $\forall(X,Y)\in\mathfrak{g}^2, [X,Y] = -[Y,X]$

[1] this result is due to Cartan and Von Neumann

- $\forall (X, Y, Z) \in \mathfrak{g}^3, [X, [Y, Z]] + [Y, [Z, X]] + [Z, [X, Y]] = 0$ *(Jacobi identity)*

A Lie algebra is an algebra in the usual sense with the product replaced by the Lie bracket that is neither commutative nor associative. The Jacobi identity can be seen as a substitute for associativity.

Definition 4 *Let G be a matrix Lie group. We will call Lie algebra of G, denoted $\mathfrak{g}$, the set of all matrices $\mathbf{X}$ such that $e^{t\mathbf{X}}$ is in G for all real numbers t. The Lie bracket is defined as: $[X, Y] = XY - YX$.*

We can check that the Lie algebra associated in this way to a Lie group verifies the properties.

5.1.3 Exponential mapping

If G is a matrix Lie group with Lie algebra $\mathfrak{g}$, then the exponential mapping for G is the map:

$$\exp : \mathfrak{g} \to G$$

In general the mapping is neither one-to-one nor onto but provides the link between the group and the Lie algebra.

We also have the following result that says that mapping is *locally* bijective. We will see later on that this is important for minimisation.

Theorem 1 *Let G be a matrix Lie group with Lie algebra $\mathfrak{g}$. There exists a neighborhood v about zero in $\mathfrak{g}$ and a neighborhood V of $\mathbf{I}$ in G such that $\exp : v \to V$ is smooth and one-to-one onto with smooth inverse.*

5.1.4 Lie algebra generators

Definition 5 *A matrix Lie group G is said to be **path-connected** if given any two matrices $\mathbf{A}$ and $\mathbf{B}$ in G, there exists a continuous path $\mathbf{A}(t)$, $a \leq t \leq b$, lying in G with $\mathbf{A}(a) = \mathbf{A}$ and $\mathbf{A}(b) = \mathbf{B}$.*

There is an equivalence between **path-connectedness** and **connectedness** for matrix Lie groups.

We will now state two important properties for minimisation.

Property 4 *The Lie groups $\mathbb{SO}(n, \mathbb{R})$, $\mathbb{SL}(n, \mathbb{R})$ and $\mathbb{SE}(n, \mathbb{R})$ are connected.*

Property 5 *If G is a matrix Lie group, then the component of G containing the identity is a subgroup of G.*

These properties indicate that if we start from the identity, we can find a continuous path to any value of the group. This is is exactly what we want to do in the case of minimisation. However we will now see that an important property is missing to these groups.

Definition 6 *A connected matrix Lie group G is said to be **simply connected** if every loop in G can be shrunk continuously to a point in G.*

We can show that a simply connected Lie group G has a natural one-to-one correspondence between the representations of G and the representations of its Lie algebra. $\mathbb{SO}(n,\mathbb{R})$, $\mathbb{SL}(n,\mathbb{R})$ and $\mathbb{SE}(n,\mathbb{R})$ are not simply connected. We will see in more details why. In particular, in the case of $\mathbb{SL}(3,\mathbb{R})$, we will see that certain elements of the group cannot be represented by elements of the Lie algebra.

In the previous section we defined the Lie algebra of a matrix Lie group as the set of all matrices $\mathbf{X}$ such that $e^{t\mathbf{X}}$. An element of $g \in G$ can be expressed in terms of n independent elements of G with n the dimension of the group ($n = \dim(G)$).

If g only depends on a parameter t_i:

$$g(t_i) = \exp(t_i \mathbf{A}_i)$$

and by differentiation:

$$\mathbf{A}_i = \left.\frac{\partial g(t_i)}{\partial t_i}\right|_{t_i=0}$$

$\mathbf{A}_i$ is referred to as a generator of the Lie algebra which is independent of t. It is named generator because with the property of connectedness it can generate a path to any the matrix of the form of g starting from the identity. If we repeat the operation for each independent element of the group, we obtain the set of all generators of the Lie algebra. We may note that two generators do not necessarily commute (i.e. $\mathbf{A}_i\mathbf{A}_j \neq \mathbf{A}_j\mathbf{A}_i$ if $i \neq j$) and the Lie bracket can be seen as the amount of non-commutativity of the Lie group. By closure of the Lie group by the Lie bracket:

$$\forall (i,j) \in [1..n]^2, \;\; [\mathbf{A}_i, \mathbf{A}_j] = \sum_k c_{ij}^k \mathbf{A}_k$$

c_{ij} are the structure constants of the Lie algebra. Intuitively, we get the feeling that the bracket can be used to obtain missing generators. (This is indeed the case and is part of an important result of control theory.)

With $\{\mathbf{A}_i | i \in [1..n]\}$ the set of generators of G, any element $\mathbf{A}(\mathbf{x})$ of $\mathfrak{g}$ can be written as a linear combination of the basis:

$$\mathbf{A}(\mathbf{x}) = \sum_{i=1}^{n} x_i \mathbf{A}_i$$

with $\mathbf{x} = (x_1, ..., x_n)$.

There is a practical advantage of using a Lie algebra parameterisation: by cancelling values corresponding to generators, we can obtain subgroups of the initial Lie group. For example, in the following sections, the parameterisation of $\mathbb{SE}(3)$ can be used to represent planar motion by selecting the appropriate generators.

We will now prove an important property to establish the second order minimisation (ESM) algorithm.

Proposition 1 *Let n be the dimension of the matrix Lie group G, let ($\mathbf{A}_1$,... $\mathbf{A}_n$) be the set of generators of the associated Lie algebra and $m \times m$ the size of the matrices representing the elements of G. Let $\mathbf{A}(\mathbf{x})$ be an $n \times n$ real matrix belonging to the algebra and seen as a function of $\mathbf{x} \in \mathbb{R}^n$:*

$$\forall \mathbf{x}_0 \in \mathbb{R}^n, \left.\frac{d(e^{-\mathbf{A}(\mathbf{x}_0)}e^{\mathbf{A}(\mathbf{x})})}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0} \mathbf{x}_0 = \left.\frac{de^{\mathbf{A}(\mathbf{x})}}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{0}} \mathbf{x}_0 = [\text{flat}(\mathbf{A}_1)^\top \; \text{flat}(\mathbf{A}_2)^\top \cdots \text{flat}(\mathbf{A}_n)^\top]_{m^2 \times n} \mathbf{x}_0$$

with: $\text{flat}(\mathbf{A}) = [a_{11} \;\; a_{12} \;\; \cdots \;\; a_{1m} \;\; a_{21} \;\; a_{22} \;\; \cdots a_{mm}]$

Proof: (An elegant proof due to Pascal Morin, another possibility is proposed in the thesis of Selim Benhimane)

The proof is valid for any Lie group.

Let g be a function of $\mathbb{R}^n$ with the property $g((1+t)\mathbf{x}_0) = g(\mathbf{x}_0)g(t\mathbf{x}_0)$ (property of the subgroup defined by $\mathbf{x}_0$),

In the statement, $g(t\mathbf{x}) = e^{t\mathbf{A}(\mathbf{x})}$.

We can now differentiate the two expressions:

$$\begin{aligned} \left.\frac{dg((1+t)\mathbf{x}_0)}{dt}\right|_{t=0} &= \left.\frac{dg(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0} \mathbf{x}_0 \\ \left.\frac{dg(\mathbf{x}_0)g(t\mathbf{x}_0)}{dt}\right|_{t=0} &= g(\mathbf{x}_0) \left.\frac{dg(\mathbf{x})}{d\mathbf{x}}\right|_{\mathbf{x}=0} \mathbf{x}_0 \end{aligned} \tag{5.1}$$

With the group property $g(\mathbf{x}_0)^{-1} = g(-\mathbf{x}_0)$, the first equality is obtained. The value on the right is then a simple matter of differentiation.

□

5.1.5 Application to iterative optimisation

With G a matrix Lie group of dimension n, let:

$$\begin{array}{rccc} f: & G & \longrightarrow & \mathbb{R} \\ & \mathbf{g} & \longmapsto & f(\mathbf{g}) \end{array}$$

Consider the following minimisation problem, with d a differentiable distance (typically L_2 norm) and $\overline{\mathbf{f}} \in \mathbb{R}$:

$$\overline{\mathbf{g}} = \min_{\mathbf{g}} d(f(\mathbf{g}), \overline{\mathbf{f}})$$

If f is non-linear, the problem is often difficult to solve and we might use an iterative gradient descent method. We start from an initial value $\widehat{\mathbf{g}}$ and at each step add a value $\mathbf{g}_k$ calculated typically from the Jacobian: $\widehat{\mathbf{g}} \leftarrow \widehat{\mathbf{g}} + \mathbf{g}_k$. The problem of such a method is that we do not guarantee that the new value of $\widehat{\mathbf{g}}$ will belong to the group G. To solve this problem, the new $\widehat{\mathbf{g}}$ is often projected onto the group manifold but this can alter the convergence speed and the region of convergence.

Alternatively, we can define a new function h. With $\mathfrak{g}$ the Lie algebra of G and $+$ the group operation:

$$\begin{array}{rccccc} h: & \mathbb{R}^n & \longrightarrow & \mathfrak{g} & \longrightarrow & \mathbb{R} \\ & \mathbf{x} & \longmapsto & G(\mathbf{x}) & \longmapsto & f(\widehat{\mathbf{g}} + e^{G(\mathbf{x})}) \end{array}$$

h is only defined *locally* by the Lie algebra parameterisation of G. If we apply a gradient descent approach to h, we will start at $\mathbf{x} = 0$ (that corresponds to the initial value of f) and the update will be written: $\widehat{\mathbf{g}} \leftarrow \widehat{\mathbf{g}} + e^{G(\mathbf{x}_k)}$. We are know sure that at each step, the new value of $\widehat{\mathbf{g}}$ belongs to the Lie group G.

To be able to link $\widehat{\mathbf{g}}$ to $\overline{\mathbf{g}}$ by infinitesimal transformation, we must make sure that the values are path-connected (i.e. belong to the same component). If this is not the case, we should apply a descent method in each component of the group. $\mathbb{O}(3,\mathbb{R}) = \{\mathbf{R} \in \mathbb{GL}(3) \mid \mathbf{R}\mathbf{R}^\top = \mathbf{I}\}$, the group of orthogonal matrices is an example of Lie group that is not connected. It has two components $\mathbb{SO}(3,\mathbb{R}) = \{\mathbf{R} \in \mathbb{O}(3,\mathbb{R}) \mid \det(\mathbf{R}) = +1\}$, the group of rotations and $\mathbb{O}^-(3,\mathbb{R}) = \{\mathbf{R} \in \mathbb{O}(3,\mathbb{R}) \mid \det(\mathbf{R}) = -1\}$. If

we initialise a minimisation with $\widehat{\mathbf{g}} = \mathbf{I}$ for example, we will only be able to connect to the values of $\mathbb{SO}(3,\mathbb{R})$.

In the following sections, we will only consider real valued matrices.

5.2 Representing rotations

The special orthogonal subgroup of dimension 3, also called rotation group, is defined as:

$$\mathbb{SO}(3) = \{\mathbf{R} \in \mathbb{GL}(3) \mid \mathbf{R}\mathbf{R}^\top = \mathbf{I}, \det(\mathbf{R}) = +1\}$$

The elements of the group can be obtained from the three transformation matrices (represented by the Euler angles) and parameterised by α:

$$\mathbf{M}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix}, \mathbf{M}_y(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}, \mathbf{M}_z(\alpha) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Let the (3×1) vectors $\mathbf{b}_x = (1,0,0)$, $\mathbf{b}_y = (0,1,0)$ and $\mathbf{b}_z = (0,0,1)$ be the canonical orthonormal basis of $\mathbb{R}^3$. By differentiation, we obtain the three generators of the Lie algebra $\mathfrak{so}(3)$:

$$\mathbf{A}_1 = [b_x]_\times, \ \mathbf{A}_2 = [b_y]_\times, \ \mathbf{A}_3 = [b_z]_\times$$

The exponential map $\exp : \mathfrak{so}(3) \to \mathbb{SO}(3)$ has an explicit form known as Rodrigues' formula (1840).

Theorem 2 *(Rodrigues' formula) Let* $\mathbf{A}(\mathbf{x}) \in \mathfrak{so}(3)$ *and* $\theta = \|\mathbf{x}\|$*:*

$$\begin{cases} e^{\mathbf{A}} = \mathbf{I}_3 + \frac{\sin\theta}{\theta}\mathbf{A} + \frac{1-\cos\theta}{\theta^2}\mathbf{A}^2 & \text{if } \theta \neq 0 \\ e^{\mathbf{A}} = \mathbf{I}_3 & \text{otherwise} \end{cases}$$

The formula has also a geometrical interpretation as the rotation of an angle θ around the axis $\mathbf{x}$. Can any rotation be written in exponential form ? The answer is yes and we can obtain an explicit formulation of the logarithm.

Theorem 3 *The exponential map is surjective onto* $\mathbb{SO}(3)$ *and the logarithm of a rotation matrix* $\mathbf{R}$ *given by:*

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

can be expressed as $e^{\mathbf{A}(\mathbf{x})}$*:*

$$\begin{cases} \mathbf{x} = \frac{\theta}{2\sin\theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} & \text{if } \theta \neq 0, \pi \\ \mathbf{x} = \theta \begin{bmatrix} \sqrt{\frac{r_{11}+1}{2}} \\ \sqrt{\frac{r_{22}+1}{2}} \\ \sqrt{\frac{r_{33}+1}{2}} \end{bmatrix} & \text{otherwise} \end{cases}$$

with $\theta = \cos^{-1}\left(\frac{\text{trace}(\mathbf{R})-1}{2}\right)$.

The proof can be obtain by identifying the exponential map given by Rodrigues' formula with the rotation matrix. Care has to be taken when $\theta = \pi$ (the formula given in [Ma et al., 2003] is incomplete). A complete proof for the surjectivity of exponential maps onto $\mathbb{SO}(n)$ can be found in [Gallier, 2001]. The exponential map onto $\mathbb{SO}(3)$ is not bijective as any value $\pm 2\pi\mathbf{x}$ will represent the same rotation.

5.3 Representing 3D motion

The previous section described pure rotations, we will now study the general motion of a rigid body. Assume a reference frame $\boldsymbol{\mathcal{F}}^*$, fixed in space and a frame $\boldsymbol{\mathcal{F}}$ fixed to a rigid body. We can represent the motion from $\boldsymbol{\mathcal{F}}^*$ to $\boldsymbol{\mathcal{F}}$ with elements of the special euclidean group $\mathbb{SE}(3)$ if we represent the position of the body by homogeneous coordinates.

$$\mathbb{SE}(3) = \{ \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{GL}(4) \mid \mathbf{R} \in \mathbb{SO}(3), \mathbf{t} \in \mathbb{R}^3 \}$$

By differentiation, we obtain the generators of the translation $(\mathbf{A}_1, ..., \mathbf{A}_3)$ and rotation $(\mathbf{A}_4, ..., \mathbf{A}_6)$ of the Lie algebra $\mathfrak{se}(3)$:

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{b}_x \\ \mathbf{0} & 0 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{b}_y \\ \mathbf{0} & 0 \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} \mathbf{0} & \mathbf{b}_z \\ \mathbf{0} & 0 \end{bmatrix}, \mathbf{A}_4 = \begin{bmatrix} [\mathbf{b}_x]_\times & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}, \mathbf{A}_5 = \begin{bmatrix} [\mathbf{b}_y]_\times & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}, \mathbf{A}_6 = \begin{bmatrix} [\mathbf{b}_z]_\times & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$$

Using Rodrigues' formula, we can also obtain an explicit formulation of the exponential map.

Theorem 4 *Let* $\mathbf{A}(\mathbf{x}) \in \mathfrak{se}(3)$*, with* $\mathbf{v} = (x_1, x_2, x_3)$ *(translational component) and* $\omega = (x_4, x_5, x_6)$ *(rotational component) :*

$$\begin{cases} e^{\mathbf{A}} = \begin{bmatrix} e^{[\omega]_\times} & \frac{\left((\mathbf{I} - e^{[\omega]_\times})[\omega]_\times + \omega\omega^\top\right)\mathbf{v}}{\|\omega\|^2} \\ \mathbf{0} & 1 \end{bmatrix} & \text{if } \|\omega\| \neq 0 \\ e^{\mathbf{A}} = \begin{bmatrix} \mathbf{I} & \mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix} & \text{otherwise} \end{cases}$$

The exponential map is surjective onto $\mathbb{SE}(n)$ [Gallier, 2001]. We can now express the logarithm for $\mathbb{SE}(3)$ explicitly using Theorem 5.

Theorem 5 *The exponential map is surjective onto* $\mathbb{SE}(3)$ *and the logarithm of a motion matrix* $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$ *is given by:*

$$\omega = \log \mathbf{R} \quad \text{(logarithm of } \mathbb{SO}(3))$$

and

$$\begin{cases} \mathbf{v} = \left(\frac{(\mathbf{I} - e^{[\omega]_\times})[\omega]_\times + \omega\omega^\top}{\|\omega\|^2} \right)^{-1} \mathbf{t} & \text{if } \|\omega\| \neq 0 \\ \mathbf{v} = \mathbf{t} & \text{otherwise} \end{cases}$$

5.4 Special linear group

The special linear subgroup of dimension 3, is defined as:

$$\mathbb{SL}(3) = \{\mathbf{H} \in \mathbb{GL}(3) \mid \det(\mathbf{H}) = +1\}$$

This group is of dimension 8. Furthermore, $\det(e^A) = e^{\text{trace}(A)} = 1$ implies that the generators have zero trace. We can thus choose the 8 following independent generators of $\mathfrak{sl}(3)$:

$$\begin{array}{llll} \mathbf{A}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \mathbf{A}_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \mathbf{A}_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \mathbf{A}_7 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\ \mathbf{A}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, & \mathbf{A}_4 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & \mathbf{A}_6 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \mathbf{A}_8 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{array} \tag{5.2}$$

Theorem 6 *The exponential map is not surjective onto* $\mathbb{SL}(3)$.

An example of a matrix of $\mathbb{SL}(3)$ than cannot be expressed by elements in the Lie algebra can be found in [Gallier, 2001]. However $\mathbb{SL}(3)$ is path-connected and will be used to represent planar homographies in Chapter 6 on visual tracking.

5.5 Conclusion

In this chapter, we have presented a mathematical method of constraining an iterative optimisation process such that the set of parameters are coherent with the object represented. Details were given for the important cases of motion and special linear group representation. $\mathbb{SL}(3)$ will appear in the problem of homography-based tracking.

Chapter 6

Visual tracking

Contents

The aim of this chapter is to illustrate efficient ways of using visual information for motion estimation. The results have implications in visual servoing and SLAM.

We will start by giving an overview of tracking approaches to situate our work on omnidirectional plane-based visual tracking. We will then concentrate more specifically on sum-of-squared differences (SSD) tracking. We will explain why SSD tracking is particularly well suited for robotic tasks.

6.1 An overview of tracking approaches in the literature

In this overview, we will only consider tracking without artificial markers or beacons in the environment (i.e. only image information). Visual tracking approaches can be classified according to several properties:

- 3D model-based/3D model-free tracking. Either we assume that the object tracked can be parameterised by a certain surface or structure (planes, quadrics, cubes, set of planes, ...), or we characterise it only by its properties (color, texture, ...).
- matching-based/direct tracking. Either we extract features in the image and then look for similar features in the image without using prior knowledge on the camera motion or the approach assumes small displacements between frames and processes the image information directly.
- 2D/3D tracking. We can either track an object in the image or estimate the 3D motion of the object in the scene from its image.
- type of object tracked: deformable or rigid.

It is not always obvious how to characterise an approach but these classes still give a general description of the method used. We could also add that some systems need a learning step (for example to improve robustness to illumination or occlusion) whereas others use robust techniques to achieve this goal. An overview of monocular model-based tracking can be found in [Lepetit and Fua, 2005]. Three branches of 3D model-free tracking will now be discussed: matching, recursive image-based and recursive 3D approaches.

6.1.1 Tracking by matching

The steps for tracking between two images by matching are:

1. extract features in the first image,
2. extract features in the second image,
3. associate the features in the two images through a distance measure. The efficiency and robustness of the data association can be enforced using a prior knowledge of the motion of the camera or of the object tracked.

The most obvious example of tracking with data association is the one commonly used in projective geometry with Harris points associated through correlation [Hartley and Zisserman, 2000].

Why don't we search for the points extracted in the first image directly in the second image? The answer is the gain we expect to obtain in terms of computation and robustness. In the example using Harris points, we only need to correlate between features instead of search in the whole image. Furthermore there is a smaller chance of having bad associations since we only correlate between

features that we expect to be able to associate. Of course, if there are a lot of features and that the distance measure is not very discriminate, we can expect to have many outliers. Matching in the whole image also has the disadvantage of being computationally expensive but the advantage of making it possible to match over big distances.

Outlier rejection is an essential process in these type of methods. For example, if we are estimating the motion of the camera from Harris points, we might use the epipolar constraint (through the essential matrix or fundamental matrix) to remove outliers. The constraint is imposed by the properties of projective geometry. It is also possible to add constraints such as planarity (planar homography matrix) or even search for specific objects.

The outlier rejection process and the salient feature extraction can make it difficult to obtain frame-rate tracking. RANSAC [Fischler and Bolles, 1981] is often used but is time-consuming. Improvements have been made to the standard RANSAC approach [Matas and Chum, 2002] and modern computers make it possible to obtain high frame rate computation even with these processing steps. Furthermore, SIFT points [Lowe, 2004] (or learning-based techniques [Lepetit et al., 2005]) produce fewer outliers, the downside being a higher computational burden compared to Harris points. However fewer outliers can remove altogether the need for RANSAC. Faster robust estimators (Tukey, Huber) are then sufficient. An advantage of these approaches comes from the robustness to occlusion and change in intensity but also the possibility to track objects with large displacements in the image.

These methods have been applied successfully for tracking planes [Simon et al., 2000], head pose [Tordoff et al., 2002] or deformable objects [Pilet et al., 2005]. Matching is also popular for augmented reality [Chia et al., 2002]. Lines can be an alternative to using points even though they are harder to match reliably between images.

6.1.2 Recursive tracking

The steps for tracking recursively between two images are:

1. extract features in the first image,
2. project the features onto the second image,
3. apply a distance measure between the information in the second image and the features from the first image and find a direction that will minimise the error.

For this approach to work, the assumption is made that the motion of the features between the subsequent frames is small. The objects are also often considered as lambertian. This assumption is reasonable if the camera frame rate is high compared to the object motion and that the computation of step (3) holds within the frame rate.

Compared to tracking by matching, we no longer need to associate the features as this step is included in the minimisation process. The approach is however generally more sensitive to occlusion and less well adapted to strong motion.

One of the first such techniques was edge-based tracking. An edge is extracted in an image and reprojected in the following image. The distance to minimise is then typically obtained by searching edge points along the normals to the initial edge. The advantage of edge-tracking is its robustness to changes in intensity but it is sensitive to occlusion. The minimisation can either be image-based or 3D (we minimise the 3D pose of the object according to the projection of its edges in the image).

Sum-of-squared differences (SSD) tracking can be traced back to the work by Lucas, Kanade and later Shi and Tomasi [Lucas and Kanade, 1981; Shi and Tomasi, 1994] (KLT tracker). SSD measures

the difference in intensity between a portion of the first image reprojected onto the second image. The minimisation (based on the image gradient) can be imaged-based (2D) for example searching for the translation (t_x, t_y) that gives the smallest reprojection error. It can also be 3D or model-based by reprojecting a 3D object and minimising the difference in the image over the position (6 degrees of freedom: rotation and translation). The advantage of this approach is precision (all the information is being used) and speed. This is why these techniques are particularly well adapted to robotic tasks such as motion estimation and visual servoing. Compared to matching approaches, SSD tracking is generally faster and more precise. The downside is the need for a strong overlap between the reprojected and the real object for the system to converge.

A closely related active field of research is model-free image-based tracking. Current algorithms enable to track complex deformable objects with strong changes in intensity and with clutter and occlusion. The idea is to loosen the constraint of same intensity imposed by the SSD and use a descriptor of the template to track. Histograms are often chosen as they are fast to compute and partly invariant to change in shape and clutter. In [Comaniciu et al., 2003], the authors use gradient descent on a histogram-based distance for the tracking. Deterministic tracking however is not very robust when the background is similar to the object tracked or if the object disappears. To improve the tracking, particle filtering approaches can be used [Isard and Blake, 1998; Pérez et al., 2002] and to deal with background clutter on-line or off-line learning is becoming popular [Collins et al., 2005]. These techniques are well adapted to image-based tracking of complex objects but cannot - as such - be used for motion estimation.

In this chapter, we will describe plane-based 3D SSD visual tracking using an omnidirectional sensor. Three contributions will be presented:

a) an extension to omnidirectional vision of the work from Benhimane and Malis [Benhimane and Malis, 2004] on Efficient Second-order Minimisation (ESM) for perspective homography-based tracking,

b) a new approach for tracking multiple planes valid for all single viewpoint sensors,

c) an analysis of how to implement the ESM algorithm efficiently but also some new variants (αESM, iESM) to the standard algorithm with better computational complexities.

The apparent difficulty of tracking with panoramic devices comes from the non-linear projection model resulting in changes of shape in the image that makes the direct use of methods such as KLT nearly impossible. Parametric models [Hager and Belhumeur, 1998; Shum and Szeliski, 2000; Baker and Matthews, 2001] such as the homography-based approach presented in this thesis are well adapted to this problem. Previous related work using homography-based tracking for perspective cameras include [Benhimane and Malis, 2004] and [Buenaposada and Baumela, 2002] which extend the work proposed by Hager [Hager and Belhumeur, 1998]. Homographies have also been used for visual servoing with central catadioptric cameras [Hadj-Abdelkader et al., 2005] and share with our approach the notion of homographies for points belonging to the sphere of the unified projection model. The single viewpoint property means it would be possible to track in an unwarped perspective view. This is however undesirable for the following reasons:

1. it introduces a discontinuity in the Jacobian (at least two planes are needed to represent the 360° field of view),

2. the non-uniform resolution is not taken into account and

3. the approach is inefficient (in terms of speed and memory usage).

To our knowledge, this is the only work on SSD tracking for omnidirectional sensors. The closest work is that of Barreto *et al* [Barreto et al., 2002]. The authors propose a method for tracking omnidirectional lines using a contour-to-point tracker to avoid the problem of quadric-based catadioptric line fitting.

6.2 Efficient second order minimisation

SSD tracking generally relies on iterative methods to find the optimal position and parameters. First-order methods (often called *forward compositional* in the visual tracking literature) or methods with final super-linear convergence (eg. Levenberg-Marquardt or Dog Leg) are generally employed as calculating the Hessian to obtain full quadratic convergence is computationally expensive.

In fact, through the Lie algebra parameterisation, we can obtain second-order convergence with a computational cost of the same order as a first order method, this technique was dubbed efficient second order minimisation (ESM) [Malis, 2004; Benhimane and Malis, 2004]. We will now derive the equations. It will become clear in the following sections how they appear and the role of the Lie algebra parameterisation.

Consider the general least-squares minimisation problem:

$$F(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n}(f_i(\mathbf{x}))^2 = \frac{1}{2}\|\mathbf{f}(\mathbf{x})\|^2 \tag{6.1}$$

The necessary condition for finding a local or the global minimum of the cost function is that there exists a stationary point $\widetilde{\mathbf{x}}$ such that:

$$[\nabla_{\mathbf{x}} F]_{\mathbf{x}=\widetilde{\mathbf{x}}} = \mathbf{0} \tag{6.2}$$

where $\nabla_{\mathbf{x}}$ is the gradient operator with respect to the parameter $\mathbf{x}$. When equation (6.2) is non-linear, a closed-form solution is generally difficult to obtain.

A second-order Taylor series of $\mathbf{f}$ about $\mathbf{x} = \mathbf{0}$ gives:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\,\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0},\mathbf{x})\mathbf{x} + \mathcal{R}(\|\mathbf{x}\|^3) \tag{6.3}$$

where $\mathbf{J}(\mathbf{0}) = [\nabla_{\mathbf{x}}\mathbf{f}]_{\mathbf{x}=\mathbf{0}}$, $\mathbf{M}(\mathbf{z},\mathbf{x}) = [\nabla_{\mathbf{x}}\mathbf{J}]_{\mathbf{x}=\mathbf{z}}\mathbf{x}$ and $\mathcal{R}(\|\mathbf{x}\|^3)$ is the third-order reminder. Similarly, we can write the Taylor series of the Jacobian $\mathbf{J}$ about $\mathbf{x} = \mathbf{0}$:

$$\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{0}) + \mathbf{M}(\mathbf{0},\mathbf{x}) + \mathcal{R}(\|\mathbf{x}\|^2) \tag{6.4}$$

Plugging (6.4) in (6.3) leads to:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{0}) + \frac{1}{2}(\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}))\,\mathbf{x} + \mathcal{R}(\|\mathbf{x}\|^3) \tag{6.5}$$

As $\widetilde{\mathbf{x}} \approx \mathbf{0}$, a second order approximation of $\mathbf{f}$ in $\widetilde{\mathbf{x}}$ is:

$$\mathbf{f}(\widetilde{\mathbf{x}}) \approx \mathbf{f}(\mathbf{0}) + \frac{1}{2}(\mathbf{J}(\mathbf{0}) + \mathbf{J}(\widetilde{\mathbf{x}}))\,\widetilde{\mathbf{x}} \tag{6.6}$$

Under certain conditions, that we will detail in the following sections, $\mathbf{J}(\widetilde{\mathbf{x}})\widetilde{\mathbf{x}}$ can be calculated without knowing the value of $\widetilde{\mathbf{x}}$. This is the basis of the ESM algorithm.

Let $\mathbf{J}(\widetilde{\mathbf{x}})\widetilde{\mathbf{x}} = \mathbf{J}'\widetilde{\mathbf{x}}$, at the solution, $\mathbf{f}(\widetilde{\mathbf{x}}) = 0$, so our second-order least-square minimiser is the solution to:

$$\widetilde{\mathbf{x}} = -\left(\frac{\mathbf{J}(\mathbf{0}) + \mathbf{J}'}{2}\right)^{+}\mathbf{f}(\mathbf{0})$$

6.3 Homography-based tracking for single viewpoint sensors

6.3.1 Incremental tracking

Figure 6.1 illustrates the underlying principal for tracking incrementally. For each new incoming image $\mathcal{I}_i$, we look for the optimal increment from the previous position corresponding to $\mathcal{I}_{i-1}$. After minimisation, we obtain an estimate of the optimal transformation between the *reference* image $\mathcal{I}^*$ and the last image $\mathcal{I}_i$. If we are able to converge at each step, we obtain the optimal parameter estimation between the first and the last view without drift.

Obviously in a SLAM framework, we would eventually need to update the reference frame(s). However by keeping a reference image over a long period, we can hope to increase the quality of the motion estimates. In particular, under a stochastic framework, we can hope to improve the map consistency.

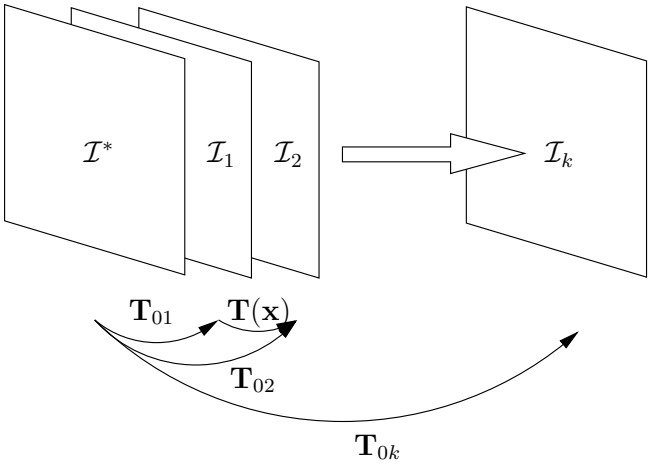


Figure 6.1: Incremental calculation of the transformation

6.3.2 Homographies for the spherical perspective projection model

Let $\mathbf{R} \in \mathbb{SO}(3)$ be the rotation of the camera and $\mathbf{t} \in \mathbb{R}^3$ its translation. The standard planar homography matrix $\mathbf{H}$ is defined up to a scale factor:

$$\mathbf{H} \sim \mathbf{R} + \mathbf{t}\mathbf{n}_d^{*\top} \tag{6.7}$$

where $\mathbf{n}_d^* = \mathbf{n}^*/d^*$ is the ratio between the normal vector to the plane $\mathbf{n}^*$ (a unit vector) and the distance d^* of the plane to the origin of the reference frame. In the following sections, we will call $\mathbf{n}_d^*$ the plane normal by "abuse of language". Homographies are projective properties and stay valid for all single viewpoint sensors. Figure 6.2 illustrates the transformation induced by a planar homography using the spherical perspective projection model. The points $\boldsymbol{\mathcal{X}}_s^*$ and $\boldsymbol{\mathcal{X}}_s$ are related by:

$$\exists(\rho, \rho^*) \in \mathbb{R}^2, \;\; \boldsymbol{\mathcal{P}} = \rho\boldsymbol{\mathcal{X}}_s = \rho^*\mathbf{H}\boldsymbol{\mathcal{X}}_s^*$$

this property is sometimes written as a collinearity constraint (with $\times$ the vector product):

$$\boldsymbol{\mathcal{X}}_s \times (\mathbf{H}\boldsymbol{\mathcal{X}}_s^*) = 0$$

A homography is defined up to a scale factor. In order to fix the scale, we choose $\det(\mathbf{H}) = 1$, i.e. $\mathbf{H} \in \mathbb{SL}(3)$ (the Special Linear group of dimension 3, discussed in Section 5.4). This choice is well justified since $\det(\mathbf{H}) = 0$ happens only if the observed plane passes through the optical center of the camera (in this singular case the plane is not visible any more).

From a planar homography, it is possible to extract the transformation and plane normal [Faugeras and Lustman, 1988]. However two solutions are obtained which explains why we have to distinguish the tracking of a single plane (through a unique homography) and tracking multiple planes. Recently, Benhimane and Malis showed that for visual servoing it is not necessary to decompose the homography [Benhimane and Malis, 2006].

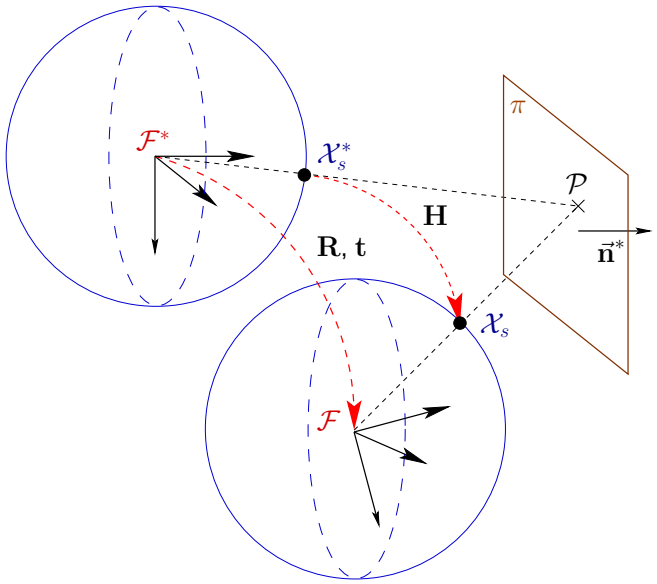


Figure 6.2: Planar homography for the spherical perspective projection

6.3.3 Warping

We will call *warping*, noted $\mathbf{w}$, a coordinate transformation induced by a homography $\mathbf{H} \in \mathbb{SL}(3)$:

$$\begin{array}{rccc} \mathbf{w}: & \mathbb{SL}(3) \times \mathbb{S}^2 & \longrightarrow & \mathbb{S}^2 \\ & (\mathbf{H}, \boldsymbol{\mathcal{X}}^*) & \longmapsto & \boldsymbol{\mathcal{X}} = \mathbf{w}\!<\!\mathbf{H}\!><\!\boldsymbol{\mathcal{X}}^*\!> \end{array}$$

The identity matrix $\mathbf{I}$ is the neutral of the transformation group. We have the following properties:

- $\mathbf{w}\!<\!\mathbf{I}\!><\!\boldsymbol{\mathcal{X}}\!>$ is the identity map:

$$\forall \boldsymbol{\mathcal{X}} \in \mathbb{S}^2, \quad \mathbf{w}\!<\!\mathbf{I}\!><\!\boldsymbol{\mathcal{X}}\!> = \boldsymbol{\mathcal{X}} \tag{6.8}$$

- the composition of two actions corresponds to the action of the composition:

$$\forall \boldsymbol{\mathcal{X}} \in \mathbb{S}^2, \ \forall (\mathbf{H}_1, \mathbf{H}_2) \in \ \mathbb{SL}(3)^2, \mathbf{w}\!<\!\mathbf{H}_1\!><\!\mathbf{w}\!<\!\mathbf{H}_2\!><\!\boldsymbol{\mathcal{X}}\!>\!> = \mathbf{w}\!<\!\mathbf{H}_1\mathbf{H}_2\!><\!\boldsymbol{\mathcal{X}}\!> \tag{6.9}$$

Practically, warping will consist in finding the intensity of the transformation of an image point in a new view. Due to discretisation, we will have to calculate an *approximate* intensity in the new position. Several standard techniques exist. *Nearest neighbour* consists in taking the closest discretised point to the new point. It has the advantage of being fast. *Bilinear interpolation* consists in calculating the average of four neighbouring pixels weighted by their relative distances. It is slower but gave much better results than the nearest neighbour in our tracking tests. "Higher order" approximations are also possible (eg. cubic) but we observed only a very small gain in quality for a higher computational load. For this reason, in all the following experiments, we will use bilinear interpolation.

For omnidirectional vision, we might ask if bilinear interpolation is still valid. The intensity in a given point depends on the solid angle formed by a pixel. Formally, to find the best bilinear interpolation, we should calculate the geodesic distance on the unit sphere. However what is important is the *relative* distance that will only change slightly due to distortion as the calculation is *local*. In our work, the image warpings were done with a bilinear transformation taken in the image.

6.3.4 Tracking a single plane

Let $\mathcal{I}^*$ be the reference image. We will call reference template, a region of size q of $\mathcal{I}^*$ corresponding to the projection of a planar region of the scene.

Consider the following diagram, illustrated by figure 6.2:

$$\begin{array}{ccc} \mathbf{p}^* & \xrightarrow{\Pi^{-1}} & \boldsymbol{\mathcal{X}}_s^* \\ & & \downarrow \quad \mathbf{w}<\mathbf{H}><.> \\ \mathbf{p} & \xleftarrow{\Pi} & \boldsymbol{\mathcal{X}}_s \end{array} \tag{6.10}$$

Π is the transformation between the sphere and the image plane.

To track the template in the current image $\mathcal{I}$ is to find the transformation $\overline{\mathbf{H}} \in \mathbb{SL}(3)$ that warps the lifting of that region to the lifting of the reference template of $\mathcal{I}^*$, i.e. find $\overline{\mathbf{H}}$ such that:

$$\forall i,\ \mathcal{I}\left(\Pi\left(\mathbf{w}<\overline{\mathbf{H}}><\boldsymbol{\mathcal{X}}_s^{i*}>\right)\right) = \mathcal{I}^*(\mathbf{p}_i^*)$$

We will now use the minimisation approach presented in Section 5.1.5 to ensure we stay in the Lie group at each step. Knowing an approximation $\widehat{\mathbf{H}}$ of the transformation $\overline{\mathbf{H}}$, the problem is to find the incremental transformation $\mathbf{H}(\mathbf{x})$ that minimizes the sum of squared differences (SSD) over all the pixels:

$$\left\{ \begin{array}{l} F(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{q} \|\mathbf{f}_i\|^2 \\ \mathbf{f}_i = \mathcal{I}\left(\Pi\left(\mathbf{w}<\widehat{\mathbf{H}}\mathbf{H}(\mathbf{x})><\boldsymbol{\mathcal{X}}_s^{i*}>\right)\right) - \mathcal{I}^*(\mathbf{p}_i^*) \end{array} \right. \tag{6.11}$$

The increment appears in a product $\widehat{\mathbf{H}}\mathbf{H}(\mathbf{x})$ as the matrix product is the group law of $\mathbb{SL}(3)$ and thus $\mathbf{H}(\mathbf{x}) \in \mathbb{SL}(3)$ implies that $\widehat{\mathbf{H}}\mathbf{H}(\mathbf{x}) \in \mathbb{SL}(3)$. $\mathbf{H}(\mathbf{x})$ is parameterised locally by the Lie algebra of $\mathbb{SL}(3)$:

$$\mathbf{H}(\mathbf{x}) = \exp\left(\sum_{i=1}^{8} x_i \mathbf{A}_i\right)$$

Surprisingly this parameterisation is not standard [Baker et al., 2006]. In equation (6.11), the incremental transformation should appear to the left ($\mathbf{H}_L(\mathbf{x})$) as $\mathbf{H}$ is the transformation from the reference frame to the current frame (we could write ${}^c\mathbf{H}_r$). However we can also write the increment

to the right, the two are related by[1]:

$$\mathbf{H}(\mathbf{x}) = \widehat{\mathbf{H}}^{-1}\mathbf{H}_L(\mathbf{x})\widehat{\mathbf{H}}$$

The advantage of writing the increment to the right appears when deriving the Jacobian that is simpler and faster to calculate.

We show in Appendix B, that the current Jacobian, noted $\mathbf{J}(\mathbf{0})$, and the reference Jacobian, noted $\mathbf{J}(\widetilde{\mathbf{x}})$, can be written as the product of four Jacobians:

$$\mathbf{J}(\mathbf{0}) = \mathbf{J}_{\mathcal{I}}\mathbf{J}_{\Pi}\mathbf{J}_w\mathbf{J}_{\mathbf{H}_\mathbf{x}}(\mathbf{0})$$

$$\mathbf{J}(\widetilde{\mathbf{x}}) = \mathbf{J}_{\mathcal{I}^*}\mathbf{J}_{\Pi}\mathbf{J}_w\mathbf{J}_{(\overline{\mathbf{H}}^{-1}\widehat{\mathbf{H}}\mathbf{H}_\mathbf{x})}(\widetilde{\mathbf{x}})$$

Using Proposition 1, Chapter 5, we have the following property:

$$\mathbf{J}_{(\overline{\mathbf{H}}^{-1}\widehat{\mathbf{H}}\mathbf{H}_\mathbf{x})}(\widetilde{\mathbf{x}})\widetilde{\mathbf{x}} = \mathbf{J}_{\mathbf{H}_\mathbf{x}}(\mathbf{0})\widetilde{\mathbf{x}}$$

Thus, in equation (6.6), we can use $\mathbf{J}_{\mathbf{H}_\mathbf{x}}(\mathbf{0})\widetilde{\mathbf{x}}$ instead of $\mathbf{J}_{(\overline{\mathbf{H}}^{-1}\widehat{\mathbf{H}}\mathbf{H}_\mathbf{x})}(\widetilde{\mathbf{x}})\widetilde{\mathbf{x}}$ where $\mathbf{J}_{\mathbf{H}_\mathbf{x}}(\mathbf{0})$ is a constant Jacobian matrix. The update $\widetilde{\mathbf{x}}$ of the second-order minimization algorithm can be computed as follows:

$$\widetilde{\mathbf{x}} = -\left(\underbrace{\left(\frac{\mathbf{J}_{\mathcal{I}} + \mathbf{J}_{\mathcal{I}^*}}{2}\right)\mathbf{J}_{\Pi}\mathbf{J}_w\mathbf{J}_{\mathbf{H}_\mathbf{x}}(\mathbf{0})}_{\mathbf{J}_{esm}}\right)^{+}\mathbf{f}(\mathbf{0}) \tag{6.12}$$

The computational complexity is almost the same as for first-order algorithms as the reference Jacobian $\mathbf{J}_{\mathcal{I}^*}$ only needs to be calculated once.

Obtaining $\mathbf{J}_{\mathcal{I}^*}$ and $\mathbf{J}_{\mathcal{I}}$, that are the Jacobians taken in the images (for example using a Sobel filter), is a remarkable property. It indicates that we can take into account the non-linear properties of the sensor simply through the Jacobian of the projection function $\mathbf{J}_{\Pi}$ (this appears in the derivation of the Jacobians in Appendix B).

6.3.5 Tracking multiple planes

When tracking multiple planes, we have the choice either to track the planes independently, or to constrain the same motion for each plane which is the object of this section.

From equation (6.7), we can parameterise a homography by a transformation $\mathbf{T} \in \mathbb{SE}(3)$ and a plane normal $\mathbf{n}_d \in \mathbb{R}^3$. With the Lie algebra parameterisation of $\mathbf{T}$:

$$\mathbf{H}(\mathbf{T}(\mathbf{x}), \mathbf{n}_d) = \mathbf{H}\left(\exp\left(\sum_{i=1}^{6} x_i\mathbf{A}_i\right), \mathbf{n}_d\right)$$

We can now reformulate the problem by parameterising each plane with the same transformation $\mathbf{T}$. To track the template j in the current image $\mathcal{I}$ is to find the transformation $\mathbf{H}(\overline{\mathbf{T}}, \overline{\mathbf{n}}_d^j)$ that warps the lifting of that region to the lifting of the reference template of $\mathcal{I}^*$:

$$\forall i, j,\ \ \mathcal{I}\left(\Pi\left(\mathbf{w}{<}\mathbf{H}(\overline{\mathbf{T}}, \overline{\mathbf{n}}_d^j){>}{<}\boldsymbol{\mathcal{X}}_s^{ij*}{>}\right)\right) = \mathcal{I}^*(\mathbf{p}_{ij}^*) \tag{6.13}$$

[1] this is the adjoint map $Ad_{\widehat{\mathbf{H}}^{-1}}$ in $\mathbb{SL}(3)$

In other words, knowing an approximation $\widehat{\mathbf{T}}$ of $\overline{\mathbf{T}}$ and $\widehat{\mathbf{n}}_d^j$ of $\overline{\mathbf{n}}_d^j$, the problem is to find the incremental transformation $\mathbf{T}(\mathbf{x})$ and $\mathbf{n}_d^j(\mathbf{x})$ that minimises the sum of squared differences (SSD) over all the pixels and over the m planes ($\mathbf{x}$ contains the 6 transformation parameters and the $3 \times m$ parameters for the normals and depths):

$$\begin{cases} F(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^{m} \sum_{i=1}^{q_j} \|\mathbf{f}_{ij}\|^2 \\ \mathbf{f}_{ij} = \mathcal{I}\left(\Pi\left(\mathbf{w}<\mathbf{H}(\mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}, \widehat{\mathbf{n}}_d^j + \mathbf{n}_d^j(\mathbf{x}))><\boldsymbol{\mathcal{X}}_s^{ij*}>\right)\right) - \mathcal{I}^*(\mathbf{p}_{ij}^*) \end{cases} \tag{6.14}$$

The minimal number of parameters in equation (6.14) is in fact $6 + 3 \times m - 1$ because the first homography has only 8 degrees of freedom. However the extra degree of freedom empirically gave better results probably due to the better conditioning of the Jacobian (all the values for the normals have the same amplitude).

Similarly to the previous case, the Jacobians $\mathbf{J}(\mathbf{0})$ and $\mathbf{J}(\widetilde{\mathbf{x}})$, that correspond respectively to the current and the reference Jacobians, can be written as (see Appendix C):

$$\mathbf{J}(\mathbf{0}) = \mathbf{J}_{\mathcal{I}}\, \mathbf{J}_{\Pi}\, [\mathbf{J}_{H_T}\mathbf{J}_T(\mathbf{0}) \quad \mathbf{J}_{H_n}\mathbf{J}_n(\mathbf{0})] \tag{6.15}$$

$$\mathbf{J}(\widetilde{\mathbf{x}}) = \mathbf{J}_{\mathcal{I}^*}\, \mathbf{J}_{\Pi}\, [\mathbf{J}_{H_T^*}\mathbf{J}_{T^*}(\widetilde{\mathbf{x}}) \quad \mathbf{J}_{H_n^*}\mathbf{J}_{n^*}(\widetilde{\mathbf{x}})] \tag{6.16}$$

with H_T the homography seen as a function of the transformation $\mathbf{T}$ and H_n the homography seen as a function of the plane normal $\mathbf{n}_d$.

Using proposition 1, we have the following property:

$$\mathbf{J}_{T^*}(\widetilde{\mathbf{x}})\widetilde{\mathbf{x}} = \mathbf{J}_T(\mathbf{0})\widetilde{\mathbf{x}} \tag{6.17}$$

we also have: $\mathbf{J}_{n^*}(\widetilde{\mathbf{x}})\widetilde{\mathbf{x}} = \mathbf{J}_n(\mathbf{0})\widetilde{\mathbf{x}}$.

If we assume that $\widehat{\mathbf{T}} \approx \overline{\mathbf{T}}$ and $\widehat{\mathbf{n}} \approx \overline{\mathbf{n}}$, $\mathbf{J}_{H_T^*} \approx \mathbf{J}_{H_T}$ and $\mathbf{J}_{H_n^*} \approx \mathbf{J}_{H_n}$, the update $\widetilde{\mathbf{x}}$ of the second-order minimisation algorithm can then be computed as follows:

$$\widetilde{\mathbf{x}} = -\left(\underbrace{\left(\frac{\mathbf{J}_{\mathcal{I}} + \mathbf{J}_{\mathcal{I}^*}}{2}\right)\mathbf{J}_{\Pi}\, [\mathbf{J}_{H_T T}(\mathbf{0}) \quad \mathbf{J}_{H_n n}(\mathbf{0})]}_{\mathbf{J}_{esm}}\right)^{+} \mathbf{f}(\mathbf{0}) \tag{6.18}$$

We may also note that the matrix is sparse so the algorithm can make the most of sparse linear algebra and avoid the full inversion of the Jacobian matrix [Hartley and Zisserman, 2000].

6.4 Efficient ESM implementation and variants

In this section, we will detail how to obtain an efficient implementation of the ESM algorithm. We will discuss how to improve the computational cost and stay close to the second order convergence rate. Validation on simulated and real data will be given in the next section. [Madsen et al., 2004] gives an overview of generic descent methods.

Algorithm 1 describes a basic iterative descent method for ESM single plane SSD tracking. The computation is split between evaluation of the cost function and calculation of the descent direction.

Algorithm 1: ESM tracking method: basic algorithm for a single plane

Data: Current image $\mathcal{I}$ and reference image $\mathcal{I}^*$. Constant transformation Jacobians $\mathbf{J}_\Pi$ and $\mathbf{J}_w$. (ε, k_{max}) : thresholds, $\widehat{\mathbf{H}}$: initial estimate of the homography
Result: Local minimum $\overline{\mathbf{H}}$
Calculate $\mathbf{J}_{\mathcal{I}^*}$ (eg. Prewitt, Sobel filters)
k := 0; **found** := false
while *(not* **found***) and (*$k < k_{max}$*)* **do**
  Calculate $\mathbf{J}_{\mathcal{I}}$.
  Calculate $\mathbf{J}_{esm}$ from equation (6.12).
  $\widetilde{\mathbf{x}} = -(\mathbf{J}_{esm}^\top \mathbf{J}_{esm})^{-1}\mathbf{J}_{esm}^\top \mathbf{f}(\mathbf{0})$
  $\widehat{\mathbf{H}} \leftarrow \widehat{\mathbf{H}} e^{\mathbf{A}(\widetilde{\mathbf{x}})}$
  if $\|\widetilde{\mathbf{x}}\| < \varepsilon$ **then**
    found := true
  end
  k := k+1
end

Let q be the number of pixels of the template (several hundred or several thousand) and m the number of parameters. In [Baker et al., 2004], the authors calculate the cost in terms of operations of the different steps appearing in SSD tracking algorithms. The pseudo-inversion and more specifically the product $\mathbf{J}_{esm}^\top \mathbf{J}_{esm}$ is particularly expensive as it requires $o(m^2 q)$ operations. Calculating the cost function in comparison has a complexity of $o(mq)$.

This has motivated the development of the inverse compositional algorithm where the increment is only based on the *reference* Jacobian, noted $\mathbf{J}_{inv}$ [Baker et al., 2004]. The advantage of such an approach is that $(\mathbf{J}_{inv}^\top \mathbf{J}_{inv})^{-1}\mathbf{J}_{inv}^\top$ can be precalculated and the complexity drops to $o(mq+m^3)$ instead of the initial $o(m^2 q + m^3)$. However the approach is not well adapted to changes in illumination as the Jacobian stays constant (there have been some improvements in this direction [Bartoli, 2006]). For the same reason, occlusion (including the template partly going out of the image) cannot be handled without needing to recalculate $\mathbf{J}_{inv}^+$. It also should not be used when estimating the motion directly *even* when the structure is known as the reference Jacobian then depends on a *combination* of the displacement and structure. We will show experimentally the effect of using the incorrect Jacobian on the section dedicated to experimental validation.

In the same frame of mind as for the inverse compositional, we can consider three possible variants of the ESM to obtain a complexity of $o(mq + m^3)$:

- **cESM**:
$$\widetilde{\mathbf{x}} = -(\mathbf{J}_{inv}^\top \mathbf{J}_{inv})^{-1}\mathbf{J}_{esm}^\top \mathbf{f}(\mathbf{0})$$
This is only valid for image-based visual tracking. We can justify the approximation by saying that as we get closer to the optimum, $\mathbf{J}_{inv} \approx \mathbf{J}_{esm}$.

- α**ESM**:
$$\widetilde{\mathbf{x}} = -\frac{(\mathbf{g}^\top \mathbf{J}_{inv}^p \mathbf{g})}{\|\mathbf{J}_{esm}\mathbf{J}_{inv}^p \mathbf{g}\|^2}\mathbf{J}_{inv}^p \mathbf{g}$$
with $\mathbf{g} = \mathbf{J}_{esm}^\top \mathbf{f}(\mathbf{0})$ and $\mathbf{J}_{inv}^p = (\mathbf{J}_{inv}^\top \mathbf{J}_{inv})^{-1}$. The corrective term was found in the following way:
  - $(-\mathbf{g})$ can be considered as a second-order steepest descent direction,

– we then look for the corrective term α that minimises the second-order approximation to the cost function:

$$\begin{aligned} F(-\alpha \mathbf{J}_{inv}^{p}\mathbf{g}) &= \frac{1}{2}\left\|\mathbf{f}(\mathbf{0}) - \alpha \mathbf{J}_{esm}\mathbf{J}_{inv}^{p}\mathbf{g}\right\|^2 \\ &= F(\mathbf{0}) - \alpha \mathbf{f}(\mathbf{0})^\top \mathbf{J}_{esm}\mathbf{J}_{inv}^{p}\mathbf{g} - \frac{1}{2}\alpha^2 \left\|\mathbf{J}_{esm}\mathbf{J}_{inv}^{p}\mathbf{g}\right\|^2 \end{aligned}$$

by differentiation we obtain the optimal α and the desired expression.

This solution can be expected to converge better than the previous approach as the corrective term means we are closer to the second-order estimate at the beginning of the minimisation. The calculation is still in $o(mq + m^3)$ but takes longer than the previous method meaning that for a fixed time, it could have a worse convergence rate.

- **iESM**:

$$\widetilde{\mathbf{x}} = -\frac{(\mathbf{g}^\top \mathbf{J}_{esm}^{p}\mathbf{g})}{\|\mathbf{J}_{esm}\mathbf{J}_{esm}^{p}\mathbf{g}\|^2}\mathbf{J}_{esm}^{p}\mathbf{g}$$

with $\mathbf{g} = \mathbf{J}_{esm}^\top \mathbf{f}(\mathbf{0})$ and $\mathbf{J}_{esm}^{p} = (\mathbf{J}_{esm}^\top \mathbf{J}_{esm})^{-1}$ calculated at the beginning of the iterations. This approach has the advantage of being valid for explicit structure and motion. It can be justified by saying that at the beginning, we have the best second-order estimate and that we then correct the Jacobian so that it stays valid at the optimum. It is not as satisfying as the previous methods as the Jacobian will *not* be correct at the optimum. What we "hope" is that it will be sufficiently good, thanks to the corrective term, to lead to the optimum anyway.

Reducing the computational cost of the iterations is only one side of the problem. Ideally we would also want to diminish the number of iterations. The ESM algorithm, as a second order method, converges faster and more often than a first order method. It is also well adapted to changes in illumination. In [Silveira et al., 2007], the authors take into account explicitly an affine illumination model.

The following section will be dedicated to comparing the different algorithms and the effect of the approximations on the convergence.

6.5 Experimental validation

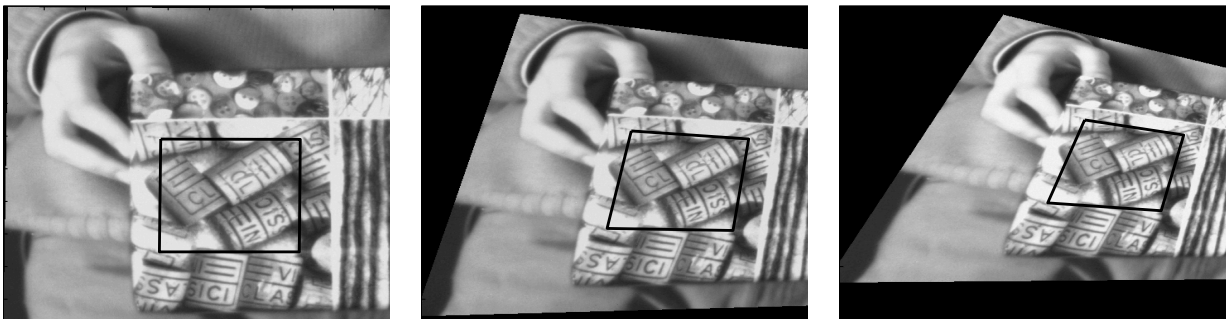
6.5.1 Simulation

6.5.1.1 Affect of the inverse compositional for explicit motion estimation

As explained previously, the reference Jacobian for multiple planes depends not only on the structure but also on the position. Using the inverse compositional (**IC**) algorithm in this case can lead to poor results as the Jacobian can have arbitrarily big errors. Alas the inverse compositional has often been used to track in this situation. We will now simulate the effect of using the **IC** in this situation.

Our simulation setup consists of a sequence of 50 images without any added noise and with small inter-frame displacements. Figures 6.3 (a),(b) and (c) show images 1, 25 and 50 respectively of the simulated sequence.

Figure 6.4 shows the number of iterations taken to converge. As we can see the inverse compositional (**IC**) takes more and more iterations. At the end of the sequence, the **IC** took more than



(a) (b) (c)

Figure 6.3: Image number 1 (a), 25 (b) and 50 (c) of the artificial sequence

2000 iterations to converge. The forward compositional (**FC**) took systematically 6 iterations and the efficient second-order minimisation (**ESM**) took 5 iterations at the same computational cost (the inter-frame displacements were small so few iterations were needed to converge).

This experiment confirms that the Jacobian of the **IC** is indeed incorrect and cannot be used for 3D tracking even when the structure is known and the inter-frame displacement is small. The Jacobian is increasingly incorrect and it becomes harder and harder to converge and the values oscillate around the true value. This result is particularly important for omnidirectional visual tracking as we expect to track templates over larger image regions than in the standard perspective case.

6.5.1.2 Comparison between methods

To compare the different algorithms, we will use the Matlab program written at the CMU for the project "Lucas-Kanade 20 Years On".

The following methods will be compared:

- forward compositional (**FC**),
- inverse compositional (**IC**),
- efficient second-order minimisation (**ESM**),
- **cESM**, α**ESM**, **iESM** as described previously.

Evaluating the computational time per iteration To evaluate the time taken for an iteration, we programmed the iteration steps in C language and tested the times for image sizes ranging from 20×20 to 500×500. The inverse compositional has the lowest computational cost and was compared to the other methods in figure 6.5. In figure 6.6, we plot the iteration time versus the number of pixels, we can see that the computational time is approximately a linear function of the number of pixels. **ESMseq** is an implementation of the ESM algorithm where the different parts are computed separately: first $\mathbf{J}_{esm}$, then $\mathbf{J}_{esm}^{+}$, $\mathbf{J}_{esm}^{\top}\mathbf{f}(\mathbf{0})$ and finally $\widetilde{\mathbf{x}}$. However building $\mathbf{J}_{esm}$ does not need to be done explicitly. It is possible to build $\mathbf{J}_{esm}^{\top}\mathbf{J}_{esm}$ and $\mathbf{J}_{esm}^{\top}\mathbf{f}(\mathbf{0})$ in the same iteration which corresponds to the **ESM** values in the figures. We can see that the computational time is halved. This can be explained by the fact that we only access a small amount of memory that has been cached by the processor. This aspect is rarely taken into account when calculating the computational cost of an

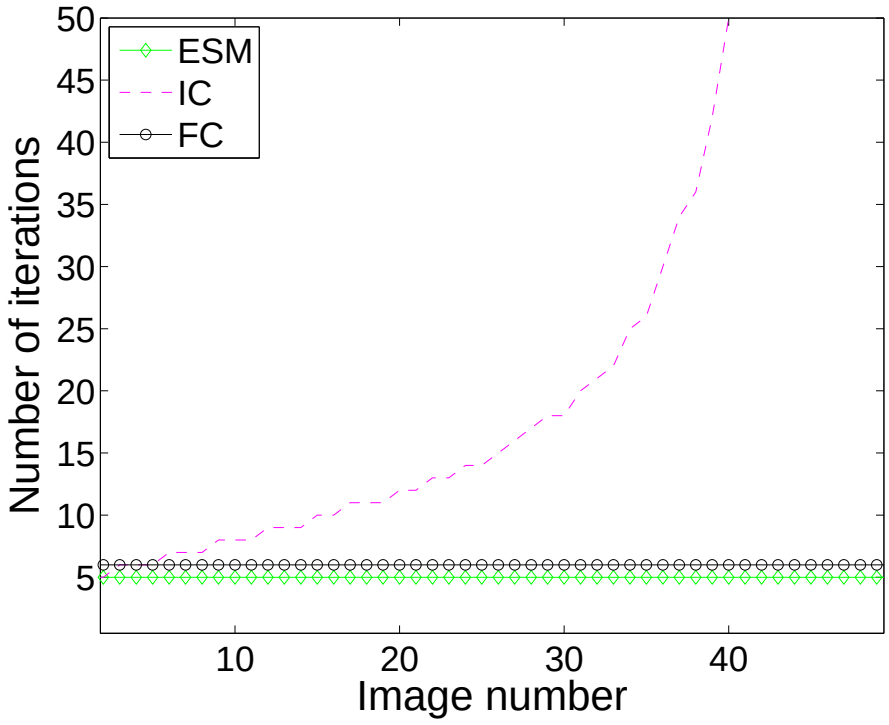


Figure 6.4: Comparaison of the number of iterations taken to converge for the simulation sequence

algorithm: it can take longer to access memory than to recompute part of the data if we use the same variables. This is what happens for **cESM** that has a similar cost to the **IC**. Accessing the precalculated $\mathbf{J}^{+}_{inv}$ value is similar to calculating the Jacobian of the current image and the value $\mathbf{J}_{esm}$ implicitly. A possible explanation of the change in computational time for **ESMseq**, α**ESM** and **iESM** when the size of patches are greater than 160×160 can be found in the need to calculate $\mathbf{J}_{esm}$. For **ESMseq**, the value is calculated and then used for $\mathbf{J}^{+}_{esm}$ and $\mathbf{J}^{\top}_{esm}\mathbf{f}$. Similarly α**ESM** and **iESM** need the value of the Jacobian to process the corrective term. In both cases, this value can be cached for the second step. However when the Jacobian data no longer holds in the first cache, it saves it in a slower cache (until eventually going in main memory and then in the swap space).

We believe this aspect of computation should be taken into account in future research on computer vision algorithms. Only considering the number of operations leads to a too simplistic analysis and incorrect conclusions.

The relative times are summarised in Table 6.1. We chose to use the "starting" values of figure 6.5 for α**ESM** and **iESM** as they correspond to the range of template sizes we would expect to track.

To give an order of magnitude of the computational time of the tracking algorithm, an iteration of the **ESM** for a template of size 100×100 on a 3.6 GHz Intel Pentium 4 took 1 ms.

Tests The following tests were undertaken:

- "at infinity". In other words we give an amount of iterations that are far more than would be used in a real tracking situation (typically 150, see Table 6.1). This is to test the convergence properties of the algorithms.

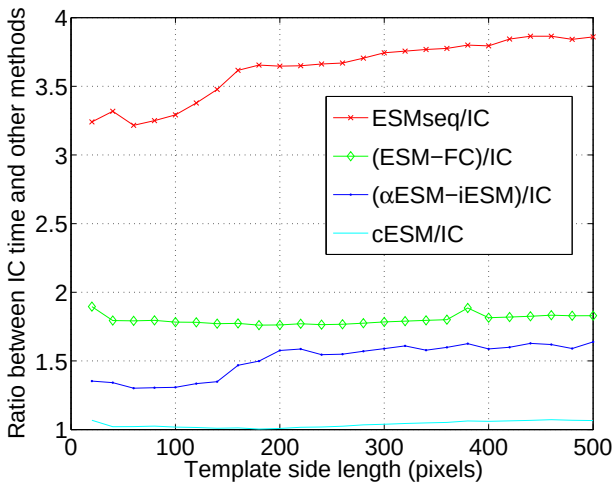


Figure 6.5: Time comparison

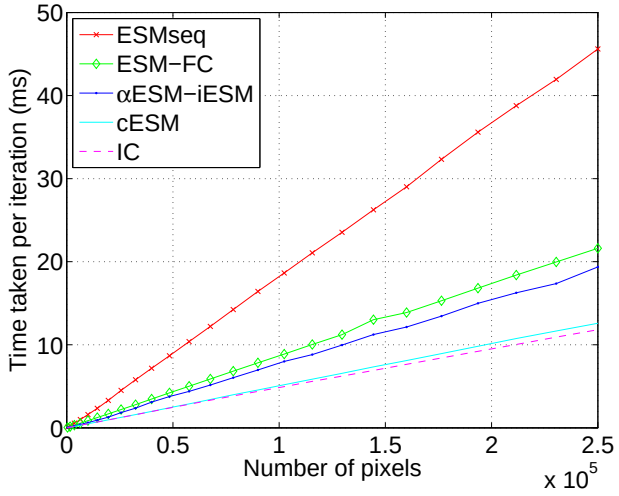


Figure 6.6: Time (ms) vs number of pixels

Table 6.1: Estimated time for an iteration and number of iterations

	Time taken per iteration (ratio with respect to the **IC**)	Number of iterations given "at infinity"	Number of iterations given at "fixed time"
IC	1	150	20
cESM	1.04	144	19
αESM-iESM	1.33	112	15
ESM-FC	1.8	83	11

- "fixed time". We give a typical amount of iterations (eg 20 for the **IC**, see Table 6.1). The number of iterations are calculated in order to give the same amount of time to each algorithm.
- "added noise": noise is added to the incoming and reference images. This is a more realistic test as it simulates unmodeled errors that could appear through lighting changes or occlusion. We used the same number of iterations than for the "fixed time" case.

Figures 6.8, 6.9 and 6.7 show the frequency of convergence of the different algorithms as the homography transformation increases (a Gaussian error is added to the plane points). The number of tests for each parameter variation was of 500 "at infinity" and 1000 for the two other cases.

The test "at infinity" in figure 6.7 confirms that there are two groups: the algorithms with a second-order convergence (ESM and variants) and the algorithms with first-order convergence properties. This experiment indicates that the ESM variants keep the second-order convergence rate despite the approximations.

The tests at "fixed time" and "with noise" show that the two variants **αESM** and **iESM** are possible alternatives to the direct **ESM** approach. The difference in convergence rate changes, as is to be expected, when noise is added to the images. The reason **iESM** has the same convergence rate as **αESM** in the test "with noise" is that the calculation of the pseudo-inverse at the beginning of the

iterations compensates in part for the image errors. Even though **cESM** has a computational cost of the same order as **IC**, it does not lead to a significant improvement in convergence rate.

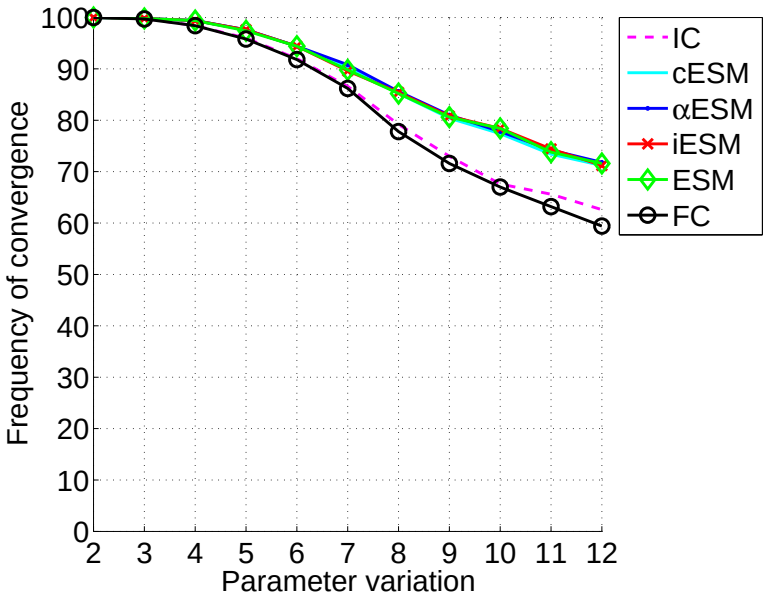


Figure 6.7: Frequency of convergence vs homography motion for an "infinite time"

6.5.1.3 Conclusion

In the simulation experiments we compared the convergence properties of the ESM algorithm and different variants to more standard approaches such as the inverse compositional (**IC**) and the forward compositional (**FC**). We showed that even though the iteration step of the ESM (and variants) is more computationally expensive than an iteration of the **IC**, the higher convergence rate makes it globally faster. If we add the possibility of working on occlusion and changes of illumination, the algorithm is altogether a good alternative to the **IC** for image-based tracking. Variants of the ESM, α**ESM** and **iESM** were shown to improve the results further. Another important experiment confirmed that the **IC** should not be used for 3D tracking even if the structure is known.

6.5.2 Real data

The algorithm was tested on real data obtained from the mobile robot ANIS. The central catadioptric camera was comprised of the S80 parabolic mirror from RemoteReality with a telecentric lens and perspective camera of resolution 1024×768[2].

The two tests for a single plane and multiple planes were done over 120 images and a distance of about 2 m. The robot odometry was considered as ground truth. These experiments were undertaken to analyse the precision of the algorithm in view of its integration into a SLAM framework. We also would like to know if it is worthwhile imposing the constraint of same camera motion when tracking different planes.

[2]the camera was calibrated using the method described in Chapter 3

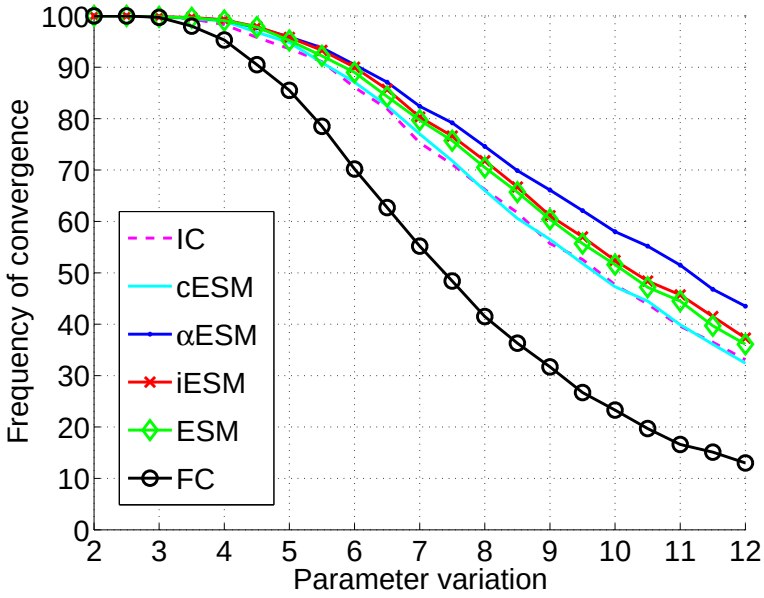


Figure 6.8: Frequency of convergence vs homography motion for a "fixed time" without noise

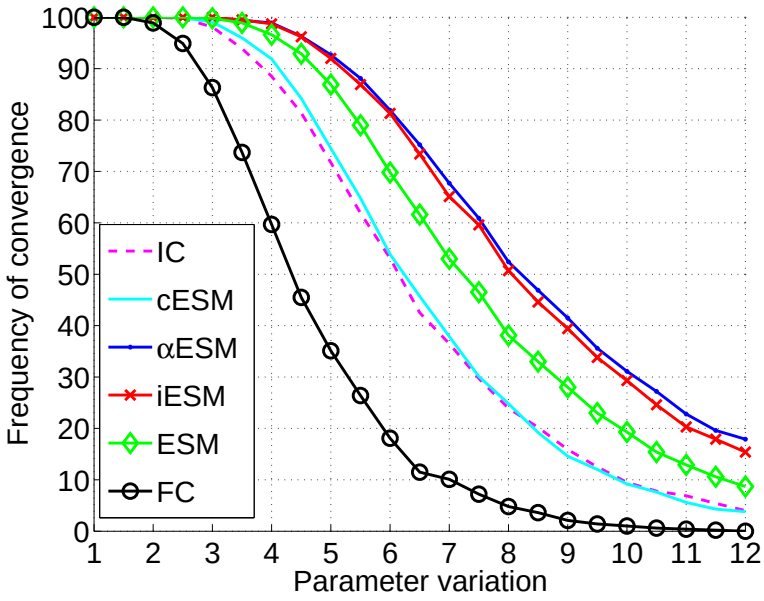


Figure 6.9: Frequency of convergence vs homography motion for a "fixed time" with noise

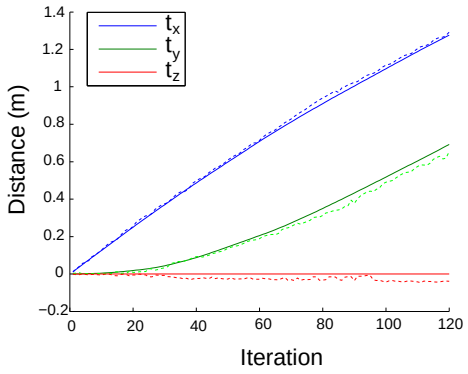
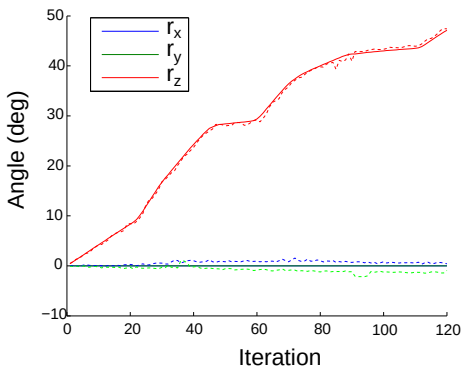
6.5.2.1 Single plane

In the case of a single plane, four reference templates were tracked (figure 6.10). They are numbered in counterclockwise order from 1 to 4 starting top left. Templates number 2 and number 3 were considered to be on the same plane (i.e. only one homography was estimated). For each homography, a translation $\mathbf{t}$ up to a scale factor and a rotation $\mathbf{R}$ can be extracted (the ambiguity was solved by using several frames). The scale was then fixed by measuring the distance of the camera to one of the planes. Table 6.2 summarises the results. The figures show the median of the estimated motions in dotted lines and the odometry in full lines. The angles estimated between the planes were of 87° and 91° .

The camera field of view was obstructed in the case of the template number 1 after the image 100 as we can see in figure 6.14, figure 6.15 and figure 6.16. The algorithm which uses a straight forward minimization was not able to find the correct homography, this does not appear in the motion estimation as a median is used.

Template number 4 was correctly tracked as we can see in figure 6.16, the complex motion depicted in the images (figure 6.11-6.15) is only due to the mirror geometry.

Table 6.2: Estimation of the parameters

	Translation (cm)	Rotation (deg)
		
Max. error	[2.7, 3.6, 7.3]	[1.6, 2.2, 1.0]
Mean abs. error	[1, 1.3, 1.4]	[0.8, 0.6, 0.3]

6.5.2.2 Multiple planes

In this experiment, we would like to know if constraining the same camera motion while tracking can improve the quality and robustness of the structure from motion. Figure 6.17 shows the templates tracked in the experiment. The planes are numbered in counterclockwise order from 1 to 3 starting top left. To fix the scale factor, we measured the distance from the camera to the third plane (0.5 m) (the plane that proved the stablest while tracking).

The sequence is composed of 120 images. The mobile robot covered a distance of about 2 m. The initial values given for the normals with depths was $[1; 0; 0]$ (the same results were obtained for values $[0; 1; 0]$, $[0; 0; 1]$, $[0; 0; 1000]$ ...). These initial values are far from the "real" values that can be deduced

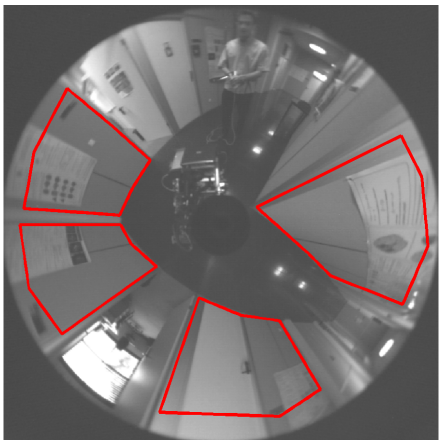


Figure 6.10: Reference image

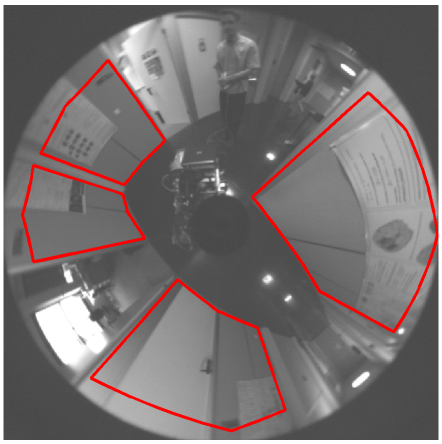


Figure 6.11: Image 25

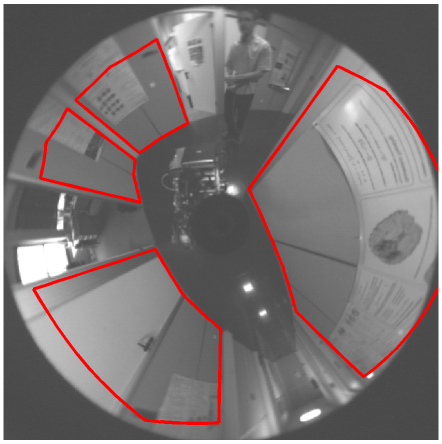


Figure 6.12: Image 50

Figure 6.13: Image 75

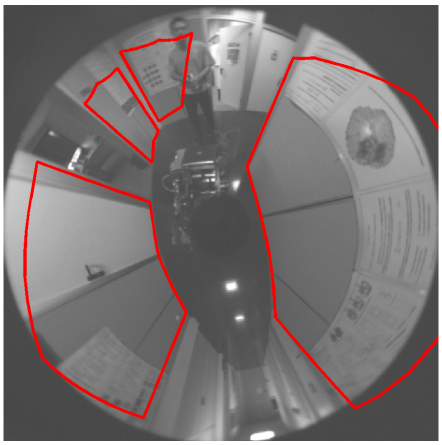


Figure 6.14: Image 100

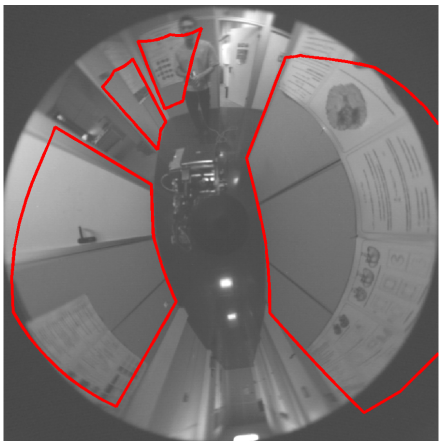


Figure 6.15: Image 120

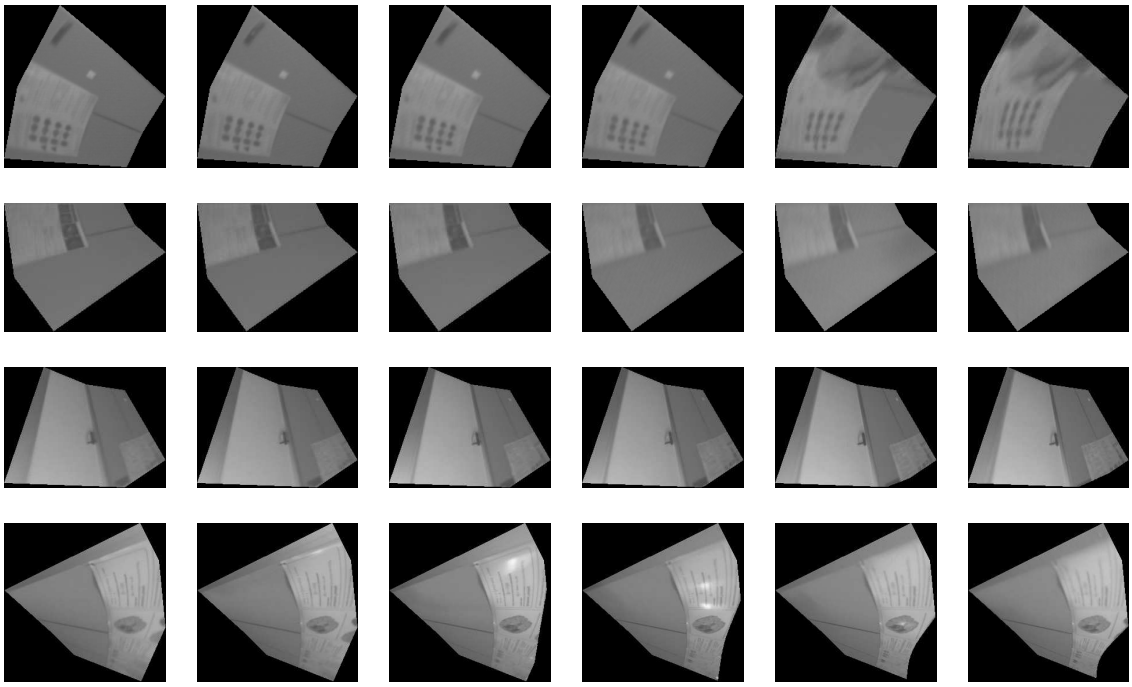


Figure 6.16: Reprojection of the templates for iterations 0,25,50,75,100,120 in the reference image using the estimated homography

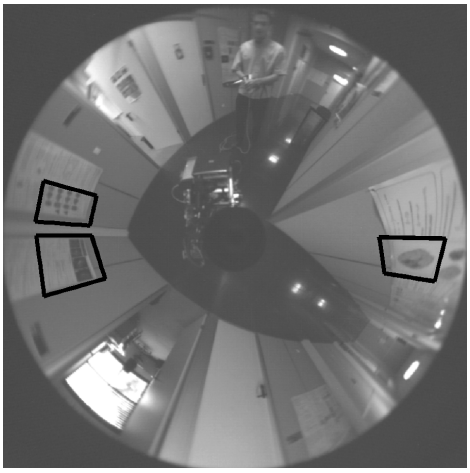


Figure 6.17: Tracked templates

from figure 6.25 and figure 6.26 (with $d_3 = 0.5$ m): $[-0.38; -0.31; 0]$, $[-0.4; 0.6; 0]$ and $[1.2; -1.6; 0]$. The algorithm proved to be relatively insensitive to the initial values when an extra degree of freedom was given. This can be explained from the normals appearing in the homography as a product with the translation.

The motion for the planes tracked independently was obtained in the same way as in the previous section, by applying the median over the rotation and translation recovered from the homographies, we will call this algorithm **SPT**.

The algorithm for the constrained tracking was tested with a forward compositional minimisation (**MPT_FC**) and with the proposed algorithm (**MPT_ESM**).

Figures 6.18 to 6.23 compare the odometry (in full lines) to the motion estimation using the **SPT**, **MPT_FC** and **MPT_ESM** algorithms (lines with symbols). The number of iterations needed to converge appears on the figure for the **MPT_FC** and **MPT_ESM** algorithms in a black dotted line with the number of iterations indicated on the right Y-axis. Figure 6.24 shows the templates at different stages in the tracking (only for **MPT_ESM**). The normals estimated on-line are represented in figure 6.25. The distances estimated for planes 1 and 2 are detailed in figure 6.26.

MPT_ESM which is a close to second-order approach gave slightly more precise results than **MPT_FC** (first-order approach) and in less iterations: 7 iterations were needed for **MPT_ESM** compared to 13 for **MPT_FC** (median value over the first 60 images). We will now compare **MPT_ESM** to **SPT**.

The motion estimation was precise except between iterations 75 and 100 where a reflection (that can be seen on the tracked templates in figure 6.24) on the poster of template 3 generated errors in the normal estimates but also in the distance estimates. We used a non-robust minimisation approach which is not able to cope adequately with illumination errors (we will detail a simple robust algorithm in the following section). However the tracking was able to recover after iteration 110. When the templates were tracked independently (i.e. the motion and normal estimates were extracted directly from the homography), the patches did not give a sufficient amount of information to enable a good estimate of the motion. With the illumination problem arising on the *stablest* estimated plane, the motion estimation becomes erratic (figure 6.18 and figure 6.19).

The **MPT_ESM** gave a translation estimation with a maximum error of [16,23,3] cm for [x,y,z] and an absolute mean error of [2.6, 2.0, 1.6] cm, for the rotation the maximum error was of [1.85, 0.43, 0.64] deg over the [x,y,z] rotation axis with a mean error of [1.14, 0.22, 0.22] deg. Estimating the distance proved sensitive to small errors, the variance over the sequence was respectively of $\sigma = 23.6$ cm and $\sigma = 7.33$ cm for planes 1 and 2.

Figure 6.27 shows the motion of the robot in the XY-plane for **MPT_ESM** with the odometry depicted in full lines and the estimated motion with connected crosses. The planes are also represented in the image from the estimates. The angles between the corridor walls were quite precisely estimated with 92.9 deg between planes 1 and 2 and 87.3 deg between planes 1 and 3. The results obtained using **SPT** depicted using connected circles in figure 6.27 gave poor results.

6.5.2.3 Conclusion

The experiments show that the ESM visual tracking can be used as an efficient method to estimate 6 DOF motion. They also confirmed that constraining the same camera motion when tracking improves the robustness of the algorithm and enables the tracking of smaller planes.

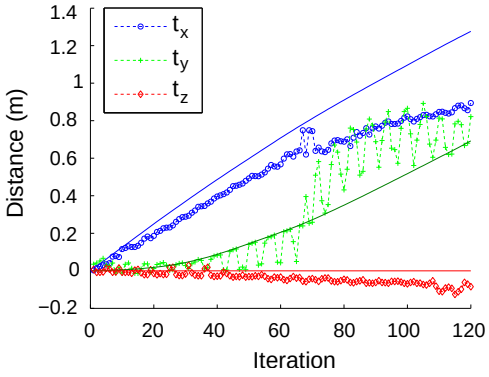


Figure 6.18: Estimation of the robot's translation (**SPT**)

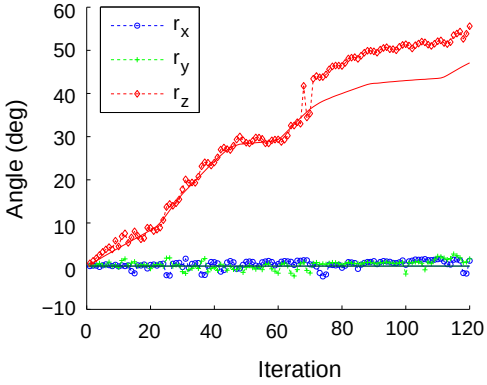


Figure 6.19: Estimation of the robot's rotation (**SPT**)

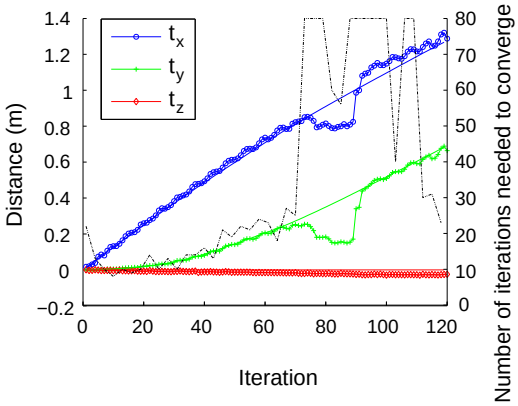


Figure 6.20: Estimation of the robot's translation (**MPT_FC**)

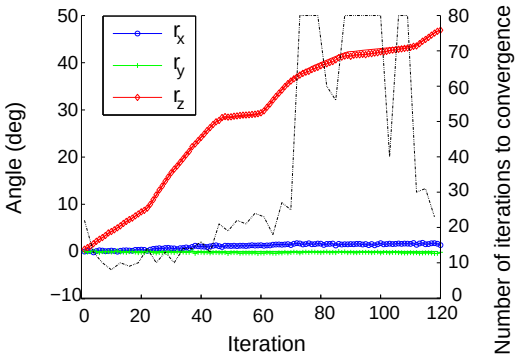


Figure 6.21: Estimation of the robot's rotation (**MPT_FC**)

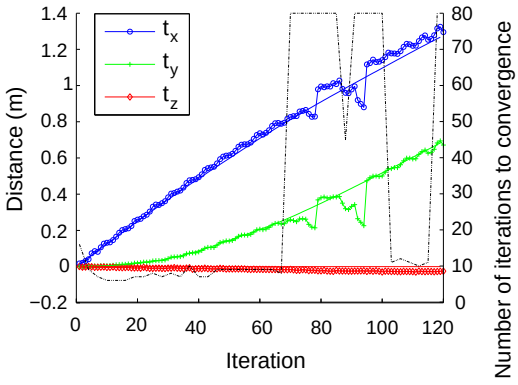


Figure 6.22: Estimation of the robot's translation (**MPT_ESM**)

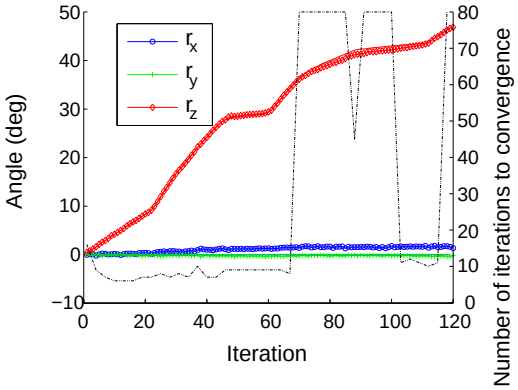


Figure 6.23: Estimation of the robot's rotation (**MPT_ESM**)

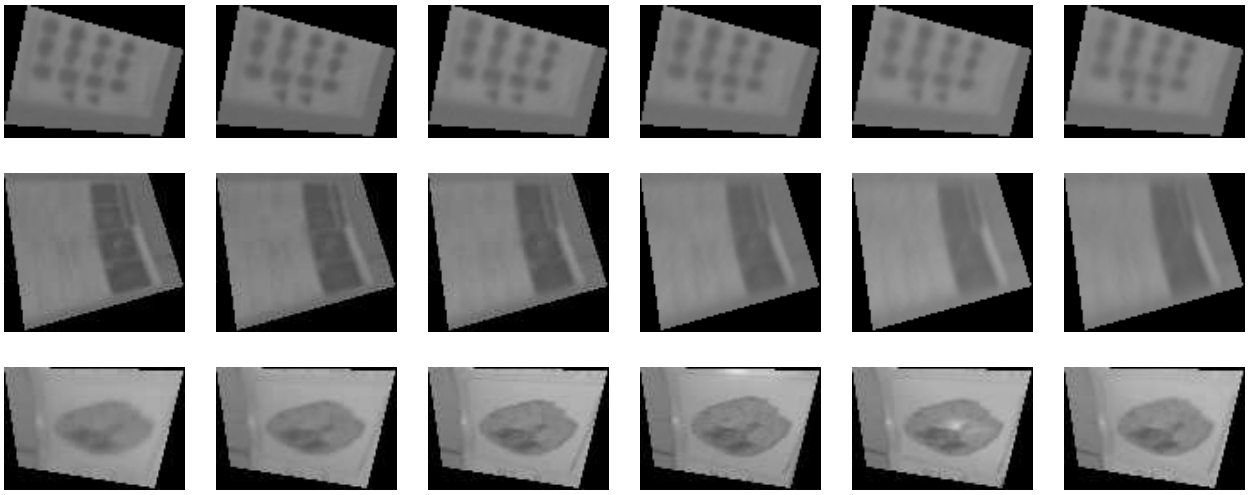


Figure 6.24: Reprojection of the templates for iterations 0,25,50,75,100,120 in the reference image using the estimated homography (**MPT_ESM**)

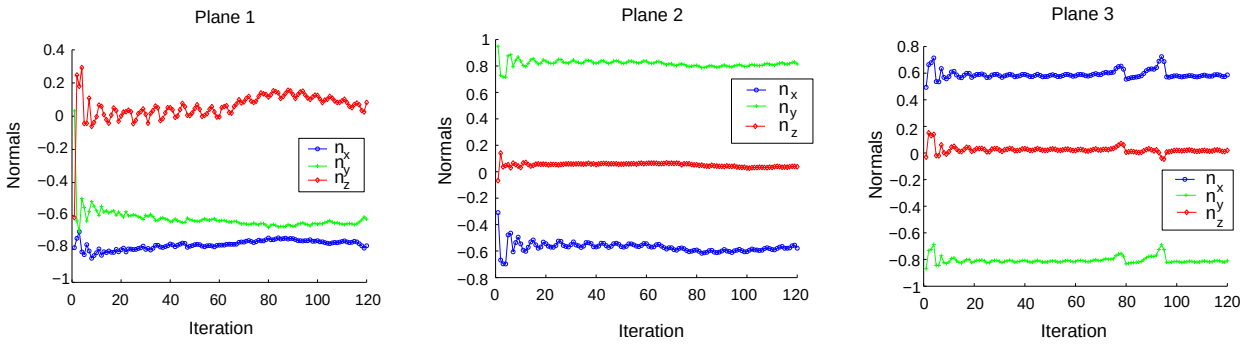


Figure 6.25: Normals estimated for planes 1 to 3 (**MPT_ESM**)

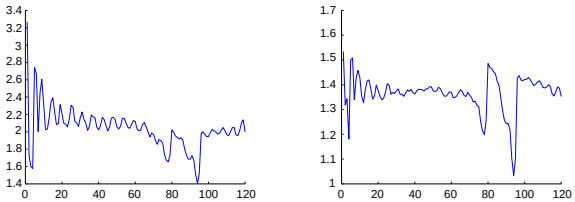


Figure 6.26: Estimation of the plane distances for planes 1 and 2 (**MPT_ESM**)

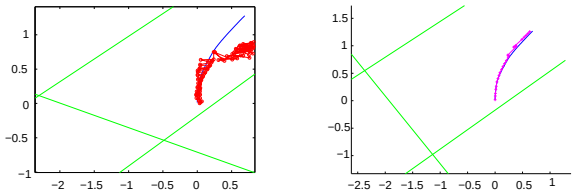


Figure 6.27: Robot's motion in the XY-plane for **SPT** and **MPT_ESM**

6.6 Outlier rejection

In the experiments of the previous section, occlusion and specularities diminished the performance of the algorithm in terms of speed and quality of the estimates. We will now investigate the problem of robustness.

A standard technique to improve least-square estimates is to use a robust cost function such as Tukey or Huber (see Appendix in [Hartley and Zisserman, 2000]). The underlying idea is that the errors should follow a given profile (eg. Gaussian distribution) and that all the points that do not verify the distribution are considered as outliers. Outliers are either rejected altogether (Tukey) or given small weights (Huber). Rejecting outliers is preferable for computational reasons. Estimating the Gaussian profile of the error is typically done with a robust mean (median) and a robust standard deviation (mean absolute deviation or MAD). Applying this technique as such to our tracking problem means we are considering each pixel as an independent value. However the error is generally spatially correlated. This observation was used in [Ishikawa et al., 2002] to devise a robust approach to tracking. The idea is to split the image in blocks and calculate an error that takes into account the standard deviation of the initial block (we expect they will be higher errors in blocks with a lot of texture).

Ishikawa *et al* [Ishikawa et al., 2002] use the following error for a block B_i:

$$\mathbf{e}_i = \frac{F(\mathbf{x})_{\mathbf{p}\in B_i}}{\sigma[\mathcal{I}^*(\mathbf{p})]_{\mathbf{p}\in B_i}}$$

They then order the errors and keep a pre-defined percentage of blocks. Choosing a pre-defined value is not satisfactory as when there are few outliers useful information will be discarded and if there is a lot of noise outliers will be kept. We propose to simply use robust statistics on the blocks and thus avoid this pre-defined value. The choice of the block size is still arbitrary.

We experimented the algorithm on the same sequence than for the validation of the tracking of multiple planes. We chose a block size of 10×10 pixels. Figures 6.30, 6.32 and 6.34 show the estimates of the translation for the ESM algorithm using respectively no robust techniques, the pixel-based Tukey robust function and the block-based Tukey robust function. Figures 6.31, 6.33 and 6.35 correspond to the estimates of the rotation and also depict the number of iterations needed to converge. Figure 6.28 shows the final 3D reconstruction and motion estimation of the mobile robot.

On this example, the pixel-based robust approach does not improve the quality of the estimates and increases the number of iterations drastically. This can be explained by the error function that only takes into account the error per pixel. The error is typically high where we have strong gradient. However by removing these regions we also remove the information that enables the minimisation to converge. By using blocks, the error is weighted by the gradient. We obtain improvements in terms of precision and also computational time (the number of iterations stays within 4 to 11 iterations). Figure 6.29 shows the specularity "being removed" by the blocks. Each column corresponds to one of the templates being tracked. The first line shows the reprojection error of the template. The second line shows the block weights being automatically assigned. The lighter the block, the stronger the penalisation weight. The last line shows the templates weighted by the block values.

These results could be improved further by using a distribution that is better adapted to $\mathbf{e}_i$.

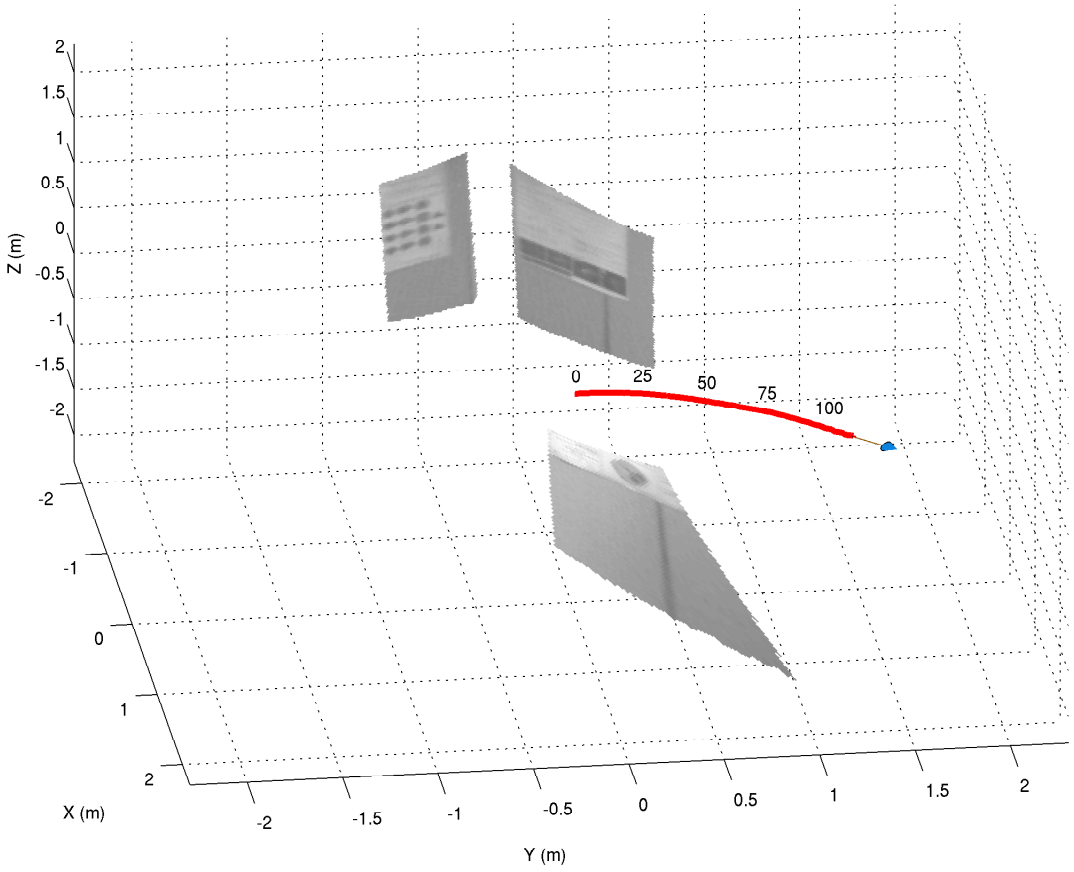


Figure 6.28: 3D reconstruction and 6-DOF motion estimation

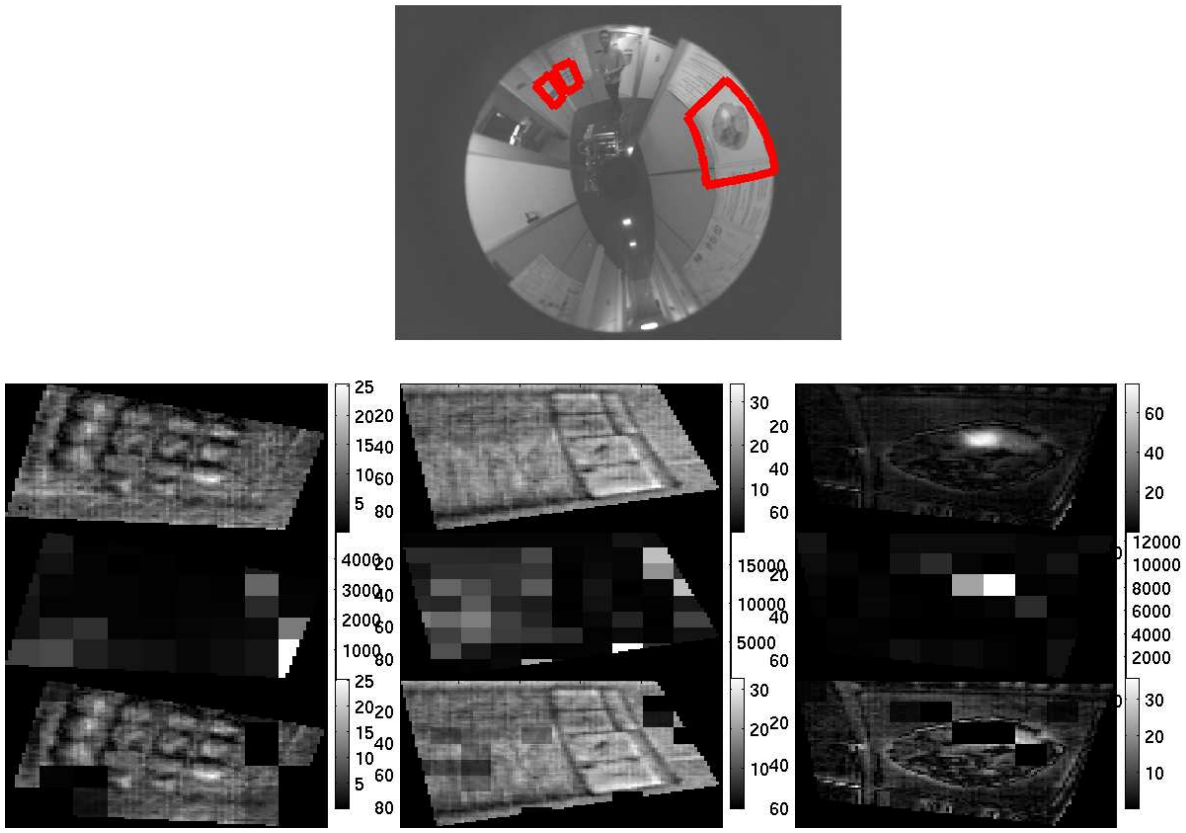


Figure 6.29: Specularities being removed by block-based robust technique

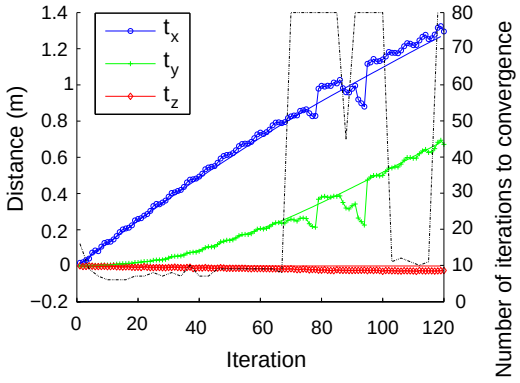


Figure 6.30: Translation estimate without a robust function

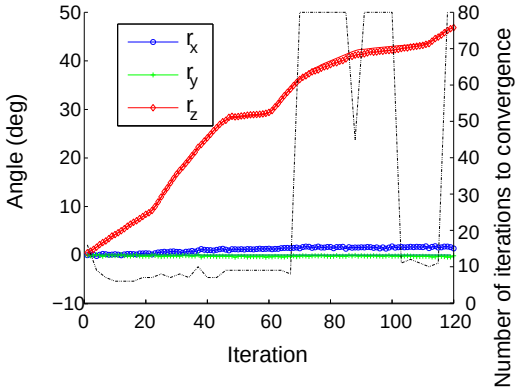


Figure 6.31: Rotation estimate without a robust function

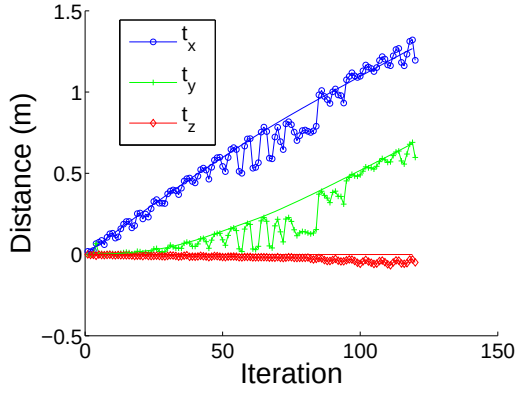


Figure 6.32: Translation estimate using pixel-based robust function

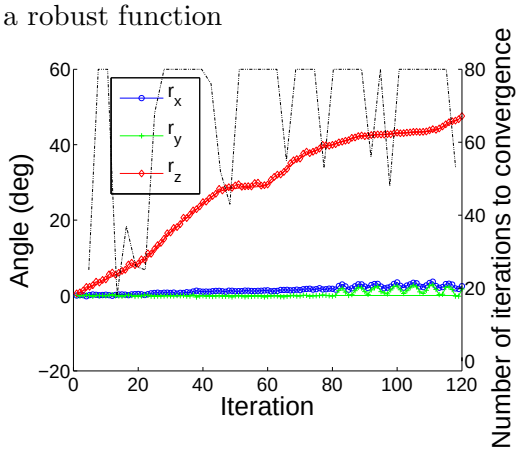


Figure 6.33: Rotation estimate using pixel-based robust function

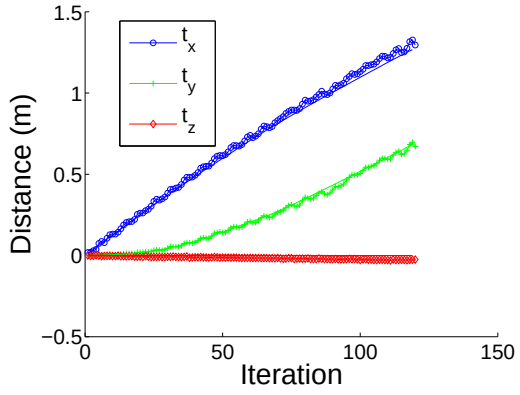


Figure 6.34: Translation estimate using block-based robust function

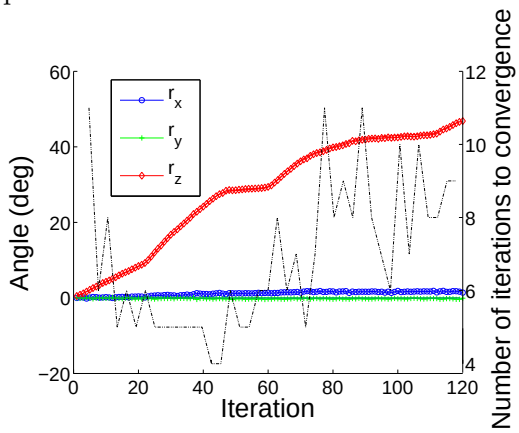


Figure 6.35: Rotation estimate using block-based robust function

6.7 Conclusion

In this chapter, we have presented an efficient parametric visual tracking algorithm that enables 6 DOF motion estimation for any single viewpoint sensor. The method relies on the Efficient Second-order Minimisation approach (ESM) that has better convergence properties to the now standard inverse compositional. We presented several variants to the ESM (αESM and iESM) with better computational properties. In view of its integration in applications for mobile robot motion estimation in potentially cluttered environment, a simple outlier rejection algorithm was also developed.

Chapter 7

Omnidirectional line images

Contents

7.1 Introduction

In the previous chapter, we discussed efficient ways of tracking planes for motion and structure estimation. However planes are not always available in the environment. Figure 7.1 shows an example of an indoor environment where there are not enough points or textured planes for structure and motion. This motivates the use of lines which is the topic of this chapter.

In the article by Devernay and Faugeras [Devernay and Faugeras, 2001] "Straight lines have to be straight" the authors devise a calibration approach based on this tautology. In the planar perspective projection model, distortion is seen as "fault" that should be removed. However we can alternatively consider distortion as simply part of the imaging process like the focal length or the principal point. In the case of omnidirectional sensors, distortion is in fact the property that enables a wide angle of view.

As explained in Chapter 6 it is desirable to work directly in omnidirectional images without unwarping the image to a perspective view. This means however that we cannot parameterise lines with for example the standard (ρ, θ) parameters. In this chapter, we will re-explore line images and introduce ways of working in omnidirectional images.

In the past, lines have been used extensively with panoramic cameras for the motion estimation and localisation of mobile robots [Yagi and Yachida, 1991; Delahoche et al., 1997] but generally ([Bosse et al., 2002] being an exception) under the assumption that the lines were radially projected in the device. This of course limits the use of the sensor to environments with sufficient vertical lines and imposes the sensor to be in a vertical position. In this chapter we aim at generalising the use of lines. Work has been done previously for line extraction [Yamazawa and Yachida, 2000; Barreto and Araujo, 2003; Vasseur and Mouaddib, 2004]. Our contribution is to generalise the approach to all central catadioptric sensors from a projective geometry perspective which leads, as we will see, to simple and efficient algorithms. To our knowledge, current line tracking for omnidirectional vision has only been done using quadric approaches such as in [Barreto et al., 2002] or vanishing points [Bosse et al., 2002]. We will see how we can use a parametric approach instead using the minimal amount of parameters.

Structure from motion from lines has been thoroughly studied in the past for normal perspective sensors [Hartley and Zisserman, 2000]. More recently, [Taylor and Kriegman, 1995] and [Bartoli and Sturm, 2005] analyse the non-linear structure from motion equations. In the context of non-linear minimisation, it is desirable to parameterise the problem using the minimum amount of parameters: the minimisation is faster, less subject to noise and consistency constraints can be directly imposed (eg. a rotation matrix must stay a rotation matrix after minimisation). In [Bartoli and Sturm, 2005], the authors introduce an orthonormal representation for Plücker coordinates to minimise only the 4 parameters representing a line. They give references to possible methods to obtain a minimal representation of the transformation but do not give specific details. In this chapter, we put forth the group structure of the line motion matrix [Bartoli and Sturm, 2004] that enables the use of Lie algebras for a minimal parameterisation as described in Section 5.1.5. We detail the specific case of calibrated cameras that applies to central catadioptric sensors. Different point-line distances are proposed.

We will call line images the projection of 3D lines in the image plane (or the normalised plane according to context).

After a short insight into the relationship between calibration and line images, the chapter is devided in three distinct parts. Each part is an essential component of a fully automatic motion estimation algorithm:

- Section 7.3 shows how the projection model can be re-written to use the properties from *projective geometry* which leads to a linear estimation of the line images and an efficient algorithm for their

extraction using Hough transforms.

- Section 7.4 is dedicated to the problem of tracking lines between images. In particular we will detail how to parameterise the line images to avoid singularities.

- Section 7.5 concerns the minimisation step to find the camera motion and the 3D line positions. As explained in Chapter 5, it is desirable to use a minimal parameterisation. We will thus focalise on group structures and their associated Lie algebras.

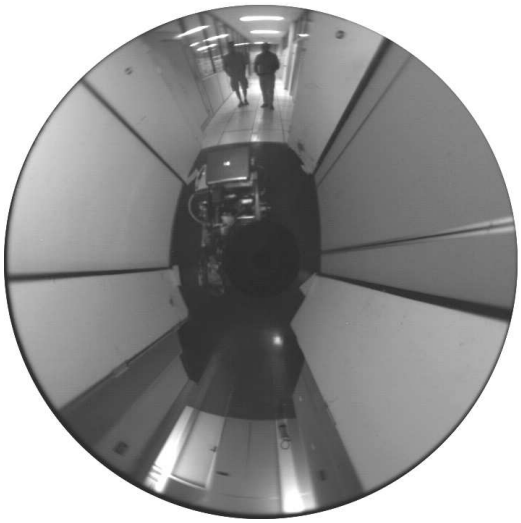


Figure 7.1: Example of an image of a difficult sequence for point or plane SFM

7.2 Relationship between line images and calibration

In [Geyer and Daniilidis, 2001], Geyer and Daniilidis study the relationship between line images and calibration. If we consider the **UPM** (see Section 2.2.2.2), line images are circles in paracatadioptric images and more general conics in the hyperbolic case. However line images only depend on two parameters (figure 7.2). Under the assumption that the unknown calibration parameters consist only of three unknowns, the focal length and the coordinates of the principal point, this implies we can calibrate the sensor with 2 line images in the parabolic case and 3 line images in the hyperbolic case.

This situation is very different to normal perspective cameras where line images are lines and do not add any constraints on the calibration parameters.

Working in with an "uncalibrated" omnidirectional sensor by extracting the conics does not have the same sense as with a perspective cameras as we could recover the calibration parameters and then work on the curves with only two degrees of freedom adding robustness to the approach. This is the underlying idea of this chapter: line images in omnidirectional sensors can be (or should be?) seen as two-dimensional curves.

7.3 Line images as projection of planes

A 3D line projected in a monocular imaging device can be parameterised by the normal noted $\mathbf{n}$ ($\mathbf{n} \in S^2$) formed by the line and the center of projection (figure 7.2).

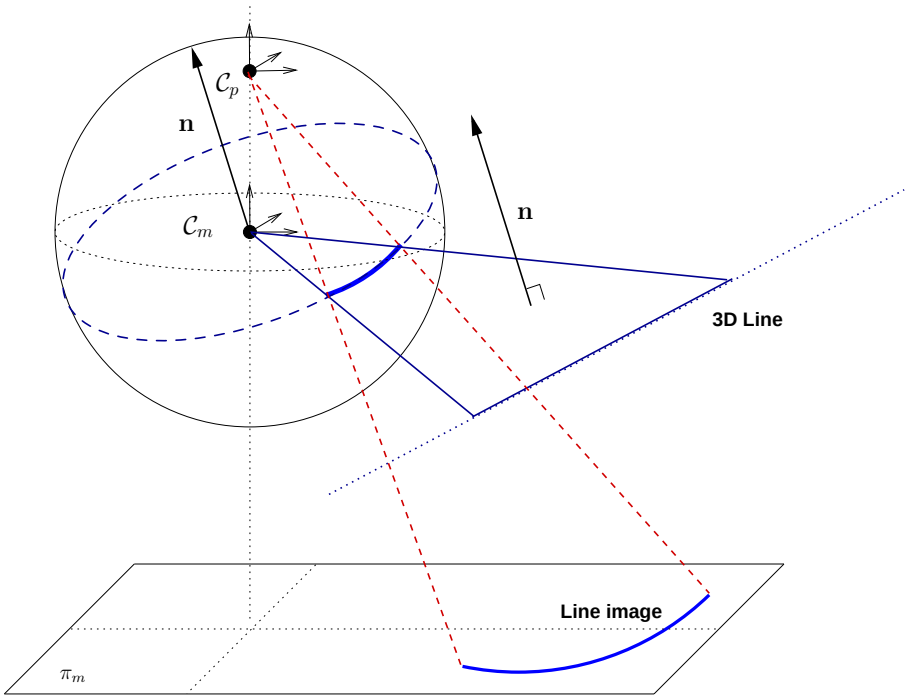


Figure 7.2: Closest point to a great circle on the sphere

The projection function Π relates a point $\boldsymbol{\mathcal{X}}_s$ on the sphere to a point $\mathbf{p}$ in the image plane. Thus, we have the following projective property: a point $\mathbf{p}$ on an line image of parameter $\mathbf{n}$ verifies:

$$\mathbf{n}^\top \Pi^{-1}(\mathbf{p}) = 0 \tag{7.1}$$

In the case where we do not need to consider the distortion induced by the lens (**UPM**), the situation is simpler. Equation (2.2), rewritten here, relates a point on the normalised plane to a projective ray through the mirror center and $\boldsymbol{\mathcal{X}}_s$:

$$\begin{cases} \hbar^{-1}(\mathbf{m}) \sim \begin{bmatrix} x \\ y \\ f(x,y) \end{bmatrix} \\ f(x,y) = 1 + \xi \frac{x^2+y^2+1}{-\xi - \sqrt{1+(1-\xi^2)(x^2+y^2)}} \end{cases}$$

This equation is valid for any central catadioptric device without lens distortion. When the sensor is calibrated the values for $f(x,y)$ can be pre-calculated and stored in a single look-up table of the size of the image to improve the efficiency of the lifting of the points. The relation between $\mathbf{m}$ and $\mathbf{p}$ is linear and not very costly to compute (in particular if $r = 1$ and $s = 0$ which is often the case with modern cameras). In the more general case (**CPM**), we would need to save each lifting of an image point which requires at least two buffers of the size of the image.

7.3.1 Line estimation

Let $\Pi^{-1}(\mathbf{p}) = \boldsymbol{\mathcal{X}}_s = [x_s\ y_s\ z_s]^\top$, if we write (7.1) for n points, we obtain:

$$\begin{bmatrix} x_{s_1} & y_{s_1} & z_{s_1} \\ x_{s_2} & y_{s_2} & z_{s_2} \\ \vdots & \vdots & \vdots \\ x_{s_n} & y_{s_n} & z_{s_n} \end{bmatrix} \mathbf{n} = \mathbf{A}\mathbf{n} = 0 \tag{7.2}$$

If we consider the singular value decomposition of $\mathbf{A}$, $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$ and order the eigenvalues of $\mathbf{S}$ in decreasing order, the normalisation of the third column of $\mathbf{V}$ will correspond to the least squares solution to (7.2). In the case where the focal length is unknown, that there is no distortion and the mirror is a parabola, the work by Barreto [Barreto and Araujo, 2003] can be used or equation (3.3).

7.3.2 Line extraction with the classic Hough transform

Let Φ be the colatitude and Θ the azimuthal angle, the normal can be written in spherical coordinates as:

$$\begin{cases} n_x = \sin\Phi\cos\Theta \\ n_y = \sin\Phi\sin\Theta \\ n_z = \cos\Phi \end{cases}$$

if we assume that $n_z \neq 1$ (ie. $\Phi \neq 0[\pi]$) and note $z = f(x, y)$, from (7.1) we obtain:

$$\Phi = \operatorname{atan}\left(\frac{x\cos(\Theta) + y\sin(\Theta)}{z}\right) \tag{7.3}$$

This result was proposed previously in [Vasseur and Mouaddib, 2004]. We discuss a way to adapt the line extraction to the non-uniform pixel resolution in Section 7.3.4. It may also be noted that values for atan cannot be pre-calculate (because the input space $\mathbb{R}$ is not bounded). To improve the efficiency, a Hough space can be built using directly $\tan(\Phi)$ with for example a "linked-list" to represent the Hough space (see [Xu and Oja, 1993] for details of different structures related to Hough parameter spaces).

7.3.3 Line extraction with the randomized Hough transform

The Randomized Hough transform (RHT) [Xu and Oja, 1993] has proved to be an efficient and robust alternative to classic Hough. It shares *convergence mapping* with RANSAC, meaning that we estimate the n parameters of the curve function ($n = 2$ for a line) by randomly extracting n values.

For estimating the parameters of a line, the authors in [Xu and Oja, 1993] extend the (ρ, θ) parameterisation from Duda and Hart. A natural and more efficient parameterisation can be obtained by directly estimating the normal: the line image joining two points $\mathbf{m}_1$ and $\mathbf{m}_2$ has for normal $\mathbf{n}$ ($\hbar^{-1}(\mathbf{m}) \in S^2$):

$$\mathbf{n} = \hbar^{-1}(\mathbf{m}_1) \times \hbar^{-1}(\mathbf{m}_2) \tag{7.4}$$

By imposing for example $n_z \geq 0$, we obtain a 2-dimensional buffer in (n_x, n_y).

7.3.4 Voting in the Hough space

The solid angle subtended by the surface represented by a pixel can be used to take into account the non-uniform resolution. A pixel $\mathbf{p}$ is bounded by $(u-0.5, v-0.5)$, $(u+0.5, v-0.5)$, $(u-0.5, v+0.5)$ and $(u+0.5, v+0.5)$. The corresponding surface on the unit sphere is then bounded by (Φ_{min}, Φ_{max}) and $(\Theta_{min}, \Theta_{max})$ (obtained through the lifting of the points) and corresponds to the following solid angle s (measured in steradians sr):

$$s = \int_{\Theta_{min}}^{\Theta_{max}} \int_{\Phi_{min}}^{\Phi_{max}} \sin(\Phi) d\Theta d\Phi \quad (7.5)$$

$$= -(\Theta_{max} - \Theta_{min})(\cos(\Phi_{max}) - \cos(\Phi_{min})) \quad (7.6)$$

The precision of the normal estimate will be inversely proportional to the subtended solid angle so $1/s$ can be used in the classic Hough voting scheme. (In the randomized case, we assumed that on average the surface subtended by two pixels was the same.)

7.4 Line tracking

Tracking is a important step for structure and motion or visual servoing [Mezouar et al., 2004]. It is essential to use a minimal representation to ensure robustness. Tracking a line $\mathcal{L}$ between two views can be done using classic edge-tracking approaches [Bouthemy, 1989; Smith et al., 2004; Marchand and Chaumette, 2005] in the following steps:

1. obtain n points on $\mathcal{L}$ uniformly distributed in the image,
2. for each point calculate the normal to the edge,
3. search (within pre-defined bounds) along the direction given by the normal to the curve for edge points with same normals (using pre-calculated convolution kernels),
4. robustly extract the equation of the new line from the edge points.

For 3) the preferred method is the Bresenham algorithm [Bresenham, 1965]. For 4), M-estimators are often chosen to extract the parameters but in presence of a lot of noise, a RANSAC is a good alternative (this is relatively fast as the model is simple to fit using (7.4) and the size of the data is small).

We will now derive a parametric equation for line images and calculate the normal in a given point. We will see that using a conic parametric function leads to singularities (but gives information on the nature of the conic). We will then propose a non-singular parametric function.

7.4.1 Conic parametric function

Equation (7.4) can be re-written to obtain a quadric form in the normalised plane [Geyer and Daniilidis, 2001; Barreto, 2003]: $\mathbf{m}^\top \mathbf{\Omega}_n \mathbf{m}$ with:

$$\mathbf{\Omega}_n = \begin{bmatrix} n_x^2(1-\xi^2) - n_z^2\xi^2 & n_x n_y(1-\xi^2) & n_x n_z \\ n_x n_y(1-\xi^2) & n_y^2(1-\xi^2) - n_z^2\xi^2 & n_y n_z \\ n_x n_z & n_y n_z & n_z^2 \end{bmatrix} \quad (7.7)$$

$\det(\widehat{\mathbf{\Omega}}_n) = \xi^4 n_z^4 (n_x^2 + n_y^2 + n_z^2) = (\xi n_z)^4$. For $\widehat{\mathbf{\Omega}}_n$ to be a proper conic, we will from now on assume that $\xi \neq 0$ (non-planar/perspective mirror) and $n_z \neq 0$.

The nature of the conic depends on the number of intersections with the line at infinity i.e. the sign of $\Delta = 1 - \xi^2 - n_z^2$ (we removed $n_z^2 \xi^2 > 0$). $\Delta > 0$ corresponds to a hyperbola, $\Delta = 0$ to a parabola and $\Delta < 0$ to an ellipse.

From the *Joachimsthal* equations, we obtain the four focal points (2 real and 2 complex) [Semple and Kneebone, 1979]. The two real values are:

$$\mathbf{f}_1 = \begin{bmatrix} n_x \\ n_y \\ n_z + \sqrt{1-\xi^2} \end{bmatrix} \quad \mathbf{f}_2 = \begin{bmatrix} n_x \\ n_y \\ n_z - \sqrt{1-\xi^2} \end{bmatrix} \tag{7.8}$$

(We can note that $\mathbf{f}_2$ is at infinity if the conic is a parabola.) If we now center the conic in $\mathbf{f}_1$ and rotate it (if $n_z \neq 1$) by an angle Θ, we obtain ($\Delta \geq -1$):

$$\mathbf{\Omega}'_m = \begin{bmatrix} \Delta & 0 & \sqrt{(1-n_z^2)(1-\xi^2)} \\ 0 & -\xi^2 n_z^2 & 0 \\ \sqrt{(1-n_z^2)(1-\xi^2)} & 0 & 1 \end{bmatrix} \tag{7.9}$$

Let $x = \rho\cos(\theta)$ and $y = \rho\sin(\theta)$, the polar equation of the line image centered in $\mathbf{f}_1$, valid for $n_z \neq 0$ and $\xi \neq 0$, is:

$$\rho = \frac{1}{\xi n_z - \sqrt{(1-n_z^2)(1-\xi^2)}\cos(\theta - \Theta)} \tag{7.10}$$

For $n_z = 0$, the conic is a straight line that goes through the origin and is parameterised by (ρ, Θ).

When $n_z \to 0$, we get closer to a degenerate conic as $\mathbf{f}_1 \to 0$. This means we will not be able to represent and sample curves when $n_z \to 0$ using the angle θ.

The impossibility to represent line images with a single model makes the representation inadequate for line tracking. Using several representations would require an arbitrary switching mechanism. In the following section, we will detail how to obtain a non-singular representation.

7.4.2 Unified non-singular parametric function

Let $B = \{\boldsymbol{\mathcal{X}}_s = (X_s, Y_s, \xi) | \boldsymbol{\mathcal{X}}_s \in S^2\}$. B is the natural boundary between the two sheets of S^2 covering $\mathbb{P}^2$ through the unified projection.

Let $\mathcal{C}$ be the arc of the great circle corresponding to $\mathcal{L}$ and parameterised by $\mathbf{n}$. $\mathbf{n}$ can be seen as an axis of rotation for the points of $\mathcal{C}$ on the sphere (figure 7.5). Let $\mathbf{w}$ be one of these points ($\mathbf{n}^\top \mathbf{w} = 0$). $\mathcal{C}$ can be parameterised with an angle θ without a singularity using Rodrigues' formula:

$$\mathbf{w}(\theta) = e^{[\mathbf{n}]_\times \theta} \mathbf{w} \tag{7.11}$$

We do not obtain a singularity because a finite 3D line spans an angle strictly inferior to π.

$\mathbf{w}(\theta)$ is a point on the sphere so its projection $\mathbf{m}(\theta)$ (defined for $\mathbf{m}(\theta) \notin B$) on the normalised plane is simply:

$$\mathbf{m}(\theta) = \frac{1}{\mathbf{w}_z(\theta) - \xi} \begin{bmatrix} \mathbf{w}_x(\theta) \\ \mathbf{w}_y(\theta) \end{bmatrix} \tag{7.12}$$

7.4.3 Curve sampling

Let s be the arc length of the parametric curve (7.12) between two points $\mathbf{m}(\theta_1)$ and $\mathbf{m}(\theta_2)$ with $\theta_2 > \theta_1$. We will make the assumption that we have a similarity between the normalised plane and the image plane (i.e. $r \approx 1$ and $s \approx 0$). (In other words a uniform sampling in the normalised plane corresponds to a uniform sampling in the image plane.)

If we wish to sample the curve in n values (to guarantee constant time), the increment arc length is $\delta s = \frac{s}{n}$ with $s = \int_{\theta_1}^{\theta_2} ds$. If we wish to obtain values separated by p pixels, the increment is $\delta s = \frac{p}{\gamma}$.

In the general case, the calculation of arc lengths for conics involves elliptic integrals of the second kind, so we cannot obtain a simple formulation for s. We may note that in the case of a paraboloid mirror, the conic is simply a circle which can be uniformly sampled by an angular increment. However we loose this property with the non-singular representation.

For an approximate calculation of s, we may use the differential form of the curve length and a small increment for θ:

$$ds = \sqrt{dx'^2 + dy'^2}d\theta \tag{7.13}$$

$$\text{with : } \begin{cases} dx' &= \frac{\mathbf{w}'_x(\theta)(\mathbf{w}_z(\theta)-\xi)-\mathbf{w}_x\mathbf{w}'_z(\theta)}{(\mathbf{w}_z(\theta)-\xi)^2} \\ dy' &= \frac{\mathbf{w}'_y(\theta)(\mathbf{w}_z(\theta)-\xi)-\mathbf{w}_y\mathbf{w}'_z(\theta)}{(\mathbf{w}_z(\theta)-\xi)^2} \\ \mathbf{w}'(\theta) &= e^{[\mathbf{n}]_\times\theta}[\mathbf{n}]_\times\mathbf{w} \end{cases} \tag{7.14}$$

Figures 7.3 and 7.4 illustrate the importance of adapting the line sampling to the curve being tracked. In figure 7.3 we can see that the projection of a uniform sampling on the sphere can lead to a poorly sampled line image. Tracking with such a sampling could introduce a bias and will probably fail if the over-sampled region is for example occluded. The approach proposed in this section can ensure that the curve is correctly sampled in the image as shown in figure 7.4.

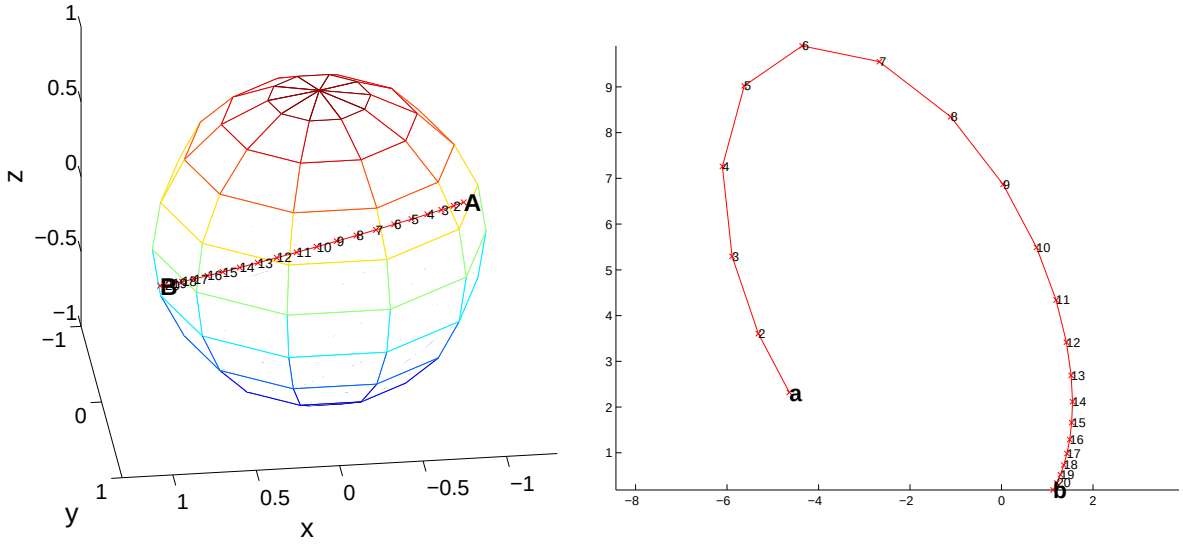


Figure 7.3: Uniform sampling of a great circle on the sphere with corresponding projection in the image

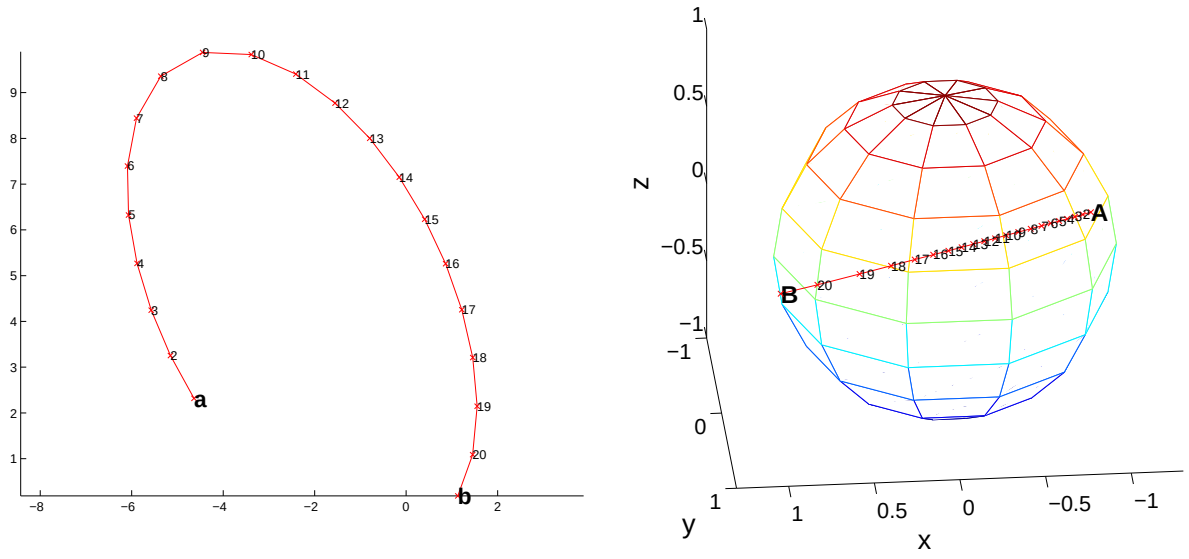


Figure 7.4: Uniform sampling of a line image with corresponding points on the sphere

7.4.4 Normal to a line image

The angle ϕ of the normal in $\mathbf{m}(\theta)$ is simply:

$$\phi = \begin{cases} \text{atan } (-\frac{dx'}{dy'}) & \text{if } dy' \neq 0 \\ \frac{\pi}{2} & \text{if } dy' = 0 \end{cases} \tag{7.15}$$

7.5 Structure from motion

Points and templates are generally chosen for motion estimation in robotic application because they offer robust and accurate results. However in cases of low-textured environments, lines can play a key role to improve estimates and provide a partial 3D reconstruction.

In real-time robotic applications, iterative approaches are generally preferred to batch algorithms as they are often faster. However in the case of lines that have been automatically extracted, the motion and 3D line estimates are generally not well constrained and sensitive to the 3D position of the lines. On the other hand, the minimisation is not computationally expensive which encourages a two step approach: 1) bundle adjustment 2) global filtering (eg. Extended Kalman Filter) when the covariance enables confidence to be put into the line and motion estimates. In this chapter, we focus on bundle adjustment.

We will start by describing the representation chosen for 3D lines and in particular the minimal orthonormal representation proposed by Bartoli and Sturm [Bartoli and Sturm, 2005]. We will then describe how 3D lines can be transferred between views through a tranformation matrix called the line motion matrix [Bartoli and Sturm, 2004]. These matrices form a Lie group which enables us to use the techniques described in Chapter 5 to ensure a minimal representation of the transformation. Finally we will define different possible distances between lines based on their end points and define the associated least-squares problem.

7.5.1 Line representation

We will represent 3D lines by the following Plücker coordinates: $\mathcal{L}^\top \sim \begin{bmatrix} \mathbf{n}^\top & \mathbf{v}^\top \end{bmatrix}$. A detailed analysis of Plücker coordinates and other line representations can be found in [Hartley and Zisserman, 2000] or [Andreff et al., 2002], dedicated more specifically to visual servoing.

Plücker coordinates are defined up to a scale factor. We choose to normalise the first component to simplify the equations on the sphere. $\mathbf{n}$ is defined as previously in the chapter and $\mathbf{v}$ is the direction of the 3D line. In order to obtain a valid line representation, the constraint $\mathbf{n}^\top \mathbf{v} = 0$ must be imposed.

Minimal representation of lines

We will use the method proposed by Bartoli and Sturm [Bartoli and Sturm, 2005] to obtain a minimal representation (4 parameters) of lines from their Plücker coordinates through an orthonormal representation.

To summarise, we can write the Plücker line representation as:

$$\begin{bmatrix} \mathbf{n} & \mathbf{v} \end{bmatrix}_{3\times 2} = \underbrace{\begin{bmatrix} \frac{\mathbf{n}}{\|\mathbf{n}\|} & \frac{\mathbf{v}}{\|\mathbf{v}\|} & \frac{\mathbf{n}\times\mathbf{v}}{\|\mathbf{n}\times\mathbf{v}\|} \end{bmatrix}}_{\mathbb{SO}(3)} \underbrace{\begin{bmatrix} \|\mathbf{n}\| & 0 \\ 0 & \|\mathbf{v}\| \\ 0 & 0 \end{bmatrix}_{3\times 2}}_{\text{one parameter space eg. } \mathbb{SO}(2)}$$

A way of obtaining this decomposition given a Plücker line representation is to use the QR "orthogonal/upper triangular" decomposition:

$$\begin{bmatrix} \mathbf{n} & \mathbf{v} \end{bmatrix}_{3\times 2} \overset{QR}{=} \mathbf{U}_{3\times 3} \begin{bmatrix} \sigma_1 & \\ & \sigma_2 \\ & \end{bmatrix}_{3\times 2}, \mathbf{W} = \frac{1}{\|\sigma\|} \begin{bmatrix} \sigma_1 & -\sigma_2 \\ \sigma_2 & \sigma_1 \end{bmatrix}$$

with $(\mathbf{U}, \mathbf{W}) \in \mathbb{SO}(3) \times \mathbb{SO}(2)$.

The other way round, let $\mathbf{x}_L$ contain the 3+1 parameters representing the matrices, we can recover the Plücker coordinates through (with $\mathbf{u}_i$ the $i-$th column of $\mathbf{U}$):

$$\mathcal{L}(\mathbf{U}(\mathbf{x}_L), \mathbf{W}(\mathbf{x}_L))^\top \rightarrow \begin{bmatrix} \mathbf{u}_1^\top & \frac{w_{21}}{w_{11}}\mathbf{u}_2^\top \end{bmatrix}$$

7.5.2 Line motion matrix

In [Bartoli and Sturm, 2004], the authors define the 6×6 matrices that act on Plücker coordinates in projective, affine and Euclidean spaces. These matrices were named "line motion matrices" as they transfer Plücker coordinate representations of 3D lines between views.

We can prove that we have in fact a group homomorphism between the transformation groups and the line motion matrix spaces for the matrix product. This result is important as it indicates that we can obtain a minimal representation through the associated Lie algebras and recover the transformations directly. We will detail the Euclidean case which is of interest for this study.

For calibrated central catadioptric cameras, the transformation between 3D lines in two views can be represented by a rotation matrix $\mathbf{R} \in \mathbb{SO}(3)$ and a translation $\mathbf{t} \in \mathbb{R}^3$ by:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & [\mathbf{t}]_\times \mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \tag{7.16}$$

We will call $\mathbb{LE}(3)$ the group formed of matrices of the previous type and $\mathfrak{le}(3)$ its associated Lie algebra. Let $\mathbf{A}_i$, with $i \in \{1, 2, ..., 6\}$, be a basis of $\mathfrak{le}(3)$. Any matrix $\mathbf{A} \in \mathfrak{le}(3)$ can be written as a linear combination of the matrices $\mathbf{A}_i$.

Let the (3×1) vectors $\mathbf{b}_x = (1, 0, 0)$, $\mathbf{b}_y = (0, 1, 0)$ and $\mathbf{b}_z = (0, 0, 1)$ be the natural orthonormal basis of $\mathbb{R}^3$. The $\mathbf{A}_i$ matrices are of dimension (6×6). The generators for the translation are:

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{0} & [\mathbf{b}_x]_\times \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} \mathbf{0} & [\mathbf{b}_y]_\times \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} \mathbf{0} & [\mathbf{b}_z]_\times \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{7.17}$$

The generators for the rotation are:

$$\mathbf{A}_4 = \begin{bmatrix} [\mathbf{b}_x]_\times & \mathbf{0} \\ \mathbf{0} & [\mathbf{b}_x]_\times \end{bmatrix}, \mathbf{A}_5 = \begin{bmatrix} [\mathbf{b}_y]_\times & \mathbf{0} \\ \mathbf{0} & [\mathbf{b}_y]_\times \end{bmatrix}, \mathbf{A}_6 = \begin{bmatrix} [\mathbf{b}_z]_\times & \mathbf{0} \\ \mathbf{0} & [\mathbf{b}_z]_\times \end{bmatrix} \tag{7.18}$$

The exponential map links the Lie algebra to the Lie Group. $\mathbf{T}$ can be locally parameterized as, with $\mathbf{x}_T = (x_1, x_2, ..., x_6)$:

$$\mathbf{T}(\mathbf{x}_T) = \exp\left(\sum_{i=1}^{6} x_i \mathbf{A}_i\right) \tag{7.19}$$

Thanks to the group homomorphism we can recover the Euclidean transformation $\mathbf{T}_e$ directly from $\mathbf{x}_T$ through the 6 generators $\mathbf{B}_i$ of $\mathbb{SE}(3)$ (detailed in Section 5.3):

$$\mathbf{T}_e(\mathbf{x}_T) = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} = \exp\left(\sum_{i=1}^{6} x_i \mathbf{B}_i\right)$$

The Euclidean transformation can then be used for example in the control loop for a visual servoing application or for combining the minimisation with other features like 3D points.

We have now described how to represent minimally 3D lines and the associated tranformation matrices. The final step consists in defining an error function that will link the reprojection error of lines to the motion and 3D position of the features.

7.5.3 Distance functions

In our framework, 3D lines will be represented by their normals $\mathbf{n}$ in a given view (figure 7.5). The tracked line images are represented by their end points. We will thus need to define a distance between an end point and a 3D line represented by its normal.

Several distance functions between a point $\boldsymbol{\mathcal{X}}_s$ and a line parameterised by $\mathbf{n}$ can be considered in the case of the sphere:

$$\begin{cases} d_A(\boldsymbol{\mathcal{X}}_s, \mathbf{n}) &= \boldsymbol{\mathcal{X}}_s^\top \mathbf{n} \\ d_R(\boldsymbol{\mathcal{X}}_s, \mathbf{n}) &= \arccos(\sqrt{1 - (\boldsymbol{\mathcal{X}}_s^\top \mathbf{n})^2}) \\ d_r(\boldsymbol{\mathcal{X}}_s, \mathbf{n}) &= d_e(\Pi(\boldsymbol{\mathcal{X}}_s^\perp), \Pi(\boldsymbol{\mathcal{X}}_s)) \\ \text{with} & \quad \boldsymbol{\mathcal{X}}_s^\perp = \frac{\boldsymbol{\mathcal{X}}_s - (\boldsymbol{\mathcal{X}}_s^\top \mathbf{n})\mathbf{n}}{\sqrt{1-(\boldsymbol{\mathcal{X}}_s^\top \mathbf{n})^2}} \end{cases} \tag{7.20}$$

$\boldsymbol{\mathcal{X}}_s^\perp$ is the closest point to the line defined by $\mathbf{n}$ from $\boldsymbol{\mathcal{X}}_s$ (figure 7.5). d_A is an algebraic distance. d_R is the distance on the sphere (Riemann distance) between $\boldsymbol{\mathcal{X}}_s^\perp$ and $\boldsymbol{\mathcal{X}}_s$. d_r is the distance between $\boldsymbol{\mathcal{X}}_s$ and $\boldsymbol{\mathcal{X}}_s^\perp$ reprojected onto the normalised image plane. It corresponds to the standard distance between a point and a line in the perspective case if $n_z = 0$.

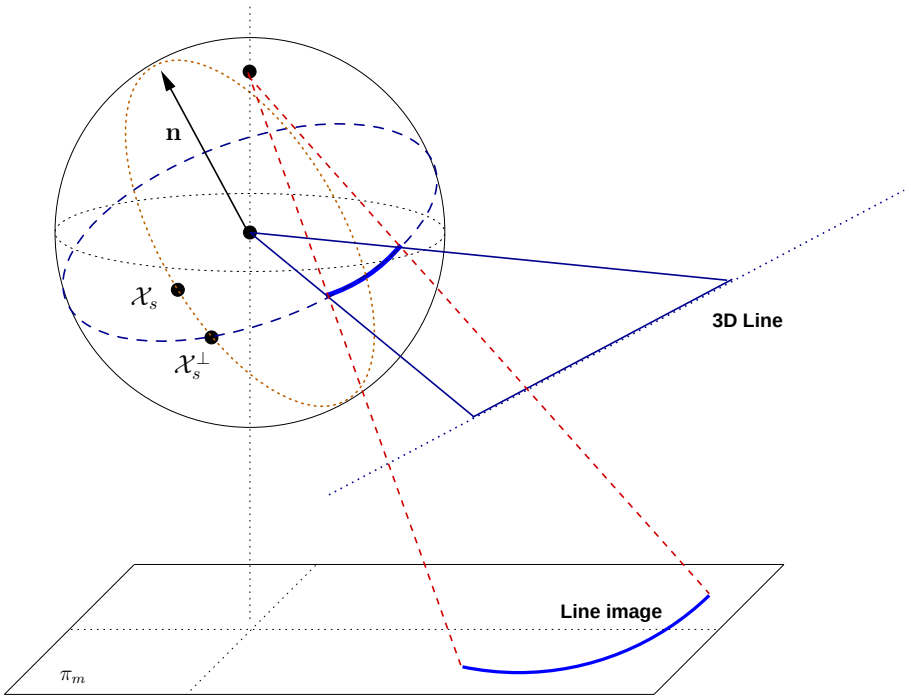


Figure 7.5: Closest point to a great circle on the sphere

7.5.4 Global cost function

Let $\mathbf{P}$ be the projection matrix for the 3D lines $\mathbf{P} = \left[\begin{array}{cc} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \end{array} \right]$. We will note $\mathbf{x}_T^i$ the minimal representation of the line motion matrix of the i-th view and $\mathbf{x}_T$ the vector containing all the motion parameters. $\mathbf{x}_L^j$ will indicate the orthonormal representation of the j-th line and $\mathbf{x}_L$ the vector containing all the line parameters. Similarly $\mathbf{x}_{TL}$ contains all the structure and motion values.

Given a line motion matrix representation $\mathbf{x}_T^i$ and a 3D line defined by $\mathbf{x}_L^j$, the normal representing the line image in the i-th view is:

$$\mathbf{n}_{ij} = \mathbf{P}\mathbf{T}(\mathbf{x}_T^i)\mathcal{L}(\mathbf{U}(\mathbf{x}_L^j), \mathbf{W}(\mathbf{x}_L^j))$$

For this reason we will now write the distances as functions of the minimal parameterisations.

Let $\boldsymbol{\mathcal{X}}_s^{ij}$ and be $\boldsymbol{\mathcal{Y}}_s^{ij}$ be the two endpoints of the j-th line in the i-th view. The cost function can be written as, with $d_.$ the chosen distance:

$$c_{ij}(\mathbf{x}_T^i, \mathbf{x}_L^j) = \left(d_.(\boldsymbol{\mathcal{X}}_s^{ij}, \mathbf{x}_T^i, \mathbf{x}_L^j)\right)^2 + \left(d_.(\boldsymbol{\mathcal{Y}}_s^{ij}, \mathbf{x}_T^i, \mathbf{x}_L^j)\right)^2 \tag{7.21}$$

We will now write our minimisation problem using increments as explained in Chapter 5. Let $\widehat{\mathbf{T}}^i$ be an approximation of the real transformation between the first and i-th view and $(\widehat{\mathbf{U}}^j, \widehat{\mathbf{W}}^j)$ an approximation of the j-th line parameters. In the case of the algebraic distance, for example, the problem is to find the incremental transformations $\mathbf{T}^i(\mathbf{x}_T)$, $\mathbf{U}^j(\mathbf{x}_L)$ and $\mathbf{W}^j(\mathbf{x}_L)$ such that the following value is minimised:

$$d_A(\boldsymbol{\mathcal{X}}_s^{ij}, \mathbf{x}_T^i, \mathbf{x}_L^j) = \boldsymbol{\mathcal{X}}_s^{ij\top}\mathbf{P}\widehat{\mathbf{T}}^i\mathbf{T}(\mathbf{x}_T^i)\mathcal{L}(\widehat{\mathbf{U}}^j\mathbf{U}(\mathbf{x}_L^j), \widehat{\mathbf{W}}^j\mathbf{W}(\mathbf{x}_L^j)) \tag{7.22}$$

The global cost function is then for m views and l lines (with $\mathbf{x}_{TL}$ the list of the parameters and $\mathbf{x}_T^1 = \mathbf{0}$) :

$$F(\mathbf{x}_{TL}) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{l} \|c_{ij}(\mathbf{x}_T^i, \mathbf{x}_L^j)\|^2 \tag{7.23}$$

The equation has $6(m-1)+4l$ unknowns. Each line in a given view adds 2 constraints. Therefore the minimal number of lines needed to constrain the system can be deduced from the formula (if we consider that each line is visible in each view):

$$l \geq \frac{6(m-1)}{2m-4} \tag{7.24}$$

Differentiation of the cost function

We will analyse more specifically the differentiation of the reprojection distance (the approach is very similar for the other distances).

$$\begin{cases} d_r(\boldsymbol{\mathcal{X}}_s, \mathbf{n}) = d_e(\Pi(\boldsymbol{\mathcal{X}}_s^\perp), \Pi(\boldsymbol{\mathcal{X}}_s)) \\ \boldsymbol{\mathcal{X}}_s^\perp = \frac{\boldsymbol{\mathcal{X}}_s - (\boldsymbol{\mathcal{X}}_s^\top \mathbf{n})\mathbf{n}}{\sqrt{1-(\boldsymbol{\mathcal{X}}_s^\top \mathbf{n})^2}} = s(\boldsymbol{\mathcal{X}}_s, \mathbf{n}) \end{cases} \tag{7.25}$$

where $\boldsymbol{\mathcal{X}}_s^\perp$ is the closest point to the line defined by $\mathbf{n}$ from $\boldsymbol{\mathcal{X}}_s$.

We will note $\mathbf{m}^\perp = \Pi(\boldsymbol{\mathcal{X}}_s^\perp)$ and $\mathbf{m} = \Pi(\boldsymbol{\mathcal{X}}_s)$.

The function we would like to differentiate with respect to $(\mathbf{x}_T, \mathbf{x}_L)$ is:

$$\begin{cases} d_r(\boldsymbol{\mathcal{X}}_s, \mathbf{x}_T, \mathbf{x}_L)^2 &= d_e(\Pi(s(\boldsymbol{\mathcal{X}}_s, S(\mathbf{n}))), \Pi(\boldsymbol{\mathcal{X}}_s))^2 \\ \mathbf{n} &= \mathbf{P}\widehat{\mathbf{T}}\mathbf{T}(\mathbf{x}_T)\mathcal{L}(\widehat{\mathbf{U}}\mathbf{U}(\mathbf{x}_L), \widehat{\mathbf{W}}\mathbf{W}(\mathbf{x}_L)) \end{cases} \tag{7.26}$$

with $S(\mathbf{n}) = \frac{\mathbf{n}}{\|\mathbf{n}\|}$.

We will derive the Jacobian of the cost function in the current image $\mathbf{x}_{TL} = \mathbf{x}_{TL}^0 = \mathbf{0}$ (the size of matrices appear in upperscript). Let l be the number of parameters and m the number of lines, $l = 6 + 3m$.

$$\frac{\partial d_r(\mathbf{x}_{TL})^2}{\partial \mathbf{x}_{TL}}\bigg|_{\mathbf{x}_{TL}=\mathbf{0}}^{1\times l} = 2d_r(\mathbf{x}_{TL}) \underbrace{\frac{\partial d_e(\mathbf{a},\mathbf{m})}{\partial \mathbf{a}}\bigg|_{\mathbf{a}=\widehat{\mathbf{m}}^\perp}^{1\times 2}}_{\mathbf{J}_{d_e}} \underbrace{\frac{\partial \Pi(\mathbf{a})}{\partial \mathbf{a}}\bigg|_{\mathbf{a}=\widehat{\boldsymbol{\mathcal{X}}}_s^\perp}^{2\times 3}}_{\mathbf{J}_\Pi} \underbrace{\frac{\partial s(\boldsymbol{\mathcal{X}}_s,\mathbf{a})}{\partial \mathbf{a}}\bigg|_{\mathbf{a}=S(\widehat{\mathbf{n}})}^{3\times 3}}_{\mathbf{J}_s} \underbrace{\frac{\partial S(\mathbf{a})}{\partial \mathbf{a}}\bigg|_{\mathbf{a}=\widehat{\mathbf{n}}}^{3\times 3}}_{\mathbf{J}_S} \frac{\partial \mathbf{n}}{\partial \mathbf{x}_{TL}}\bigg|_{\mathbf{x}_{TL}=\mathbf{0}}^{3\times l}$$

We can note that the differentiation of $\mathbf{n}$ appears for all proposed distances when applying the chain rule.

The jacobians specific to d_r have the following values:

$$\boxed{\mathbf{J}_{d_e} = \frac{1}{d_e(\widehat{\mathbf{m}}^\perp, \mathbf{m})}(\widehat{\mathbf{m}}^\perp - \mathbf{m})^\top}$$

$\mathbf{J}_\Pi$ is detailed in Appendix A.

$$\boxed{\mathbf{J}_S = \frac{\mathbf{I}_3}{\|\widehat{\mathbf{n}}\|} - \frac{\widehat{\mathbf{n}}\widehat{\mathbf{n}}^\top}{\|\widehat{\mathbf{n}}\|^3}}$$

$$\boxed{\mathbf{J}_s = -\frac{\widehat{\mathbf{n}}\boldsymbol{\mathcal{X}}_s^\top + (\boldsymbol{\mathcal{X}}_s^\top\widehat{\mathbf{n}})\mathbf{I}_3}{\sqrt{1-(\boldsymbol{\mathcal{X}}_s^\top\widehat{\mathbf{n}})^2}} + \frac{(\boldsymbol{\mathcal{X}}_s^\top\widehat{\mathbf{n}})\, s(\boldsymbol{\mathcal{X}}_s,\widehat{\mathbf{n}})\, \boldsymbol{\mathcal{X}}_s^\top}{1-(\boldsymbol{\mathcal{X}}_s^\top\widehat{\mathbf{n}})^2}}$$

Differentiation with respect to $\mathbf{x}_T$

We will use the following notation to simplify the expressions: $\widehat{\mathcal{L}} = [\widehat{\mathcal{L}}_{1:3}\ \widehat{\mathcal{L}}_{4:6}] = \mathcal{L}(\widehat{\mathbf{U}}, \widehat{\mathbf{W}})$.

$$\left.\frac{\partial \mathbf{n}}{\partial \mathbf{x}_T}\right|_{\mathbf{0}}^{3\times 6} = (\mathbf{P}\widehat{\mathbf{T}})^{3\times 6} \left.\frac{\partial \mathbf{T}(\mathbf{x}_T)\widehat{\mathcal{L}}}{\partial \mathbf{x}_T}\right|_{\mathbf{0}}^{6\times 6}$$

$$\left.\frac{\partial \mathbf{T}(\mathbf{x}_T)\widehat{\mathcal{L}}}{\partial \mathbf{x}_T}\right|_{\mathbf{0}}^{6\times 6} = \left.\frac{\partial \mathbf{T}\widehat{\mathcal{L}}}{\partial \mathbf{T}}\right|_{\mathbf{T}=\mathbf{I}}^{6\times 36} \left.\frac{\partial \mathbf{T}(\mathbf{x}_T)}{\partial \mathbf{x}_T}\right|_{\mathbf{0}}^{36\times 6}$$

$$\left.\frac{\partial \mathbf{T}\widehat{\mathcal{L}}}{\partial \mathbf{T}}\right|_{\mathbf{T}=\mathbf{I}}^{6\times 36} = \begin{bmatrix} \widehat{\mathcal{L}}^\top & & & & & \\ & \widehat{\mathcal{L}}^\top & & & & \\ & & \widehat{\mathcal{L}}^\top & & & \\ & & & \widehat{\mathcal{L}}^\top & & \\ & & & & \widehat{\mathcal{L}}^\top & \\ & & & & & \widehat{\mathcal{L}}^\top \end{bmatrix}$$

With $\mathbf{A}_i$ the generators of the Lie algebra:

$$\left.\frac{\partial \mathbf{T}(\mathbf{x}_T)}{\partial \mathbf{x}_T}\right|_{\mathbf{0}}^{36\times 6} = [\mathrm{flat}(\mathbf{A}_1)^\top\ \mathrm{flat}(\mathbf{A}_2)^\top\ \cdots\ \mathrm{flat}(\mathbf{A}_6)^\top]_{36\times 6}$$

with: $\mathrm{flat}(\mathbf{M}_{n\times m}) = [m_{11}\ m_{12}\ \cdots\ m_{1m}\ m_{21}\ m_{22}\ \cdots m_{nm}]$

After simplification:

$$\left.\frac{\partial \mathbf{T}(\mathbf{x}_T)\widehat{\mathcal{L}}}{\partial \mathbf{x}_T}\right|_{\mathbf{0}}^{6\times 6} = -\begin{bmatrix} \left[\widehat{\mathcal{L}}_{4:6}\right]_\times & \left[\widehat{\mathcal{L}}_{1:3}\right]_\times \\ \mathbf{0}_{3\times 3} & \left[\widehat{\mathcal{L}}_{4:6}\right]_\times \end{bmatrix}$$

We finally obtain:

$$\boxed{\left.\frac{\partial \mathbf{n}}{\partial \mathbf{x}_T}\right|_{\mathbf{0}}^{3\times 6} = -\left[\left(\widehat{\mathbf{R}}\left[\widehat{\mathcal{L}}_{4:6}\right]_\times\right) \quad \left(\widehat{\mathbf{R}}\left[\widehat{\mathcal{L}}_{1:3}\right]_\times + \left[\widehat{\mathbf{t}}\right]_\times \widehat{\mathbf{R}}\left[\widehat{\mathcal{L}}_{4:6}\right]_\times\right)\right]}$$

Differentiation with respect to $\mathbf{x}_L$

$$\left.\frac{\partial \mathbf{n}}{\partial \mathbf{x}_L}\right|_{\mathbf{0}}^{3\times 4} = (\mathbf{P}\widehat{\mathbf{T}})^{3\times 6} \left.\frac{\partial \mathcal{L}(\widehat{\mathbf{U}}\mathbf{U}(\mathbf{x}_L), \widehat{\mathbf{W}}\mathbf{W}(\mathbf{x}_L))}{\partial \mathbf{x}_L}\right|_{\mathbf{0}}^{6\times 4}$$

$\mathbf{U}(\mathbf{x}_L)$ only depends on the first three values of $\mathbf{x}_L$ noted $\mathbf{x}_L^{1:3}$ and $\mathbf{W}(\mathbf{x}_L)$ only depends on the last value of $\mathbf{x}_L$ noted $\mathbf{x}_L^4$, thus:

$$\left.\frac{\partial \mathcal{L}(\widehat{\mathbf{U}}\mathbf{U}(\mathbf{x}_L), \widehat{\mathbf{W}}\mathbf{W}(\mathbf{x}_L))}{\partial \mathbf{x}_L}\right|_{\mathbf{0}}^{6\times 4} = \left[\left.\frac{\partial \mathcal{L}(\widehat{\mathbf{U}}\mathbf{U}(\mathbf{x}_L), \widehat{\mathbf{W}}\mathbf{W}(\mathbf{x}_L))}{\partial \mathbf{x}_L^{1:3}}\right|_{\mathbf{0}}^{6\times 3} \quad \left.\frac{\partial \mathcal{L}(\widehat{\mathbf{U}}\mathbf{U}(\mathbf{x}_L), \widehat{\mathbf{W}}\mathbf{W}(\mathbf{x}_L))}{\partial \mathbf{x}_L^{4}}\right|_{\mathbf{0}}^{6\times 1}\right]$$

$$\left.\frac{\partial \mathcal{L}(\widehat{\mathbf{U}}\mathbf{U}(\mathbf{x}_L), \widehat{\mathbf{W}}\mathbf{W}(\mathbf{x}_L))}{\partial \mathbf{x}_L^{1:3}}\right|_{\mathbf{0}}^{6\times 3} = \left.\frac{\partial \mathcal{L}(\mathbf{A}, \widehat{\mathbf{W}})}{\partial \mathbf{A}}\right|_{\mathbf{A}=\widehat{\mathbf{U}}}^{6\times 9} \left.\frac{\partial \widehat{\mathbf{U}}\mathbf{B}}{\partial \mathbf{B}}\right|_{\mathbf{B}=\mathbf{I}}^{9\times 9} \left.\frac{\partial \mathbf{U}(\mathbf{x}_L)}{\partial \mathbf{x}_L^{1:3}}\right|_{\mathbf{0}}^{9\times 3}$$

$$\left.\frac{\partial \mathcal{L}(\widehat{\mathbf{U}}\mathbf{U}(\mathbf{x}_L), \widehat{\mathbf{W}}\mathbf{W}(\mathbf{x}_L))}{\partial \mathbf{x}_L^{4}}\right|_{\mathbf{0}}^{6\times 1} = \left.\frac{\partial \mathcal{L}(\widehat{\mathbf{U}}, \mathbf{C})}{\partial \mathbf{C}}\right|_{\mathbf{C}=\widehat{\mathbf{W}}}^{6\times 4} \left.\frac{\partial \widehat{\mathbf{W}}\mathbf{D}}{\partial \mathbf{D}}\right|_{\mathbf{D}=\mathbf{I}}^{4\times 4} \left.\frac{\partial \mathbf{W}(\mathbf{x}_L)}{\partial \mathbf{x}_L^{4}}\right|_{\mathbf{0}}^{4\times 1}$$

The generators for $\mathfrak{so}(3)$ are:

$$\mathbf{C}_1 = [\mathbf{b}_x]_\times, \mathbf{C}_2 = [\mathbf{b}_y]_\times, \mathbf{C}_3 = [\mathbf{b}_z]_\times$$

and the generator for $\mathfrak{so}(2)$ is:

$$\mathbf{D} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Let $\widehat{\mathbf{u}}_i$ be the $i-$th column of $\widehat{\mathbf{U}}$. With a similar derivation to that of $\mathbf{x}_T$, we obtain:

$$\left.\frac{\partial \mathbf{n}}{\partial \mathbf{x}_L}\right|_{\mathbf{0}}^{3\times 4} = \begin{bmatrix} \widehat{\mathbf{R}} & \left[\widehat{\mathbf{t}}\right]_\times \widehat{\mathbf{R}} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{3\times 1} & -\widehat{\mathbf{u}}_3 & \widehat{\mathbf{u}}_2 & \mathbf{0}_{3\times 1} \\ \frac{\widehat{\mathbf{W}}_{21}}{\widehat{\mathbf{W}}_{11}}\widehat{\mathbf{u}}_3 & \mathbf{0}_{3\times 1} & -\frac{\widehat{\mathbf{W}}_{21}}{\widehat{\mathbf{W}}_{11}}\widehat{\mathbf{u}}_1 & \frac{1}{\widehat{\mathbf{W}}_{11}^2}\widehat{\mathbf{u}}_2 \end{bmatrix}$$

7.6 Experimental results

7.6.1 Simulated data

Our experimental setup consists of a parabolic mirror ($\xi = 1$) with a generalised focal length of $\gamma = 270$ (this value was chosen from a real camera). Lines were randomly generated at a distance of the camera between 0 and 8 m. The images were spaced by a random transformation with a translation between $[0; 10]$ cm and a rotation between $[0; \pi/2]$ rad to simulate an incremental motion. We added Gaussian noise to the end-points of each line projected in the image. The given values are the mean over 40 trials. The aim of these experiments was to assess the quality of the distances on the sphere. We also wanted to answer the question: is it better to have a lot of lines with few images or a lot of images with few lines ? (i.e. the trade-off between frame rate and the processing time taken for the line extraction and tracking)

Figure 7.6 shows the effect of errors in the image on the estimation of the translation for 10 lines seen in 15 images (the rotation gave similar results). The reprojection distance d_r gave a far better accuracy than the Riemann d_R and algebraic d_A distances with similar results.

Figure 7.7 shows the effect of the number of lines over the quality of the estimation for a fixed noise of 2 pixels and 10 images (only d_r is shown, the other distances gave similar results). The number of lines has a strong influence over the accuracy of the estimates. Figure 7.8 assesses the improvement in accuracy as the amount of images increases for 3 lines and a fixed noise of 2 pixels. The number of images only improves the estimates slightly. For robotic applications, these results indicate that it might be preferable to estimate and track as many lines as possible rather than obtain many images (with for example a high frame rate). This is coherent with (7.24).

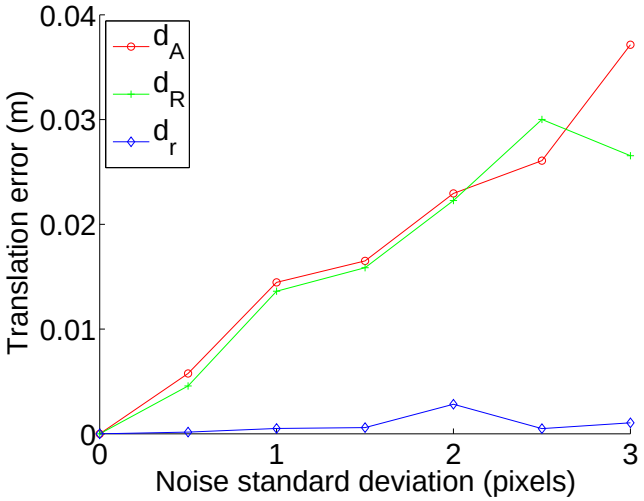


Figure 7.6: Translation error for different distances when varying the added noise on the line end-points

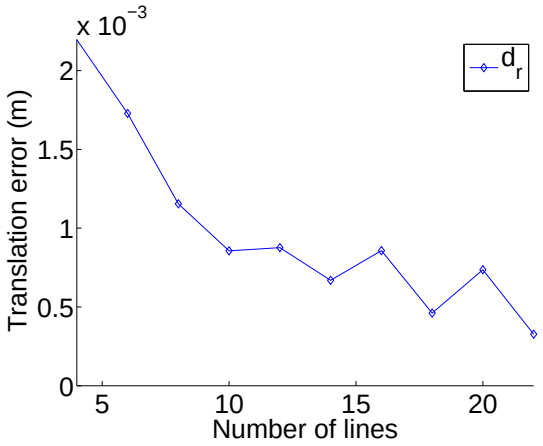


Figure 7.7: Translation error when varying the number of lines

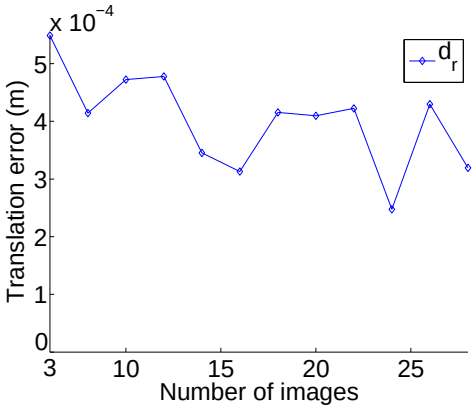


Figure 7.8: Translation error when varying the number of images

7.6.2 Real data

7.6.2.1 Technical details

The RHT was used for the line extraction. In the case of non-perspective omnidirectional sensors, the large field of view and the relative low resolution means that we obtain strong gradient responses typically around the mirror border. To improve the line image extraction, the voting scheme used a measure of confidence based on the expected gradient direction (from (7.1) and (7.14)) and the observed gradient direction in the image.

For the tracking, after a search along the normals, the line image parameters were extracted with RANSAC followed by a least-square minimisation (7.2).

To avoid line images "jumping" between two potential lines, we only considered lines with relatively few outliers ($\sim 40\%$) and with "enough" supporting points.

7.6.2.2 Experiment

The validation was done on a sequence of 35 images where point or template-based approaches gave unsatisfying results. The sensor used is a parabolic mirror with a telecentric lens and a perspective camera of resolution 1280×1024[1]. The motion was constrained in a plane by only estimating 1 rotation and 2 translation parameters in the Lie algebra. The initial values given were the identity for the transformations and the cross product between $\mathbf{n}$ and a point $\boldsymbol{\mathcal{X}}_s$ for the second component of the Plücker coordinates. The initial pixel reprojection error was of 36.3 pixels. After minimisation of the cost function with the Levenberg-Marquardt algorithm, it was reduced to 0.86 pixels.

Figure 7.9 shows the first and last image of the corridor sequence (the images were flipped to ease the comparison with the 3D model). No new lines were added during the tracking. Not all lines could be tracked through the entire sequence. Figure 7.10 shows two views of the reconstructed scene with the robot motion. Without being entirely satisfying, the results are sufficiently accurate (~ 5 cm over 1 m) to be beneficially in combination with other visual features or extra sensors.

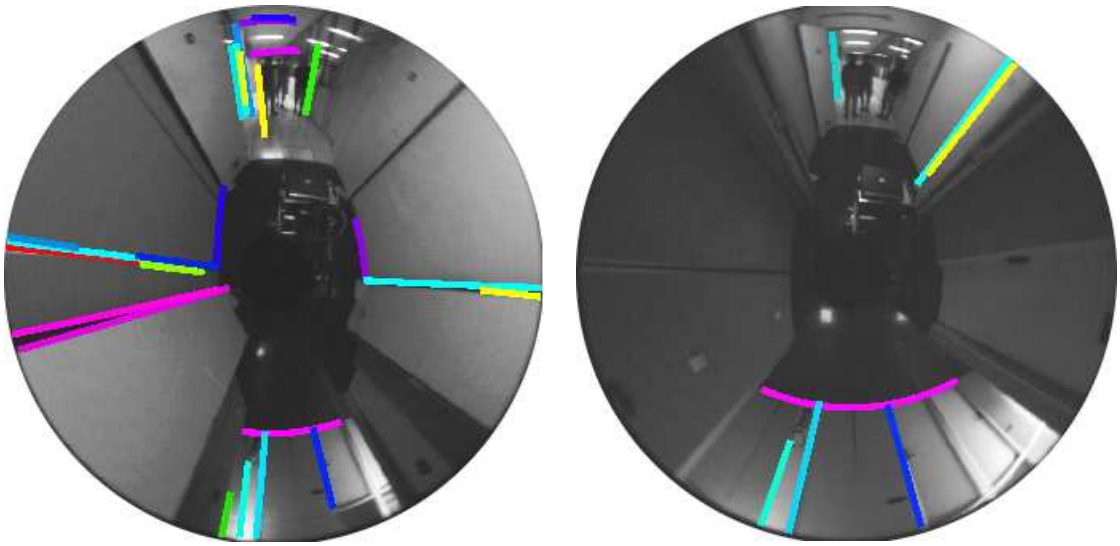


Figure 7.9: First and last image of the corridor sequence

[1] the toolbox used for the calibration is available as open-source software on the author's website

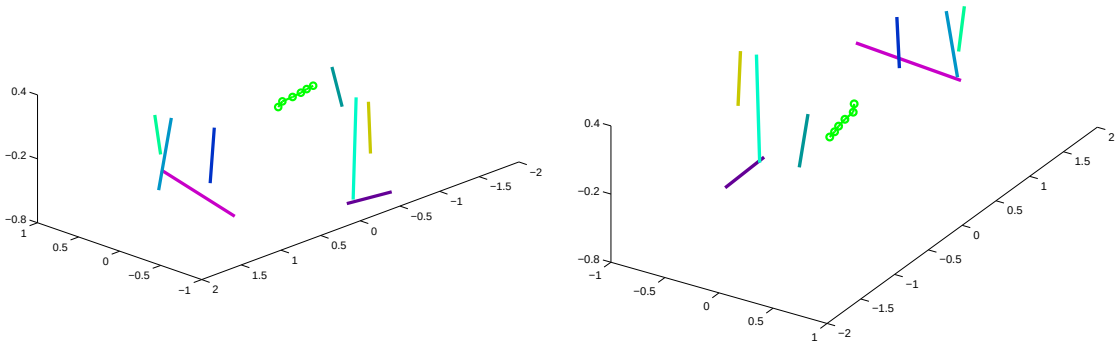


Figure 7.10: Two views of the 3D reconstruction of the scene with the robot motion depicted by the green line with circles

7.7 Conclusion

In this chapter we have presented algorithms for automatic structure from motion from lines for central catadioptric sensors. We focused on important aspects for robotic applications such as robustness and minimal parameterisation. From our experimental and simulation results, we do not believe lines constrain the motion sufficiently to be used alone. They can however provide additional robustness to mapping and navigational tasks in low-textured man-made environments.

Part III

Simultaneous localisation and mapping from a laser range finder and an omnidirectional camera

Chapter 8

A short overview of SLAM

Contents

8.1 Introduction

8.1.1 Simultaneous localisation and mapping

Localisation is the problem of estimating the motion of a robot given a *known map*. Mapping is the creation of a map of the environment from measurements *knowing the true path* of the robot. When neither the robot path nor the map are known, localisation and mapping must be considered concurrently. This problem is known as *Simultaneous* Localisation And Mapping (SLAM). The aim is to recover the path of the robot and a map based on measurements of its ego-motion and of features in the environment, both corrupted by noise. This problem is central to building autonomous robots and has been at the focus of a lot of research since the 1980's. A survey of robotic mapping can be found in [Thrun, 2002] or in the more recent book dedicated to the subject [Thrun et al., 2005]. Two recent tutorials by Hugh Durrant-Whyte and Tim Bailey [Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006] describe some of the standard methods for solving the SLAM problem but also some more recent algorithms. They contain up-to-date references to online software and datasets.

8.1.2 Applications of SLAM

SLAM has found many applications in areas where accurate global positioning (obtained through GPS or using beacons) is not available. It has proved essential for mapping dangerous areas such as abandoned mines (Figure 8.1 (a)) or regions where it is difficult or even impossible to send humans such as underwater environments, ([Williams and Mahon, 2004] and Figure 8.1 (b)) or distant planets. Autonomous systems can also ensure accurate and safe navigation for applications such as cargo handling (Figure 8.1 (c)).

The motivations for using autonomous systems can be the security of people involved, the repeatability of the task in particular for surveying large areas and the precision of the maps obtained.

The flexibility of the approach makes it possible to deploy robots without requiring human intervention in environments that change with time.

8.1.3 Historical overview

Localisation and mapping has been an active field of research since the 1980's.

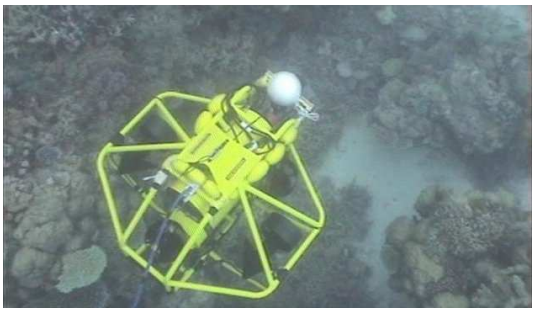
With mapping naturally arises the question: what should be represented, how and for what applications ?

Topological maps, often represented as graphs, describe the connectivity of places whereas metric maps capture the geometric structure of the environment. The difference is not always evident as most topological maps include a metric notion and most navigation algorithms based on geometric representations use some sort of topological abstraction.

Occupancy grids [Elfes, 1989] or polyhedral representations [Chatila and Laumond, 1985] belong to some of the early map representations. They are metric in the sense that we can measure the distance covered by the robot based on the map but topological as the possible ways to access places are directly included in the representation. This is of course essential for motion planing and explains the success of occupancy grids in real applications [Thrun et al., 2000]. The topological-metric relationship is not always present in particular when geometric features are used.

The problem of how to build the maps probabilistically appeared in the same time as the representations were being studied. Advances appeared simultaneously in visual navigation [Ayache and Faugeras, 1988] and in sonar-based navigation of mobile robots [Chatila and Laumond, 1985; Moutarlier and Chatila, 1989]. Since the articles by Smith, Self and Cheeseman [Smith and Cheeseman, 1986;

(a)



(b)

(c)

Figure 8.1: Different applications of autonomous systems: (a) Mine mapping at CMU (b) Exploration of the coral reef at the ACFR (c) Transporting cargo containers at the ACFR

Smith et al., 1990] in the 1990's, stochastic mapping has become the dominant approach to SLAM. The term SLAM itself appeared a few years later [Leonard and Durrant-Whyte, 1991]. Important theoretical results concerning the convergence of the SLAM problem if considered as a whole where studied by Csorba [Csorba, 1997] and Dissanayake *et al.* [Dissanayake et al., 2001]. Many probabilistic approaches to SLAM exist in the literature. The most widely used is the Kalman filter with maps generally consisting of a set of landmarks representing the environment. Bundle adjustment [Triggs et al., 1999] from the computer vision literature concerns all the maximum likelihood (ML) or maximum a posteriori (MAP) techniques including Kalman filtering even though the tendency in computer vision is to use batch algorithms that require all the data. Expectation maximisation [Dempster et al., 1977] can also be applied to the SLAM problem, it requires all the data but can solve the correspondence/association problem (i.e. if measurements correspond to same physical entity). Recent research has focused on solving the correspondence problem in real-time through mixtures of Gaussians or particle filters [Montemerlo, 2003]. It is however unclear how to determine the number of Gaussians or particles. These methods are shown to be inconsistent in the general case [Bailey et al., 2006b].

Robotic exploration is often considered independently from the map building task. This is somewhat surprising as the aim of an autonomous system is rarely the map itself but often the navigation in the environment. Combining navigation and mapping can lead to elegant solutions [Victorino, 2002]. Sensor-based approaches such as visual servoing show interesting and promising solutions to navigation, localisation and mapping [Silveira et al., 2006; Remazeilles et al., 2006].

8.2 Solutions to the SLAM problem

In this section, we will describe some of solutions and issues regarding the *estimation* of the motion and uncertainty in probabilistic mapping.

8.2.1 Notations and formulation of probabilistic SLAM

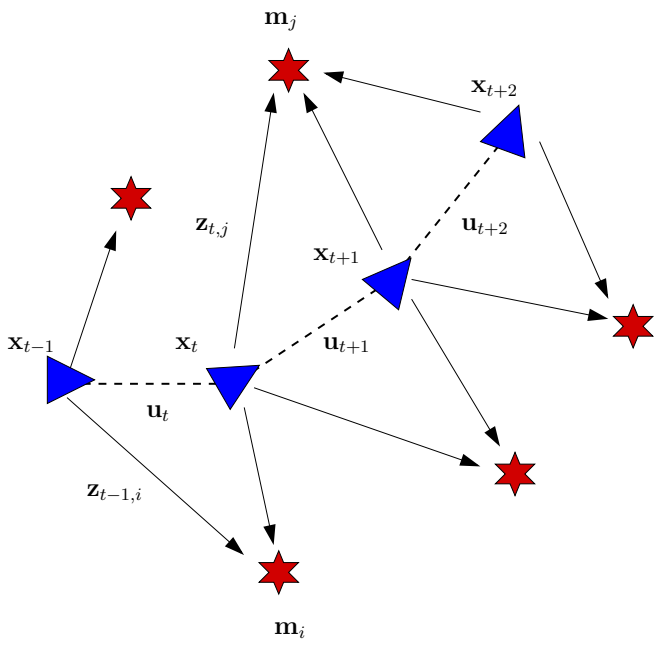


Figure 8.2: Notations for the SLAM problem

The following notations will be used [Durrant-Whyte, 2002; Durrant-Whyte and Bailey, 2006] and figure 8.2:

- a discrete time index $t = 1, 2, ...,$
- $\mathbf{x}_t$ the true location of the robot at a discrete time t,
- $\mathbf{u}_t$ a control vector applied at time $t-1$ to drive the robot from $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ at time t,
- $\mathbf{m}_i$ the position of the i^{th} feature or landmark,
- $\mathbf{z}_{t,i}$ an observation or measure of the i^{th} feature made in $\mathbf{x}_t$ at time t,
- $\mathbf{z}_t$ a generic observation of all the landmarks at time t.

We can also define the following sets:

- a history of past states: $\mathbf{X}_{0:t} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t\} = \{\mathbf{X}_{0:t-1}, \mathbf{x}_t\}$
- a history of control inputs: $\mathbf{U}_{0:t} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_t\} = \{\mathbf{U}_{0:t-1}, \mathbf{u}_t\}$

- the set of all landmarks: $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_M\}$
- the history of the landmark observations: $\mathbf{Z}_{0:t} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t\} = \{\mathbf{Z}_{0:t-1}, \mathbf{z}_t\}$

In the simultaneous localisation problem, we assume that:

- no prior information is available on the features $\mathbf{m}$ that form the map,
- the initial position or origin $\mathbf{x}_0$ is known,
- the control sequence $\mathbf{U}_{0:t}$ is also known.

From the observations or measures acquired by the robot, the problem is then to build incrementally and simultaneously the map $\mathbf{m}$ and the set of positions $\mathbf{X}_{0:t}$ of the robot. These two problems are coupled, as from a single measure $\mathbf{z}$, we wish to find two quantities, the position of the robot $\mathbf{x}_t$ and the position of the landmark $\mathbf{m}$. A solution can only be found by considering these problems concurrently. Written in probabilistic form, the solution to the SLAM problem requires the estimation of the following joint posterior density of the vehicle state and landmarks knowing all the observations and controls given to the robot:

$$P(\mathbf{x}_t, \mathbf{m} | \mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{x}_0) \tag{8.1}$$

Recursive formulation of the problem The *motion model* or *vehicle model* expresses the current pose knowing the previous poses and controls. We make the assumption that the positions form a Markov chain: given the present state, the future and past states are independent. Assuming that the pose does not depend on the map, this can be expressed formally as:

$$P(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) = P(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{X}_{0:t-2}, \mathbf{U}_{0:t-1}, \mathbf{m})$$

Under this assumption, the law of total probability[1] and Bayes' rule[2] leads to the *time update equation*:

$$\begin{aligned} P(\mathbf{x}_t, \mathbf{m} | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0) &= \int P(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{m} | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0) d\mathbf{x}_{t-1} \\ &= \int P(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{m}, \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0) P(\mathbf{x}_{t-1}, \mathbf{m} | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t-1}, \mathbf{x}_0) d\mathbf{x}_{t-1} \\ &= \int P(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) P(\mathbf{x}_{t-1}, \mathbf{m} | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t-1}, \mathbf{x}_0) d\mathbf{x}_{t-1} \end{aligned} \tag{8.2}$$

We also assume that the measurements are conditionally independent and only depend on the current position, which is a reasonable assumption in practice:

$$P(\mathbf{Z}_{0:t} | \mathbf{X}_{0:t}, \mathbf{m}) = \Pi_{i=1}^{t} P(\mathbf{z}_i | \mathbf{X}_{0:t}, \mathbf{m})$$

Applying Bayes' rule to expand the joint distribution in terms of the state and then in terms of the landmark observations gives the following two equalities:

$$\begin{aligned} P(\mathbf{x}_t, \mathbf{m}, \mathbf{z}_t | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0) &= P(\mathbf{x}_t, \mathbf{m} | \mathbf{Z}_{0:\mathbf{t}}, \mathbf{U}_{0:t}, \mathbf{x}_0) P(\mathbf{z}_t | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0) \\ P(\mathbf{x}_t, \mathbf{m}, \mathbf{z}_t | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0) &= P(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}) P(\mathbf{x}_t, \mathbf{m} | \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0) \end{aligned}$$

[1] Reminder: $P(x) = \int P(x, y) dy$
[2] Reminder: $P(x, y) = P(x|y) P(y)$

$P(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m})$ is often called the *observation model* or *measurement probability* and requires accurate modeling of the measurement errors of the sensor.

By combining these equations, we obtain what is referred to as the *measurement update equation*:

$$P(\mathbf{x}_t, \mathbf{m}|\mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{x}_0) = \frac{P(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m})P(\mathbf{x}_t, \mathbf{m}|\mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0)}{P(\mathbf{z}_t|\mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t})} \tag{8.3}$$

From equations (8.2) and (8.3), we obtain the recursive formulation to the SLAM problem:

$$P(\mathbf{x}_t, \mathbf{m}|\mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{x}_0) = \eta P(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m}) \int P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)P(\mathbf{x}_{t-1}, \mathbf{m}|\mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t-1}, \mathbf{x}_0)d\mathbf{x}_{t-1}$$

with η a normalising constant.

We will now describe different solutions to the SLAM problem:

- recursive solutions: Kalman filter, particle filter and SLAM occupancy grids,
- global minimisation methods that require the entire dataset: bundle adjustment and expected maximisation.

8.2.2 Kalman filters

The Kalman filter and more generally the Extended Kalman Filter approximates the joint posterior density (equation (8.1)) as a high dimensional Gaussian over the robot pose and the map features. In Appendix D, we describe the equations of the Kalman filter. A basic implementation of this filter was used for the experiments in Chapter 9.

By keeping explicitly the correlation between the landmarks and the robot pose, the Kalman filter can ensure convergence towards a lower bound determined by the initial uncertainty in the robot pose and measurements.

The Kalman filter has become a standard approach to solving the simultaneous localisation problem. The popularity of the approach, besides the ease of implementation, can be explained by its optimality properties. The Kalman filter provides a linear minimum variance estimation of discrete-time systems. The following assumptions are needed to ensure optimality:

- the process noise $\mathbf{w}_t$ and measurement noise $\mathbf{v}_t$ have zero mean, uncorrelated white-noise processes with known covariance matrices.
- the initial position considered as a random vector $\mathbf{x}_0$ is uncorrelated to $\mathbf{w}_t$ and $\mathbf{v}_t$ and has a known mean and covariance.

For complex robotic systems, the Gaussian nature of the measurement noise is often justified by invoking the Central Limit theorem that states that if a sum of the variables has a finite variance, then it will be approximately normally distributed.

There are however some important issues using the Kalman filter for SLAM:

- linear approximation. The linearisation used in the EKF leads to inconsistencies in the solution. Results on the convergence and consistency[3] of the filter have only been shown in the linear case.

[3] in other words the system will become over-confident regarding its pose and those of the landmarks

- complexity. For each new landmark, the correlation with all the values of the state vector must be saved. The observation update equation also requires an update of the landmark poses and joint covariance. In other words, a naive implementation of the filter is quadratic in time and memory usage.

- data association. the standard approach to Kalman filtering is to assign an observation to the landmark the most likely to have generated it. This can be done by nearest neighbour gating or better by joint compatibility test [Neira and Tardos, 2001]. However the Kalman filter makes a single data association hypothesis at every time step and is sensitive to incorrect associations. Some authors refer to this as the lack of robustness of the EKF solution [Newman et al., 2006]. This difficulty is also enhanced by the inconsistencies introduced by the linearisation.

These issues have all been studied extensively over the past decade.

The Unscented Kalman Filter (UKF) [Julier and Uhlmann, 1997] addresses the problem of the errors introduced in the true posterior mean and covariance by the linearisation of the non-linear equations. By using a deterministic sampling, a set of sample points are chosen around the mean. These points are then propagated through the non-linear functions and the covariance of the estimate is recovered. This captures the posterior mean and covariance accurately to the 3rd order for any nonlinearity compared to a first-order accuracy for the EKF. This method gives better results than the standard EKF but in no way removes the underlying problem due to the linearisation.

The problem of complexity has been thoroughly studied to enable real-time mapping of large environments. The methods generally exploit the sparsity in the dependencies between the local robot position and distant landmarks to build local maps. [Guivant and Nebot, 2000] achieve a linear complexity without any approximations. In [Leonard and Feder, 1999], the authors achieve constant time updates but the solution does not guarantee consistency. The ATLAS framework [Bosse et al., 2003] achieves a constant time performance but does not compute state estimates with respect to a single global reference frame. More recently Leonard and Newman also proposed a consistent and constant-time SLAM approach which also ensures convergence [Leonard and Newman, 2003]

Other approaches to not require the sub-map framework and in this sense are more systematic. The sparse information filter [Thrun et al., 2004] exploits the sparsity of the inverse of the covariance matrix to derive constant time updates for the time and measurement equations. Modifications have to be made however to the initial framework to ensure consistency [Walter et al., 2005]. Alternatives exists such as the *covariance intersection* [Julier and Uhlmann, 2001] which also provably avoids over-confidence or *thin junction trees* [Paskin, 2002].

What should be noted however about the notion of convergence and consistency is that they are only proved in these studies for the linear case. To summarise, several methods now exists with constant-time updates which are provably consistent and convergent in the linear case. In the non-linear case, difficulties remain. Authors have even come to question if the framework proposed 20 years earlier by Smith, Self and Cheeseman was well-founded [Julier and Uhlmann, 2001]. These concerns have been confirmed more recently [Bailey et al., 2006a].

The ambiguity in data association is another difficult problem to solve in EKF-SLAM frameworks. Several solutions have been proposed such as using the Hough transform [Tardos et al., 2002] which gives impressive results on difficult sonar data. In [Dissanayake et al., 2001] a maximum likelihood approach is proposed, later used in [Thrun et al., 2004] with notations that make the data association explicit in the formulation of the SLAM problem. These approaches however all choose a single data association to update the Kalman filter. In [Nebot et al., 2003], the authors combine the EKF with a particle filter to remove the ambiguity of local associations. A situation where data association

is critical is when the uncertainty related to the robot pose and landmarks becomes too important for simple maximum likelihood matching, or when the filter becomes altogether inconsistent. This problem is generally referred to as "loop closing" in the literature. It is one of the motivations for using omnidirectional vision as we will see in Chapter 9.

Particle filters and in particular FastSLAM [Montemerlo, 2003] have paved the way for a different framework: the distributions are no longer considered Gaussian and the data association can be modeled explicitly in the estimation process. The next section gives a insight into this work.

8.2.3 Particle filters

Equation (8.1) will be rewritten to take into account explicitly the landmark correspondences as in [Montemerlo, 2003; Thrun et al., 2004].

To simplify the notations, we will assume that a single measurement is obtained at a given time step. This does not affect the generality of the approach that can be applied sequentially. The landmark corresponding to the observation $\mathbf{z}_t$ will be identified by a value $\mathbf{c}_t$. $\mathbf{C}_{0:t}$ will correspond to the set of associations:

$$\mathbf{C}_{0:t} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_t\}$$

Rewriting equation (8.1) to put an emphasis on the known correspondences becomes:

$$P(\mathbf{x}_t, \mathbf{m} | \mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{C}_{0:t}, \mathbf{x}_0) \tag{8.4}$$

The idea of particle filtering with SLAM comes from the observation that the correlation between observations only occurs through the robot pose. If the full trajectory (all the poses $\mathbf{X}_{0:t}$ are known), we have a simple mapping problem with conditionally independent landmarks.

Equation (8.4) can be written:

$$P(\mathbf{X}_{0:t}, \mathbf{m} | \mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{C}_{0:t}, \mathbf{x}_0) = P(\mathbf{m} | \mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{C}_{0:t}, \mathbf{x}_0) P(\mathbf{X}_{0:t} | \mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{C}_{0:t}, \mathbf{x}_0)$$

The idea is to represent this posterior by a set of particles representing the sample path of the robot. For the problem to be tractable in the high dimensional search space, a Rao-Blackwellisation filter [Doucet et al., 2000] is used. The sampling can be done not only over the poses but also over the possible data associations.

FastSLAM achieves an $O(M \log(N))$ complexity, with N the number of features and M the number of particles, and has shown its ability to map large areas [Montemerlo, 2003]. However the dependency between the pose and measurement history means that the resampling prevents a consistent long-term estimate of the joint posterior [Bailey et al., 2006b].

8.2.4 Bundle adjustment and expected maximisation

Batch update methods - generally maximum likelihood approaches - are very popular in computer vision where the problem of recursive real-time estimation is less central [Triggs et al., 1999]. The vast majority of these approaches assume that the data association has been made. They rely on minimising all the poses and the structure from the data association. In vision, for example, the data could be points projected in several views and matched by correlation. Bundle adjustment would then consist in minimising the reprojection error to recover the camera and point positions. For the calculation to stay tractable, the sparsity of the SLAM problem is exploited.

These type of methods can be used in robotics by "local" optimisation or "incremental maximum likelihood" but as such the values cannot be revised and the algorithm cannot map cyclic environments.

A way of overcoming these limitations is to keep a notion of uncertainty to enable global optimisation over the poses when loop closing situations are found [Gutmann and Konolige, 1999]. The updates are then constant time during motion estimation and mapping but during loop closure the complexity depends on the size of the loop. The limitations of these methods are: 1) data association. There has to be a reliable mechanism for the loop closing to take place. 2) computational time. The loop closing can be expensive if there are a lot of poses and means the method is not real-time. 3) local optima. The optimisation or "loop closing" can fail to converge towards the global optima, this can lead to an inconsistent map. This approach was also used for mapping with teams of robots [Konolige et al., 1999; Thrun, 2001].

An alternative optimisation approach to solve the correspondence problem is expected maximisation (EM) [Dempster et al., 1977; Thrun et al., 1998; Burgard et al., 1999]. The underlying idea is similar to particle filtering. As building the map with a known robot path is relatively simple, the algorithm separates the estimation of the global posterior over the poses and map by two optimisation steps. The first, the *expectation step*, calculates the posterior over the robot poses for a given map. The second, the *maximisation step* calculates the most likely pose given the robot poses. At each cycle the maps become increasingly accurate. The algorithm has proved to give good results with particularly difficult correspondence problems. However, as with any non-linear optimisation method, this approach also suffers from the risk of local maxima and is computationally expensive. Recent work has shown the possibility of applying this approach at "exploration compatible frame rate" under the assumption of good initial values [Thrun et al., 2003]. Strictly speaking it is not real-time as for loop closing the complexity will depend on the size of the loop as with bundle adjustment.

8.3 Map representations

In the previous sections, we presented the theoretical issues and some solutions to probabilistic SLAM. We have not discussed however what to represent and how. This is of course an essential part of map building and is strongly related to the application and the sensors. We will now discuss some of the choices made in the research community.

Maps can be described by one or several of the following categories:

- topological. Topological maps are generally represented by a graph. An arc indicates the accessibility of different sites in the environment. The vertices often contain identifiers of the visited places: image descriptors, sets of points (eg. SIFT points), *etc.*

- geometrical. Features such as points, lines, curves are used to represent the map. A measure of probability is generally associated to the position of the features. A geometrical approach enables a direct measure of distance. However these methods require to choose an appropriate feature representation.

- grid space. A grid representation generally indicates the free space in the environment. It is well-adapted to obstacle avoidance. However a higher level of abstraction is needed to define the notion of distance between places or possible objects in the environment.

Robot exploration in probabilistic maps are often based on maximising the information gain. However probabilistic maps are not the only solution to exploration, topological representations that are not subject to problems such as inconsistency are interesting alternatives. Ensuring that the entire environment is mapped is difficult to guarantee with probabilitic methods. Voronoï diagrams [Choset

and Nagatani, 2001; Kuipers and Byun, 1991] or Morse decompositions [Acar et al., 2002] can provide topological completeness.

We will now describe some of the geometric representations used for SLAM:

- points. Points are particularly popular with visual sensors [Davison, 2003].
- lines. They are very common for indoor map building from lasers [Newman et al., 2002] as they provide relatively dense representations of these environments. They have also recently be used successfully with visual sensors [Smith et al., 2006; Eade and Drummond, 2006]. They have the advantage of being common in indoor environments even when points are difficult to extract,
- planes. Planes are interesting 3D representations as they can be used to represent complex structures with few parameters. They have been successfully applied to laser-based reconstruction in [Thrun et al., 2003], where the planes are found by Expected Maximisation. In [Molton et al., 2004], the authors compute the normals around points considered as planes but do not include the normals explicitly in the EKF state vector. As we have shown in Chapter 6, planes can be tracked efficiently and if tracked jointly provide precise structure and motion estimates.

Recently scan-slam [Nieto et al., 2006] has provided a framework for representing the uncertainty of features defined by "parts" of a laser range scan. The approach avoids the difficult (and slightly arbitrary) choice of a geometric representation and enables the use of features of any shape in an EKF filter.

2D [Elfes, 1989; Moravec, 1988] or 3D [Moravec and Martin, 1996] occupancy grids are a particularly popular probabilistic representation. Each element of the grid represents the probability of being occupied. The posterior is calculated using Bayes' rule and a sensor model. These grids have the advantage of representing the environment in a way compatible with navigation. They are also robust and able to use raw sensor information. The drawback of the grid space is the need to define a fixed granularity and it cannot - as such - incorporate motion uncertainty.

However this last point can be addressed by particle filtering as each particle represents a given pose and thus provides a way of applying directly Bayes' rule [Hähnel et al., 2003a].

In summary, maps can be chosen to represent paths between identifiable sites (topological representations), geometric features (points, lines, planes) or the free space. Geometric features are often chosen according to the sensor and the environment that should be explored. Methods that can work on raw measurements such as scan-slam or occupancy grids are popular by their ability to use nearly all the available sensor data. This might not always be a good choice especially when the quantity of information as with visual sensors. Points, lines or planes might then be better adapted to real-time constraints.

8.4 Sensors and SLAM

The evolution of SLAM is strongly linked to the sensors used. Choosing a solution (EFK, particle filters,...) should also take into account the type of incoming information. This section will illustrate some of the sensors used for SLAM and discuss the methods chosen to solve the problem.

8.4.1 Range bearing: sonars and laser range finders

Sonars were within the first sensors used in robotics for mapping. However they give relatively noisy measurements of the environment (such as phantom targets caused by crosstalk, noise from external

sources, *etc.*). Even after 20 years of research, these sensors are still considered difficult to use and advanced mapping techniques are needed to solve the correspondence problem such as Hough transforms [Tardos et al., 2002] or expected maximisation [Burgard et al., 1999] in the off-line mapping case.

Lasers have a better signal to noise ratio and have often replaced sonars. Diffusion or specularities can also introduce noise in the laser data but a pre-processing step is generally sufficient. The quality of the sensor data simplifies the correspondence problem to the point where indoor environments without too "drastic" loop closure measures [Gutmann and Konolige, 1999] are possible.

When building bigger maps, data association becomes more challenging. Furthermore, 2D sonars and lasers limit to a great extent the application of SLAM to 3-DOF estimation and indoor environments.

3D lasers are now appearing regularly in the robotics literature [Newman et al., 2006] but inherit the difficult data association problem from the nature of the sensor. The acquisition time required by these sensors is also currently incompatible with real-time exploration.

Vision is also becoming an increasingly important topic in robotic SLAM but current methods are not able to map large environments [Valls Miro et al., 2006] in real-time. The motivation for using visual sensors is the high perceptual information that greatly reduces the data association problem. With vision however, we no longer have the direct range information as with sonars or lasers.

We will now discuss more specifically vision-based SLAM.

8.4.2 Vision-based SLAM

Solutions to iterative structure and motion have (re-)appeared recently in the computer vision.

In [Chiuso et al., 2002] the authors use a Kalman filter to estimate the full 6-DOF motion of the camera using points identified in the images. This approach is not strictly speaking SLAM as they remove points regularly from the filter to guarantee constant time.

The first attempt at SLAM with monocular vision seems to have been [Broida et al., 1990]. However real-time SLAM has only become possible recently with faster computers and ways of selecting sparse but distinct features. Davison [Davison, 2003] proposed an approach that registers features during the whole SLAM process, avoiding the risk of drifting as when transient tracking points are used. He also showed that a large field of view [Davison et al., 2004] (in this article a fish-eye lens) makes it easier to find and to follow salient landmarks. The approach has only been shown to work in small office-size environments with relatively few landmarks and has not proved it could solve large loop closing problems. A specific difficulty with monocular EKF-SLAM is point initialisation as the scene is partially observable and the point depths have to be found. To overcome this problem *delayed* point insertion was first proposed such as local bundle adjustment [Deans and Hebert, 2000] or particle filtering [Davison, 2003]. More recently *undelayed* approaches have made it possible to reduce the computational cost and lead to a more systematic approach [Kwok and Dissanayake, 2004; Solà et al., 2005]. Bearing-only SLAM has been recently applied successfully in a full framework combining a loop closing approach using a panoramic camera [Lemaire and Lacroix, 2006].

Recently, visual odometry [Nistér et al., 2006; Mouragnon et al., 2006] applied to monocular and stereo vision has shown the possibility of using Maximum Likelihood approaches for 6-DOF motion estimation. This work is based on a classic framework from projective geometry (Harris points+RANSAC+optimisation [Hartley and Zisserman, 2000]) but has shown real-time capabilities and good precision. These approaches however do not have the possibility of re-capturing features or applying loop closing and are subject to drift. However this work is in fact very close to the initial work by Gutmann and Konolige [Gutmann and Konolige, 1999] and by including uncertainty estimation and loop closing it could provide precise results [Konolige, 2005]: the state estimate is not systemati-

cally linearised, the problem of 3D point initialisation is automatically solved and local robust bundle adjustment can prevent incorrect point associations.

Generally speaking, the issues that have to be solved for vision based SLAM are:

1. initialisation of the features. In the monocular case, to apply the EKF the 3D point estimates must follow a Gaussian profile. In the stereo or multiple view case, efficient ways of associating the data between views has to be found.

2. problem of observability. For directional monocular cameras, pure rotations (or motion close to pure rotation) renders the motion unobservable. This problem has been rarely addressed in the literature but is essential to produce a working system.

3. 6-DOF. Estimating the full motion of the camera is more challenging as the approximations introduced by the linearisation will have a stronger impact and techniques such as particle filtering will require even more particles to capture the non linearities.

To conclude we could say that vision in SLAM has the advantage of providing rich perceptual information compared to lasers and consequently low data association ambiguity. However visual sensors introduce other challenges such as point initialisation, observability and the non-linearities of 6-DOF estimation.

We will now describe sensor fusion and see how combining sensors can solve some these issues.

8.4.3 Combination of sensors

Within popular combination of sensors, GPS combined with odometry or with laser ranging or millimeter wave radar are within the most popular. The advantages of GPS are obvious: there is no drift and high precision can be obtained. This should not however hide the fact that GPS is not considered as a reliable sensor. Perturbations, the dependence on the satellite positioning make this sensor less attractive. Furthermore in many situations where autonomous robots are used such as in mines, underwater, on distance planets or even within towns, GPS is either unavailable or simply too imprecise. This motivates research with sensors that do not provide global positioning and are subject to drift.

Fusing sensor information can either be done at low level by combining the data before applying filtering. The information can also be considered independently at a higher conceptual level as with a Federated filter [Carlson, 1990]. This type of information management makes the system more flexible with "plug-and-play" possibilities. However the direct combination can give more reliable information and the possibility of working on the problem of observability. Combining the data also reduces the size of the state space in a probabilistic framework.

In the last part of this thesis we will show how combining laser and vision can simplify some the issues in SLAM. In Chapter 9 we analyse the problem of 3-DOF SLAM and the possibility of detecting and closing loops. In Chapter 10 we show how 2D range bearing information from the laser with omnidirectional images can provide full robust 6-DOF motion estimation.

8.5 Open problems in SLAM

Research in SLAM has generally assumed that the world was static. Initial work on dynamic environments mainly focused on the simpler problem of localisation with moving objects [Fox et al., 1999]. In SLAM, moving objects were initially considered as noise or outliers that should be rejected by the

mapping process. However identifying moving objects and anticipating their movement can improve the mapping process itself: predicting the motion of objects simplifies their identification in the data. It is also an essential component of human-machine interaction. From a navigation point of view, it can add the possibility of *active* obstacle avoidance.

Recent research has improved the understanding of mapping in dynamic environments [Biswas et al., 2002; Hähnel et al., 2003b; Wang et al., 2003; Wolf and Sukhatme, 2004]. It is interesting to note that dynamic mapping raises several specific problems: how to take into account objects that were previously considered as static objects and have moved (updating maps), how to represent dynamic objects in the maps, how to ensure that the problem stays tractable, how to define the notion of consistency for dynamic environments and objects.

The underlying theoretical framework is currently not fully understood, problems such as consistency, the relationship between real-time and multiple object tracking are still important topics of research. Even in static environments, an algorithm which is provably consistent for non-linear models does not currently exist. This is however of essential importance as it is the founding of stochastic mapping.

Computer vision is likely to become an essential component for mapping dynamic environments. Problems such as motion segmentation, tracking and behaviour understanding have a long history in the vision literature [Hu et al., 2004; Lepetit and Fua, 2005].

Other new topics of research include multiple robot mapping, that was studied for example in [Thrun, 2001]. The problems encountered are similar to the data association problem. Localisation, through omnidirectional vision for example, can help reduce the uncertainty when joining maps. Finding map representations compatible with human-robot or robot-robot interaction (eg. for solving complex tasks) is also a potentially rich topic of research.

Navigation and path planning are also essential components of fully autonomous systems. Maps should not only represent an environment precisely but also be built according to the action that has to be undertaken. In [Victorino, 2002] for example, the author chooses to combine topological completeness through Voronoï diagrams directly with the command of the robot. The metric maps are then only built locally. There are however difficult obstacles to overcome to ensure the topological representation is robust to sensor noise and complex environments.

SLAM with vision is currently a very active field of research. Interesting issues have appeared that have been less central to SLAM with lasers such as initialisation, observability and efficient data extraction. Compared to laser scans, visual information is very rich and using all the data is computationally expensive. Techniques such as active vision [Davison, 2005] that base feature extraction and tracking on the gain in information are interesting directions of research.

This thesis contributes to solving the SLAM problem by combining the complementary information obtained from 2D lasers and omnidirectional vision. These sensors improve the quality of the maps by reducing the ambiguity of landmark associations and reducing the drift by detecting and applying loop closing.

Chapter 9

Combining omnidirectional vision and laser for 3-DOF SLAM

Contents

9.1 Introduction

Simultaneous localisation and mapping is the process of finding the robot path and concurrently building a map of the environment. In this chapter, a laser range finder will be combined with an omnidirectional camera to improve the computational cost and robustness of laser-only probabilistic map building solutions.

The algorithm relies on the standard Extended Kalman Filter (EFK). Section 9.2 describes the state vector and feature representation.

The environment will be represented by a set of *salient* points: features the robot can reliably identify during exploration. How to build and re-acquire these points will be the object of Section 9.3. The choice of points was made partly to avoid the common assumption of a piecewise linear environment that is not always valid in cluttered office spaces.

As the robot explores the environment, its position will become increasingly uncertain with respect to the points it first observed. This uncertainty combined with possible bad data association or inconsistency makes it particularly difficult to re-identify previously observed landmarks after a *loop* in the environment. This problem is know as *loop closing*. It is desirable to re-identify points with strong uncertainty as this will reduce the overall uncertainty. In the literature, most loop closing approaches study how to find previously visited places. However it is also important to explicitly control the loop closing mechanism that could take place incorrectly. In Section 9.4, we combine a topological representation with a metric representation to control the association between features when the uncertainty becomes important. We then describe image descriptors than can be used to identify areas and how to match the landmarks and close the loop.

In Section 9.5, we extend Iterative Closest Point (ICP) scan matching to take into account the intensity information from the image. In corridor-like sequences this reduces the problem of observability.

Finally, we validate the salient features, loop closing and scan matching approaches in an EKF-SLAM experiment on real data.

The described approach will assume the relative position between the omnidirectional camera and the laser range finder has been obtained using for example the results of Chapter 4. The position of the vision sensor itself is not constrained. In particular we do not assume the sensor to be in vertical position even though this configuration is ideal for combining the visual and laser information in particular for loop-closing.

9.2 Map building and localisation

In Chapter 8, we gave an overview of different approaches for building a probabilistic map. The Extended Kalman Filter (EFK) is an extension to the Kalman filter to cope with non-linear state transition and measurement functions. In this chapter, we will use a simple implementation of the filter. More computationally efficient formulations are also possible but do not change the underlying difficulties of obtaining correct data associations and closing loops which is the focus of this chapter. The notations used in this section were defined in Section 8.2.1. The generic equations are available in Appendix D.2.

9.2.1 State vector and covariance

The current estimate of the robot position and map features will be stored in the state system vector $\mathbf{x}$ and the associated uncertainty in the covariance matrix $\mathbf{P}$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_v \\ \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_m \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathbf{x}_v\mathbf{x}_v} & \mathbf{P}_{\mathbf{x}_v\mathbf{m}_1} & \mathbf{P}_{\mathbf{x}_v\mathbf{m}_2} & \cdots & \mathbf{P}_{\mathbf{x}_v\mathbf{m}_m} \\ \mathbf{P}_{\mathbf{m}_1\mathbf{x}_v} & \mathbf{P}_{\mathbf{m}_1\mathbf{m}_1} & \mathbf{P}_{\mathbf{m}_1\mathbf{m}_2} & \cdots & \mathbf{P}_{\mathbf{m}_1\mathbf{m}_m} \\ \mathbf{P}_{\mathbf{m}_2\mathbf{x}_v} & \mathbf{P}_{\mathbf{m}_2\mathbf{m}_1} & \mathbf{P}_{\mathbf{m}_2\mathbf{m}_2} & \cdots & \mathbf{P}_{\mathbf{m}_2\mathbf{m}_m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{\mathbf{m}_m\mathbf{x}_v} & \mathbf{P}_{\mathbf{m}_m\mathbf{m}_1} & \mathbf{P}_{\mathbf{m}_m\mathbf{m}_2} & \cdots & \mathbf{P}_{\mathbf{m}_m\mathbf{m}_m} \end{bmatrix}$$

$\mathbf{x}_v$ is the robot position estimate in Cartesian coordinates, $\mathbf{m}_i = [x_i \;\; y_i]^\top$ is the estimate of the location of feature i in the environment. These coordinates are expressed with respect to the world frame. As will be detailed in Section 9.3, we represent our map by points in the environment. We assume that the mobile robot has a smooth motion and choose a constant velocity, constant angular velocity model. This means that on average we expect undetermined accelerations to occur with a Gaussian profile. Let $[v_x, v_y]$ be the linear velocity of the robot and ω its angular velocity. These terms are added to the position state vector:

$$\mathbf{x}_v = \begin{bmatrix} x \\ y \\ \theta \\ v_x \\ v_y \\ \omega \end{bmatrix}$$

The initial position of the robot is chosen at the origin of the world frame and is known exactly. We also assume that the robot starts with no initial speed. At the beginning, no features have been detected in the environment, so the initial values of the filter are:

$$\mathbf{x}_v = \mathbf{0}_{6\times 1}, \quad \mathbf{P} = \mathbf{0}_{6\times 6}$$

9.2.2 Prediction: time update equation

We make the assumption that at each time step, an unknown acceleration $\mathbf{a}$ and angular acceleration α of zero mean and Gaussian distribution add noise to the state prediction in the form of an impulse in velocity $[a_x\Delta t, a_y\Delta t]$ and angular velocity $\alpha\Delta t$. The vector representing the noise will be noted $\mathbf{n}$:

$$\mathbf{n} = \begin{bmatrix} a_x\Delta t \\ a_y\Delta t \\ \alpha\Delta t \end{bmatrix}$$

The state update equation f_v and associated Jacobians can then be written:

$$f_v(\mathbf{x}_v) = \begin{bmatrix} x + v_x\Delta t \\ y + v_y\Delta t \\ \theta + \omega\Delta t \\ v_x \\ v_y \\ \omega \end{bmatrix} + \begin{bmatrix} \mathbf{n}\Delta t \\ \mathbf{n} \end{bmatrix}, \quad \nabla_{\mathbf{x}_v} f_v = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3\Delta t \\ \mathbf{0}_{3\times 3} & \mathbf{I}_3 \end{bmatrix}, \quad \nabla_{\mathbf{n}} f_v = \begin{bmatrix} \mathbf{I}_3\Delta t \\ \mathbf{I}_3 \end{bmatrix}$$

The prediction obtained by the update equation is accompanied by an increase in state uncertainty noted $\mathbf{Q}_v$. Let $\mathbf{P}_n$ be the uncertainty of the noise vector $\mathbf{n}$:

$$\mathbf{Q}_v = (\nabla_{\mathbf{n}} f_v)\mathbf{P}_n(\nabla_{\mathbf{n}} f_v)^\top$$

The values for $\mathbf{P}_n$ are chosen according to the type of motion expected. High values will be given to $\mathbf{P}_n$ if we expect big accelerations to occur. Good measurements will then be needed to reduce the uncertainty. Small values for $\mathbf{P}_n$ on the contrary indicate we expect smooth motion, the uncertainty will then be reduced but the model will be unable to cope with sudden accelerations.

The full time update equations are:

$$\begin{array}{rcl} \mathbf{x}_{t+1/t} & = & f_v(\mathbf{x}_{t/t}) \\ \mathbf{P}_{t+1/t} & = & (\nabla_{\mathbf{x}} f_v)_{t/t}\mathbf{P}_{t/t}(\nabla_{\mathbf{x}} f_v)_{t/t}^\top + \mathbf{Q}_{v,t} \end{array}$$

9.2.3 Measurement prediction and correction: measurement update equation

In this chapter, we decided to consider only the laser measurements and not the odometry as accurate odometry is not always available on mobile robots. The laser makes range-bearing observations of the landmarks. The observation of feature i can be predicted from $\mathbf{x}_{t+1/t}$:

$$\begin{array}{rcl} \mathbf{z}_{i,t+1/t} & = & h(\mathbf{x}_{t+1/t}) \\ & = & \begin{bmatrix} d \\ \arctan(\frac{y_{i,t+1/t}-y_{t+1/t}}{x_{i,t+1/t}-x_{t+1/t}}) - \theta_{t+1/t} \end{bmatrix} \\ \nabla_{\mathbf{x}_v} h & = & \begin{bmatrix} -\frac{x_{i,t+1/t}-x_{t+1/t}}{d} & -\frac{y_{i,t+1/t}-y_{t+1/t}}{d} & 0 \\ \frac{y_{i,t+1/t}-y_{t+1/t}}{d^2} & -\frac{x_{i,t+1/t}-x_{t+1/t}}{d^2} & -1 \end{bmatrix} \\ \nabla_{\mathbf{m}_i} h & = & \begin{bmatrix} \frac{x_{i,t+1/t}-x_{t+1/t}}{d} & \frac{y_{i,t+1/t}-y_{t+1/t}}{d} \\ -\frac{y_{i,t+1/t}-y_{t+1/t}}{d^2} & \frac{x_{i,t+1/t}-x_{t+1/t}}{d^2} \end{bmatrix} \\ \nabla_{\mathbf{m}_j, j\neq i} h & = & \mathbf{0}_{2\times 2} \end{array}$$

with $d = \sqrt{(x_{i,t+1/t} - x_{t+1/t})^2 + (y_{i,t+1/t} - y_{t+1/t})^2}$ the distance predicted from the robot position and estimated landmark location.

Let $\mathbf{z}_{i,t+1}$ be the measurement obtained at time $t+1$. The innovation noted $\mathbf{v}$ corresponds to the amount of unpredicted information obtained from the new measurement:

$$\mathbf{v}_{t+1} = \mathbf{z}_{i,t+1} - \mathbf{z}_{i,t+1/t}$$

The covariance between the true and measured value, called innovation covariance, is then:

$$\begin{array}{rcl} \mathbf{S}_{i,t+1} & = & (\nabla_{\mathbf{x}} h)_{t/t}\mathbf{P}_{t+1/t}(\nabla_{\mathbf{x}} h)_{t/t}^\top + \mathbf{R} \\ & = & (\nabla_{\mathbf{x}_v} h)\mathbf{P}_{\mathbf{x}_v\mathbf{x}_v}(\nabla_{\mathbf{x}_v} h)^\top + 2(\nabla_{\mathbf{x}_v} h)\mathbf{P}_{\mathbf{x}_v\mathbf{m}_i}(\nabla_{\mathbf{m}_i} h)^\top + (\nabla_{\mathbf{m}_i} h)\mathbf{P}_{\mathbf{m}_i\mathbf{m}_i}(\nabla_{\mathbf{m}_i} h)^\top + \mathbf{R} \end{array}$$

$\mathbf{R}$ is the measurement noise and depends on the sensor.

The innovation covariance can be used for data association under small uncertainty (eg. between t and $t+1$). This is commonly done through a Mahalanobis distance test:

$$\mathbf{v}_{t+1}^\top \mathbf{S}_{t+1}^{-1} \mathbf{v}_{t+1} < \chi^2_{d,\alpha}$$

with $d = \dim(\mathbf{v})$ ($d = 2$ in our case) and α the desired confidence level. This test can be used for choosing between measurements but also for accepting new values that are sufficiently far from other

features to avoid future confusion. We will refer to this method as "nearest neighbour gating" (NNG). We may note that under strong uncertainty and with features with low saliency, it is preferable to use a joint compatibility test [Neira and Tardos, 2001]. This test is however more computationally expensive and by using vision we can obtain reliable correspondences directly.

The final measurement update equations are:

$$\begin{aligned} \mathbf{x}_{t+1/t+1} &= \mathbf{x}_{t+1/t} + \mathbf{W}_{t+1}\mathbf{v}_{t+1} \\ \mathbf{P}_{t+1/t+1} &= \mathbf{P}_{t+1/t} - \mathbf{W}_{t+1}\mathbf{S}_{t+1}\mathbf{W}_{t+1}^{\top} \\ \mathbf{W}_{t+1} &= \mathbf{P}_{t+1/t}(\nabla_{\mathbf{x}}h)_{t+1/t}^{\top}\mathbf{S}_{t+1}^{-1} \end{aligned}$$

$\mathbf{W}$ is the Kalman gain and indicates how much trust we can have in the measurements.

9.3 Laser-vision features

In the previous section, we described how a probabilistic representation of the environment can be obtained by integrating motion and measurement uncertainty with the Extended Kalman Filter. We will now explain how we obtain the points that form the map.

When using laser range scans for metric-based SLAM (as opposed to occupancy grid), corners or lines are generally extracted and form the map. Lines are well adapted to indoor environments but with corners no piecewise linear assumptions need to be made on the topology of the environment. Figure 9.1 illustrates possible features that could be used for SLAM. The problem with this approach is that the number of points can be very low (eg. in corridors) as previously remarked by Lu and Milios [Lu and Milios, 1997a]. These points might also be unreliable if they belong to regions with a strong incidence angle. Ideally we would want to work with laser points measured in a region with a low incidence angle and that are salient. This is contradictory when working with laser information alone. However with visual information we can obtain such points as we will see in this section. These points will be used as landmarks in the map.

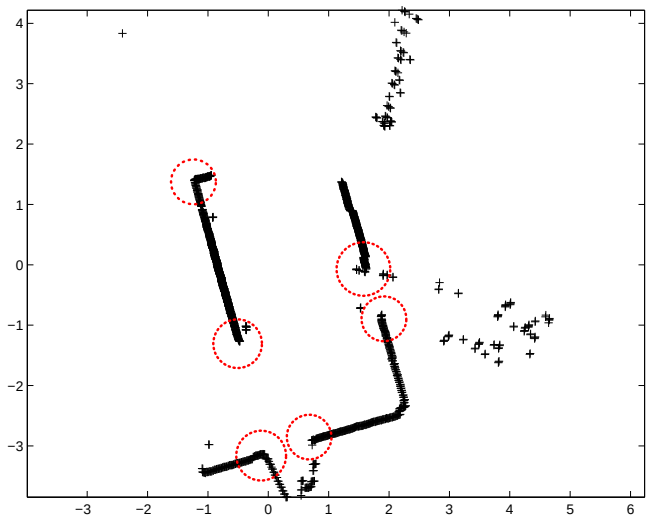


Figure 9.1: Possible SLAM features extracted from a laser scan

9.3.1 Algorithm for extracting salient points

Algorithm 2 describes the extraction process to obtain salient points from a laser scan. We start by removing points with strong incidence angles that could be unreliable and could belong to regions with clutter. We then look for edges in the image that intersect the laser trace[1] forming an angle as close as possible to 90° and with a strong gradient magnitude. These points form a "cross" between an edge in the image and the laser trace. Under the assumption of planar motion, these points can be easily re-observed by looking (eg. using correlation) along the laser trace.

Extracting the normal to the laser scan The normal to the laser scan is calculated using several points for robustness. Let m be the number of points (typically $m \sim 5$). To find the normal (n_x, n_y), we solve the following least-squares equation for m laser points (s_x, s_y):

$$\begin{cases} \min_{n_x,n_y,d} \sum_{i=1}^{m} (s_x^i n_x + s_y^i n_y - d)^2 \\ n_x^2 + n_y^2 = 1 \end{cases}$$

This problem can be solved in closed form. Let:

$$\overline{x} = \frac{1}{m}\sum_{i=1}^{m} s_x^i \qquad \overline{y} = \frac{1}{m}\sum_{i=1}^{m} s_y^i$$

$$S_{xy} = \sum_{i=1}^{m}(s_x^i - \overline{x})(s_y^i - \overline{y}) \qquad S_{x^2} = \sum_{i=1}^{m}(s_x^i - \overline{x})^2 \qquad S_{y^2} = \sum_{i=1}^{m}(s_y^i - \overline{y})^2$$

The least-squares solutions are then:

$$\begin{aligned} n_x &= -sign(S_{xy})\sqrt{\frac{1}{2}\left(1 + \frac{S_{y^2} - S_{x^2}}{\sqrt{(S_{y^2} - S_{x^2})^2 + 4S_{xy}^2}}\right)} \\ n_y &= \sqrt{\frac{1}{2}\left(1 - \frac{S_{y^2} - S_{x^2}}{\sqrt{(S_{y^2} - S_{x^2})^2 + 4S_{xy}^2}}\right)} \\ d &= \overline{x}n_x + \overline{y}n_y \end{aligned}$$

[1] we call "laser trace" the reprojection of the laser scan in the image

Algorithm 2: Laser-vision salient point extraction

Data: Current image $\mathcal{I}$ and current laser scan $\mathcal{S}$, Thresholds:

- e_{inc}: maximal incidence angle,
- e_{ang}: maximal angle between the image gradient and the normal to the laser curve,
- e_{magn}: minimal gradient magnitude.

Result: List of salient points

1. remove laser points with strong incidence angles (e_{inc}),
2. keep the points with a strong gradient magnitude (e_{magn}) and with a gradient direction close to the normal to the laser trace in the image (e_{ang}). (To add robustness to the approach, Gaussian smoothing is applied to the image before the extraction.)

Figure 9.2 shows an example of the reprojection of a laser scan in the omnidirectional image (i.e. "laser trace"). In figure 9.3 are shown the normals to the laser trace: the points with a circle are rejected because of a strong incidence angle and the other points (with small crosses) are kept. We may note that this process rejects outliers, some of these being produced by the metal strips on the walls or the posters with plastic coating that diffuse the laser beam. The rejected points are also shown in the laser scan in figure 9.4[2]. We then calculate the gradient direction and magnitude and keep the points with a strong magnitude (figure 9.5 shows the points kept and the gradient direction). We then choose the points where the gradient direction and normal to the laser trace are similar (figure 9.6).

9.3.2 Improving the discriminancy of salient points

In the previous section we described an approach to define salient points. The EKF requires reliable data correspondences to insure a correct update of the robot position and state uncertainty. We will now define a measure of similarity that combines the metric and visual information and will enable to distinguish between points reliably.

A straight-forward method would be to associate a small image template around the salient points and use cross-correlation as a measure of similarity. However this would not be a good choice since the visual appearance changes with the viewpoint, this change being particularly important with omnidirectional vision.

The metric information from the laser scan can help build a descriptor invariant to changes in viewpoint. Figure 9.7 illustrates the approach. We start by defining a region around a salient point S containing all the points at a distance inferior to d_{max}. The laser scan is given an orientation from the angles, for example using the clockwise order. The interval $[-d_{max}; d_{max}]$ is partitioned into n equal bins. The signal s associated to S is then calculated from the average intensity of the ELS points belonging to each bin.

The similarity score between points is then defined as the normalised correlation between signals. Let s_i and s_j be the signals associated respectively to the points S_i and S_j, the score can be calculated

[2] there is a reflexion between the two views introduced by the mirror

Figure 9.2: Laser trace

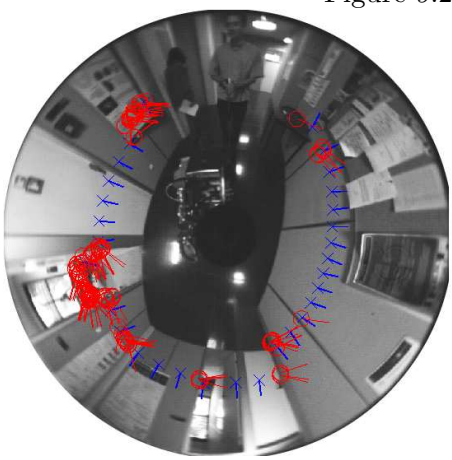


Figure 9.3: Laser trace with rejected points based on their incidence angle

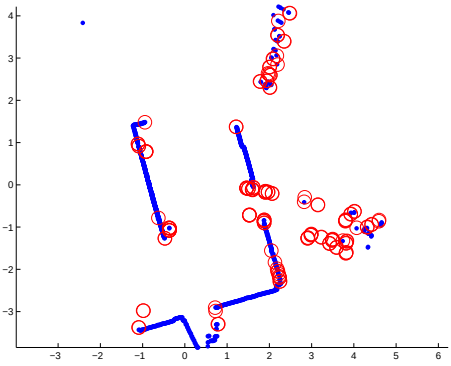


Figure 9.4: Laser scan with rejected points based on their incidence angle

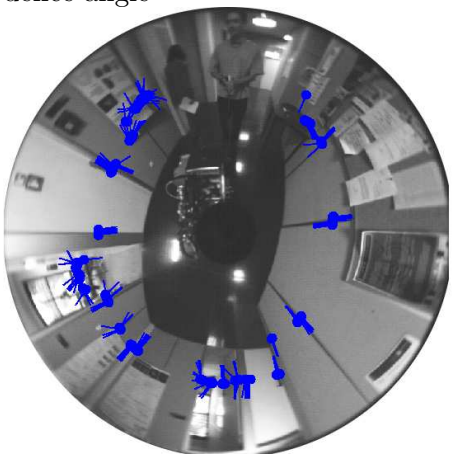


Figure 9.5: Image points with strong gradient magnitudes and belonging to the laser trace

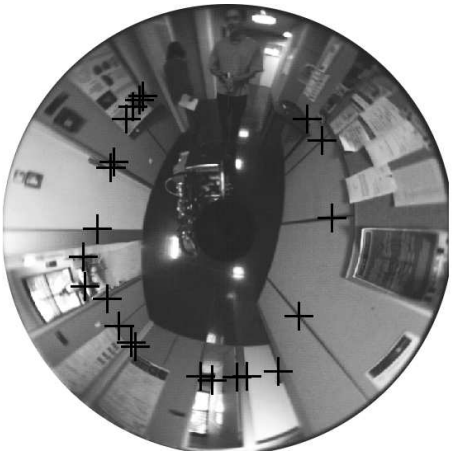


Figure 9.6: Salient points: high gradient magnitude and gradient direction orthogonal to the laser trace

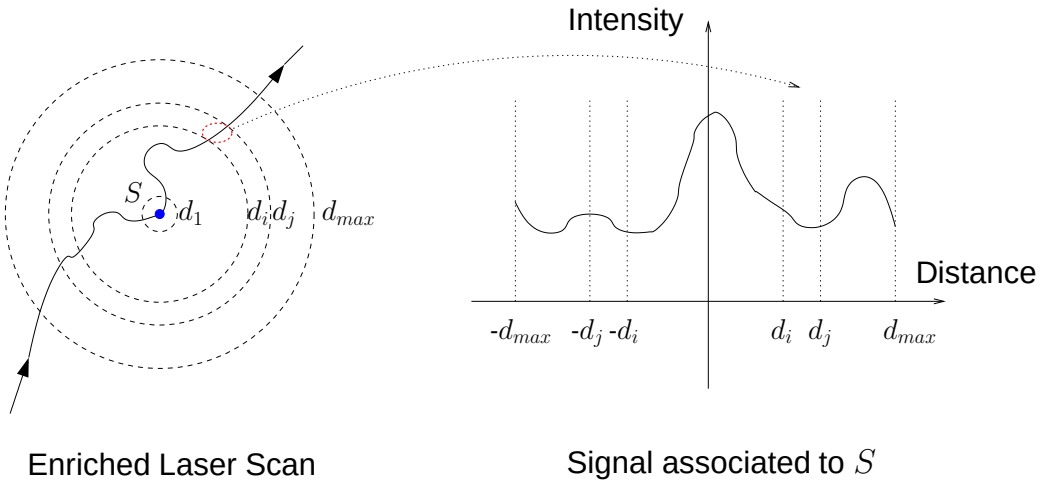


Figure 9.7: Signal associated to a salient point S

by:

$$\text{score}(S_i, S_j) = \frac{\sum_{d=-d_{max}}^{d_{max}} [s_i(d) - \overline{s_i}][s_j(d) - \overline{s_j}]}{\sqrt{\left(\sum_{d=-d_{max}}^{d_{max}} [s_i(d) - \overline{s_i}]^2\right)\left(\sum_{d=-d_{max}}^{d_{max}} [s_j(d) - \overline{s_j}]^2\right)}}$$

with $\overline{s}$ the mean of the signal.

Figures 9.8 and 9.10 show the matching of points between two views using this descriptor. We only keep the points that give best matches in both directions to improve the robustness. In this example, there are some outliers but most values were correctly matched. Figures 9.9 and 9.11 correspond to the 1D signals of point z_8 for each view. The distance axis in these figures corresponds to the division of an interval of size 0.8 m in regions of 0.04 m centered in the salient point.

This descriptor is particularly useful for distinguishing points that are close to each other and when the measure of uncertainty from the filter is not sufficient to differentiate between the values accurately.

The NNG will be used to associate features between "local" points. However when the uncertainty becomes important or the filter becomes inconsistent, the measure of uncertainty is no longer a valid way to associate data as will be explained in the next section on "loop closing".

9.4 Loop closing

The uncertainty of the robot position with respect to the initial position will accumulate as the robot explores the environment if previously mapped landmarks are not matched. Detecting previously observed features that are not in the direct vicinity of the robot has become known as loop closing. The name indicates that this situation generally occurs after a loop in the environment brings the robot back to a previously explored area. It is desirable to re-identify areas and in particular to match features as this will reduce the overall uncertainty but is also an essential step to ensure the completeness of the exploration.

The filter uncertainty does not provide a reliable way to match points after a long period. The linearisation introduced in the Kalman filter and unpredicted measurement errors can render the filter

Figure 9.8: Salient points matched between two views using 1D signal descriptor

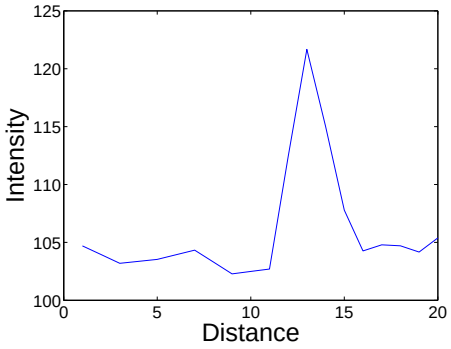


Figure 9.9: Example of 1D signal corresponding to z_8 in first image

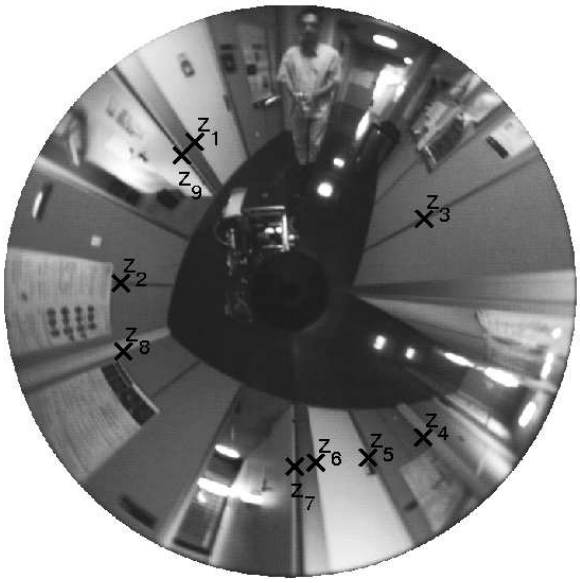


Figure 9.10: Salient points matched between two views using 1D signal descriptor

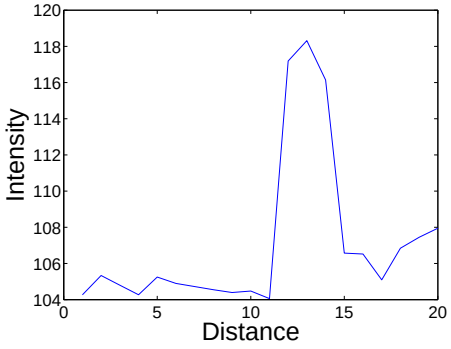


Figure 9.11: Example of 1D signal corresponding to z_8 in second image

inconsistent. In the literature, the problem of missing a loop closure is often emphasised. Another possibility, particularly present in punctual feature-based mapping (as opposed to lines), is the detection of a loop based on bad data association. Figure 9.12 illustrates a loop closure that was "activated" by a single feature. After this incorrect association, the robot can then no longer register the subsequent values correctly. Combining several features can improve the situation but if the filter becomes inconsistent, an incorrect loop closure could take place anyway. For this reason, we propose to explicitly control the data association through a topological representation of the explored region as will be detailed in Section 9.4.1.

Section 9.4.2 will describe ways of recognising areas from images and the image descriptor chosen for this study. Finding a previously visited site is not sufficient for loop closing, we also need to match the features reliably. This will be done in two step. First we estimate the rotation between views based on the visual information, then we find the relative pose and associate the features using scan matching. The scan matching algorithm itself will be described in the subsequent section (Section 9.5).

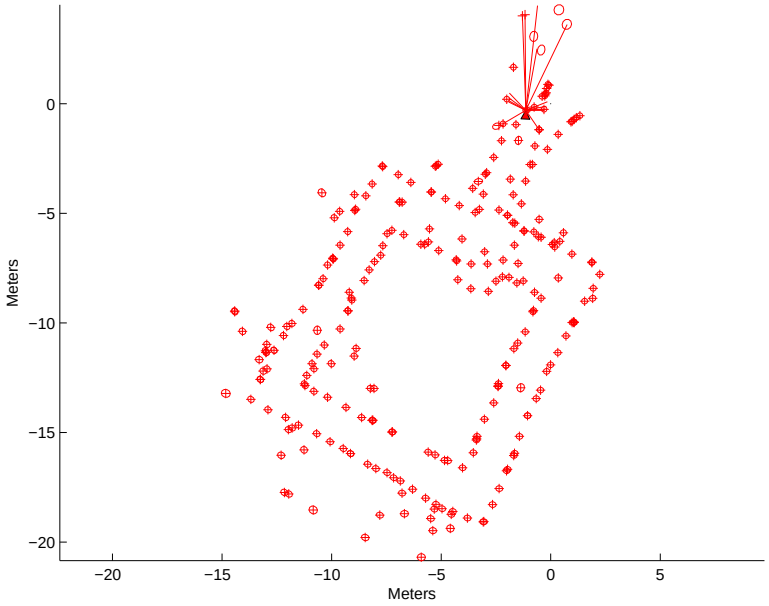


Figure 9.12: Incorrect loop closure

9.4.1 Controlling loop closing

As explained previously, data association cannot be made reliably after "long" periods using only the predictions from the Kalman filter. For this reason, we have chosen a conservative approach to data association and loop closing. We only accept to match points with the innovation covariance (NNG) "locally" and control the data association between "old" points. We will now clarify what we mean by "locally" and "old". The described approach is generic and does not make any assumptions on the descriptors used to identify the areas. The choice of descriptor for this study will be detailed in Section 9.4.2.

The proposed approach is based on a topological representation. The graph is build iteratively by adding nodes with a unique identifier and a descriptor. These nodes will also be called *key frames*. A

new key frame is added when it is significantly different from the key frames at a graph distance of two or less, according to a measure defined over the key frame descriptor. Figure 9.13 illustrates this concept, key frame n is tested with regards to frames $1, 2, 3, 4$. If the value is different to the adjacent key frame but similar to values at a graph distance of two, we assume we have returned to a previously explored area. For example, in the image, if the value of n is different to 2 but similar to 1, we assume we are in 1. An incorrect topological association does not affect the metric map directly.

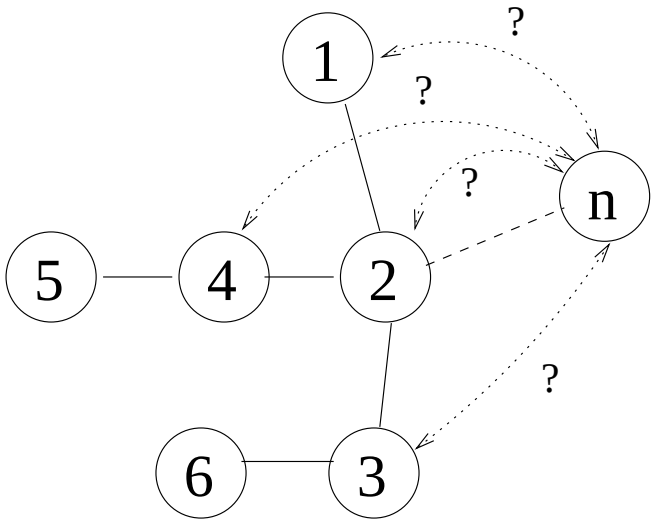


Figure 9.13: Adding a new key frame n to the topological map

Metric information is added to the arcs between vertices to define a "topological distance" (this metric information does not need to be precise and is not updated).

When a new feature is observed, it is tagged by the current key frame identifier. When a feature is re-observed, two situations occur:

1. The feature is tagged with the current key frame identifier or it is "topologically close" to the previous identifier in that case it is accepted. The topological distance is defined as the metric distance in the graph. This distance is different from the distance taken in the metric map: before loop closing, the points can be close in the metric map but the topological distance is then the distance of the entire loop.

2. It is "topologically far" from the key frame and is considered as "old" and not matched. The only situation were an "old" feature can be matched to a new feature is through loop closing. Loop closing is tested each time a new key frame is created. This means we assume that changes in the environment measured by the descriptor occur in similar places which is a reasonable assumption.

Figure 9.14 illustrates the approach. In the first image, the topological distance between key frame 7 and key frame 2 is important compared to the metric distance. Points viewed in 7 would be prevented from being automatically matched using the innovation covariance, the matching would require explicit loop closing. In the second image, the loop has been closed and the points in 7 are now topologically close to the points in 2 or 1 and could be automatically matched.

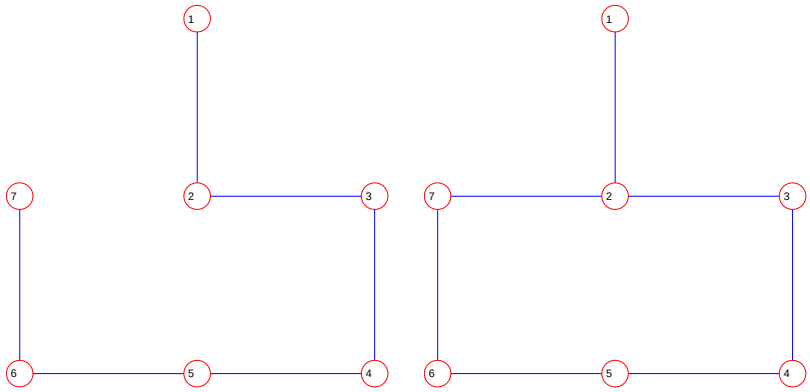


Figure 9.14: Example of a topological graph *before* and *after* loop closing

9.4.2 Recognising scenes

9.4.2.1 Image descriptors

Recognising places has been a strong incentive for using panoramic vision in robotics. For planar motion (and also to a certain extend for 3D motion), it is simple to design descriptors that are rotationally invariant.

For loop closing, we are interested by image retrieval systems that are adapted to the iterative mapping process. This rejects methods such as the principal component analysis that requires the whole data to find classes that separate the data according to a certain criteria (the variance over a linear basis for the PCA). We would also want the algorithm to be robust to occlusion, robust to changes in illumination and fast to compute. False positives are particularly detrimental for a loop closing algorithm. However our approach will not be based solely on visual clues, we will also be using the metric values obtained by the laser. In other words, we can trade off some of the discriminancy of the image identifiers for computational speed for example. We may also argue that ambiguity is inevitable and should be taken into account in the filter by using multiple hypothesis. This was not considered in this chapter where our main focus is showing the advantages of combining vision and laser data under a simple EKF framework.

Image indexing and registration are active fields of research. Features with invariance properties (invariance to affine transformations, illumination...) have lead to impressive results [Yang et al., 2006]. Popular methods include SIFT [Lowe, 2004] or affine-invariant multiscale Harris corners [Mikolajczyk and Schmid, 2004]. The advantage with feature-based methods for place recognition is the possibility to apply geometric constraints that greatly diminish the false positives. These approaches are however still difficult to use on real-time systems even though some progress has been made using for example GPUs [Sinha et al., 2006]. An alternative that is particularly valid for panoramic cameras is the use of image descriptors. Fourier transforms or zero phase representation [Pajdla and Hlavac, 1999] have been used previously but are not robust to occlusion. What may seem surprising at first is that image histograms have shown to give surprisingly good results at a very low computational cost in particular on colour images [Ulrich and Nourbakhsh, 2000]. They have the advantage of being invariant to rotation for planar motion and robust to outliers. The histograms can either be taken on the image pixels directly or on local attributes for better results [Gonzalez-Barbosa and Lacroix, 2002]. Haar invariant features [Siggelkow and Burkhardt, 2002; Charron et al., 2005, 2006] that add local invariance to translation can help improve histogram-based approaches.

In our implementation, we tried a different measure from the field of image indexing: correlograms [Huang et al., 1997].

9.4.2.2 Correlograms

As explained by the authors [Huang et al., 1997]: informally, a correlogram is a table indexed by grayscale (or color) pairs where the k-th entry for $< i, j >$ specifies the probability of finding a pixel of value j at a distance k from a pixel of value i. A pixel-based histogram only represents the values present in the image. Correlograms describe globally how the pixel values are locally distributed. This enables robustness to large changes in appearance and partial occlusion but is more discriminative than a simple histogram. We may note that correlograms are not restricted to pairing pixel values but can also be defined on local attributes such as the ones proposed by Barbosa [Gonzalez-Barbosa and Lacroix, 2002]. For our work, the grayscale values proved sufficient.

Let $\mathcal{I}$ be an image of size $n \times n$ with m different values v_i. If $\mathbf{p}_i$ is an image coordinate $v_i = \mathcal{I}(\mathbf{p}_i)$ is its value. $\mathcal{I}_v = \{\mathbf{p} \mid \mathcal{I}(\mathbf{p}) = v\}$ corresponds to the set of image coordinates with value v. Let k be a given distance, the correlogram is defined as:

$$\gamma^{(k)}_{v_i,v_j} = \underset{\mathbf{p}_1 \in \mathcal{I}_{v_i}, \mathbf{p}_2 \in \mathcal{I}}{P}(\mathbf{p}_2 \in \mathcal{I}_{v_i} \mid |\mathbf{p}_1 - \mathbf{p}_2| = k)$$

We may note the similarity with the cooccurrence matrix [Haralick et al., 1973] defined for texture analysis.

We will note d the number of distances (k values) considered. The correlogram has a size of $O(m^2 d)$ which makes it difficult to use as such. Generally the auto-correlogram of size $O(md)$ is used:

$$\alpha^{(k)}_c = \gamma^{(k)}_{c,c}(\mathcal{I})$$

In [Huang et al., 1997] efficient ways of calculating the correlogram are discussed that lead to an algorithm with a complexity of $O(n^2 d)$. To obtain a descriptor size compatible with real-time queries and large environments, the grayscale images were reduced to 64 values (instead of 256) and we chose $\mathbf{d} = [1, 3, 5, 10]$ for the k values generating a descriptor of only 256 values.

When working on histograms or descriptors, several distances are possible [Rubner et al., 1998; Ulrich and Nourbakhsh, 2000; Gonzalez-Barbosa and Lacroix, 2002]. The results presented in the articles on panoramic vision show similar classification results for the Jeffrey divergence, χ^2 statistic and earth mover's distance. The χ^2 is particularly cheap to compute which motivates its use for real-time robotic applications. We chose to use the symmetric χ^2 histogram distance [Schiele and Crowley, 2000] between two histograms $\mathbf{h}$ and $\mathbf{g}$:

$$d(\mathbf{h}, \mathbf{g}) = \sum_i \frac{(h_i - g_i)^2}{h_i + g_i}$$

We tested the discriminancy of the auto-correlogram on our loop closure sequence. In the figures, the reference auto-correlogram is indicated by a red cross, the blue bars indicate the probability that the values come from the same distribution according to the χ^2 distribution. Figure 9.15 shows the response in the vicinity of the loop closure. The region is correctly identified on the way back. An ambiguous case is depicted in figure 9.16 with the corresponding images in figures 9.17 and 9.18. These images are difficult to distinguish and we conjecture that colour images would help in this situation. However there will always be ambiguities and a full SLAM implementation would probably use delayed Kalman filtering [Newman et al., 2006] or particle filters [Montemerlo, 2003].

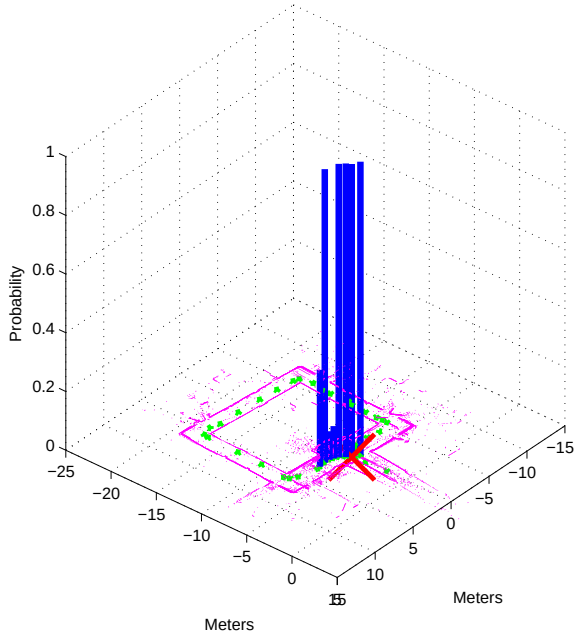


Figure 9.15: Detection of a loop closure situation

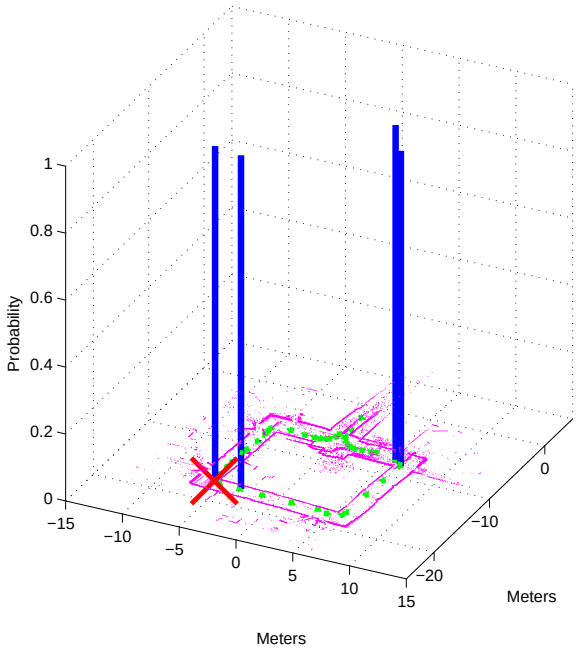


Figure 9.16: Ambiguous localisation

Figure 9.17: Image corresponding to the ambiguous match

Figure 9.18: Image corresponding to the ambiguous match

To summarise our approach, our loop closure detection framework will rely on a database of auto-correlograms built iteratively. A new key frame will be added each time the difference with the previous pose is significant according to the χ^2 test. To check for a possible loop closure, we will calculate the symmetric χ^2 distance over the database. As only 256 values are used, the calculation is particularly fast. To give an order of magnitude, on a Pentium IV CPU 3.60 GHz, a lookup over 10^4 key frames will take about 200 ms.

Finding a possible loop closure is not sufficient, we must also match the features to update the filter. In the next section, we describe an approach to estimate the rotation between the views. This will help initialise dense scan matching that will be used to associate the landmarks.

9.4.2.3 Estimating the rotation for point matching

Finding the rotation between two panoramic images has been done previously through correlation with the Fourier transform [Pajdla and Hlavac, 1999]. This approach proved too sensitive to occlusion in our loop closing tests. We designed a more *ad hoc* method that gave experimentally satisfying results.

The image is first reprojected onto a cylinder and divided into several bands (figures 9.21 and 9.22). A histogram is calculated for each region. To recover the rotation between two different views, we calculated the "best" association between histograms. Several measures are possible, for example combining the χ^2 distance over all the histograms. Figures 9.19 and 9.20 correspond to a loop closing situation, we may note that the change in viewpoint and the occlusion requires the use of a robust approach. Figures 9.21 and 9.22 show the projection of the images in a cylindrical view. The images are divided in regions of same size and a histogram is calculated for each band. In our experiments the histograms were taken over 64 graylevels (instead of the full 256) and 30 bands were used. This gives a precision of an order of $\sim 360/30 = 12^\circ$. We assumed that the precision was of about 30° after the rotation estimation. There is of course a risk in trying to extract quantitative measures from an approach that is mainly qualitative, however in a difficult loop closing situation, this method can provide important boundaries on the relative orientation. We may note that SIFT points did not give any matches between these two views (except on the robot).

To confirm that this approach does improve matching even with the large interval of uncertainty, we reproduce the matching example of Section 9.3.2 (the images are reproduced alongside the new results for ease of comparison). Figures 9.24 and 9.25 show the matches between two views without angular boundaries and figures 9.26 and 9.27 when we add the angular constraints obtained by our rotation estimation algorithm with an interval of $\pm 30^\circ$. In the first case, 9 values were matched with 2 outliers. In the second case, 11 values were matched with only 1 outlier which confirms the usefulness of this pre-processing step.

9.4.2.4 Loop closing strategy

The proposed loop closing strategy represents the environment by a topological map with a set of image descriptors representing regions or sites. The first descriptor helps identify a possible loop closure situation. The second estimates the rotation between views. We will now discuss how to associate landmarks to update the filter.

We studied two strategies to find the correspondence between landmarks. The first consists in using points matched as described in Section 9.3.2. This approach gave satisfying results between adjacent frames. However in a loop closing situation, part of the environment is likely not to be mapped and the points can then be very sparse. A dense approach is thus desirable. Associating whole scans to our image descriptors is not satisfying in terms of memory usage. Instead we associated "parts" of

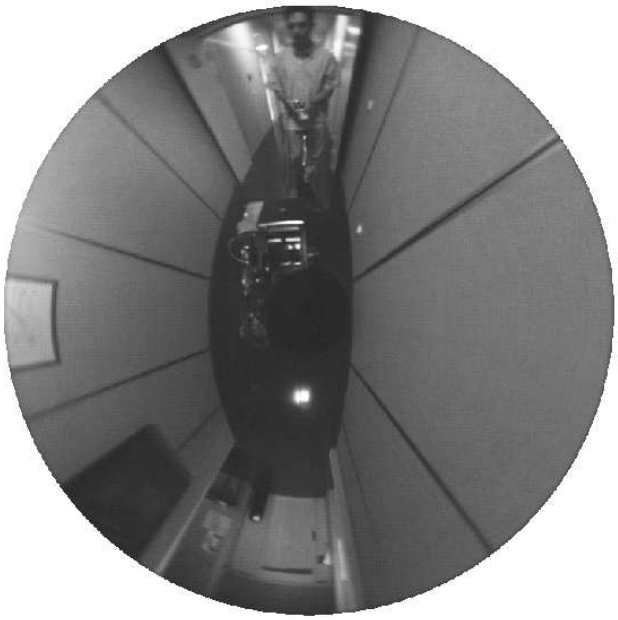


Figure 9.19: Image 430 of a loop closing sequence

Figure 9.20: Image 1490 of a loop closing sequence

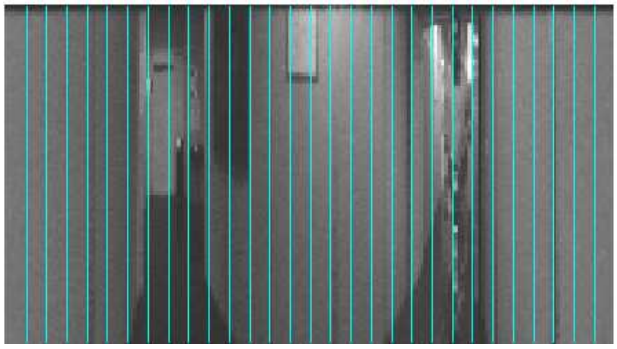


Figure 9.21: Image 430 reprojected on a cylindrical view and divided for calculating histograms

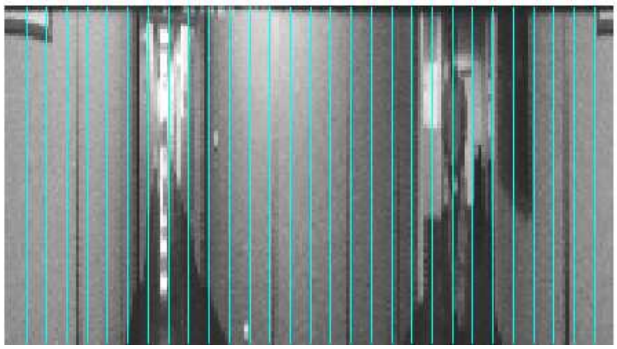


Figure 9.22: Image 1490 reprojected on a cylindrical view and divided for calculating histograms

Figure 9.23: Image 430 rectified according to the estimated rotation between views

Figure 9.24: Matched points between image 640 and 645 without angular boundaries

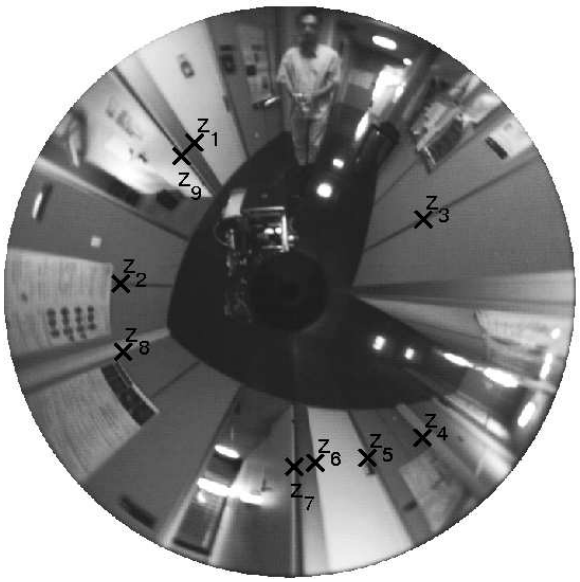


Figure 9.25: Matched points between image 645 and 640 without angular boundaries

Figure 9.26: Matched points between image 640 and 645 with angular boundaries

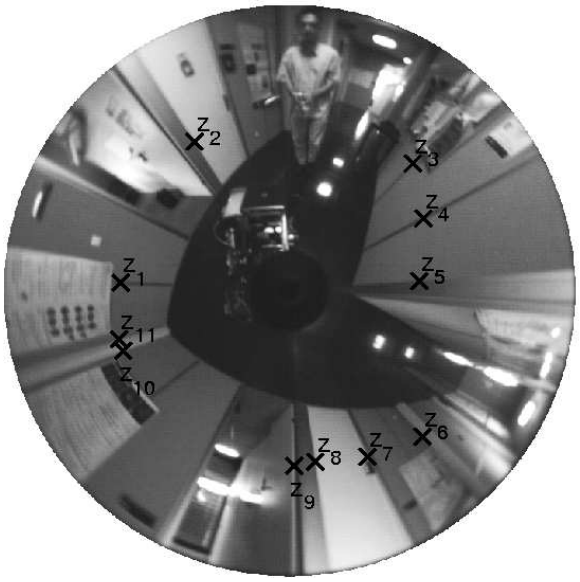


Figure 9.27: Matched points between image 645 and 640 with angular boundaries

a scan to the *landmarks*. When testing for loop closing, we reconstruct two scans from the points topologically close to the two regions and apply the scan matching algorithm. This provides stronger metric constraints than sparse features and gave experimentally satisfying results. After minimisation, we associate the feature points based on a distance threshold and the feature descriptors defined in Section 9.3.2.

9.4.3 Summary of the loop closing approach

Algorithm 3 summarises the key frame creation mechanism, the control of the loop closing and the data association process.

Algorithm 3: Loop closing and feature association

Data: Current image $\mathcal{I}$ and current laser scan $\mathcal{S}$, topological map $\mathcal{T}$, list of SLAM features l_z, state vector $\mathbf{x}$ and covariance $\mathbf{P}$ after the prediction step.
Result: Update of l_k, l_z, $\mathbf{x}$ and $\mathbf{P}$

z = extractCurrentFeatures($\mathcal{I}$,$\mathcal{S}$) `// Extract and tag salient points, build 1D signal`

$[l_a, l_n]$ = associateFeatures(z,l_z,$\mathbf{x}$,$\mathbf{P}$,$\mathcal{T}$) `// Associate features through NNG and topological distance.` l_a `contains the feature associations,` l_n `the new features.`

$[\mathbf{x}, \mathbf{P}]$ = updateFilter($\mathbf{x}$,$\mathbf{P}$,l_a) `// Use measurement equation to update filter`

$[\mathbf{x}, \mathbf{P}, l_z]$ = addNewFeatures(l_n,$\mathbf{x}$,$\mathbf{P}$) `// Add new features with their covariances`

$[b_{new\_keyframe}, \mathcal{T}]$ = isNewKeyframe($\mathcal{I}$,$\mathcal{T}$) `// Test if the current image is a new key frame, if this is the case build orientation histograms and add the values to` $\mathcal{T}$

if $b_{new\_keyframe}$ **then**
  $[b_{loop}, l_a]$ = testLoopClosure(l_z,$\mathcal{T}$) `// Test for loop closure and build feature associations, the topological map and the list of features are used to build the scans in the vicinity of the detected loop closure`
  if b_{loop} **then**
    $[\mathbf{x}, \mathbf{P}, l_z]$ = updateFilterLoop($\mathbf{x}$,$\mathbf{P}$,l_z,l_a) `// Use measurement equation to the update filter and remove multiple values by backtracking over tags`
  end
end

9.5 Laser scan matching with vision

In the previous section on loop closing, we proposed a method to identify a previously visited area and estimate the rotation between the views. The final data association step consists in matching the features. The sparsity of the available data when closing a loop encourages the use of dense methods

which is the object of this section.

Scan matching is the process of finding the relative position between two laser scans. It is an important step in many map building frameworks. The contribution of this section is to extend the standard Iterative Closest Point (ICP) to include the vision information.

9.5.1 Different scan matching approaches

We will now discuss methods for tracking laser scans without explicit data association[3].

The most common approach is the Iterative Closest Point (ICP) [Besl and McKay, 1992; Chen and Medioni, 1991] which consists in minimising iteratively a distance between two scans, we will discuss this approach more in detail in this section.

Scan correlation is used frequently when building occupancy grid maps [Elfes, 1989]. The optimal pose of an incoming scan is obtained by a correlation search over the rotation and translation. This can be particularly costly and predefined bounds are generally needed for the calculation to be computationally feasible.

An alternative minimisation approach to correlation that relies on a similar concept to occupancy grids was proposed by Biber [Biber et al., 2004]. The values in the grid however no longer indicate the probability of occupancy but a probability of correctness. The incoming scan is then registered by an iterative minimisation. This work has also been extended to match several scans simultaneously [Biber and Strasser, 2006] using an energy function defined on a grid. The interesting aspect of this method is the efficient and robust iterative minimisation approach. It has however some shortcomings, an overlap is needed for the the scans to register. This would make it difficult to match part of a scan to a bigger scan. Furthermore the probability of correctness is not associated to the sensor model so the measure of uncertainty obtained from the minimisation would not be adequate for an EKF-SLAM framework.

9.5.2 Iterative closest point with vision information

We propose to extend the work by Lu and Milios [Lu and Milios, 1997b] to take into account the visual information. The ICP for scan registration between a reference scan $\mathcal{S}_R$ and a current scan $\mathcal{S}_C$ is generally composed of the following two steps:

(1) each point of $\mathcal{S}_R$ is associated to the closest point of $\mathcal{S}_C$ (closest in the general sense, different metrics are possible)

(2) the distance error is minimised. In the case of points, a closed form solution exists. Step (1) is repeated until convergence.

If the metric chosen is the Euclidean distance between n points $\mathbf{s}_i^r$ from the reference scan and n points $\mathbf{s}_i^c$ from the current scan, at each iteration step, we solve the least-squares problem (a closed form solution exists):

$$\min_{\mathbf{t},\theta} \sum_{i=1}^{n} (\mathbf{R}(\theta)\mathbf{s}_i^c + \mathbf{t} - \mathbf{s}_i^r)^2 \tag{9.1}$$

with $\mathbf{t}$ the translation between laser scans and θ the rotation angle.

The authors combine the ICP with an extra "matching-range-point" (MRP) rule. The ICP converges slowly over the rotational component. The MRP illustrated by figure 9.28 adds a constraint

[3] these methods are generally called scan matching but they are similar to tracking approaches in computer vision

from the position that generated the scan and has a good convergence speed over the rotation. The combination of ICP and MRP leads to an efficient algorithm which can handle sensor noise and partial occlusion.

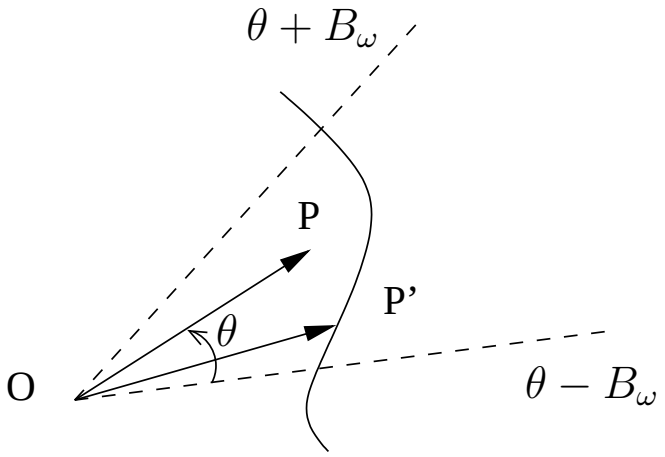


Figure 9.28: Matching-range-point rule: For a point P, the corresponding point P' on the scan lies within the interval $[\theta - B_\omega; \theta + B_\omega]$ with $\|OP'\|$ closest to $\|OP\|$ (from [Lu and Milios, 1997b])

We will call Enriched Laser Scan (ELS) the triplet (ρ, θ, i) containing the polar coordinates of the laser points (ρ, θ) and the intensity i obtained from the omnidirectional image by reprojection[4] with a bilinear approximation. The main difficulty encountered with introducing intensity in the minimisation was the non-uniform change in intensity that can introduce strong errors in particular in the rotational component of the estimation. However the intensity information improves observability. When mapping a corridor, the translation is generally poorly constrained by the laser information. Vision can help render the estimation observable. However the metric information is more reliable and should be used in priority. Based this observation, we derive the following algorithm. At each step a point P from the reference scan is mapped to three values calculated from the incoming scan:

(1) the closest point for euclidean distance (ICP)

(2) the closest point in terms of intensity (ICI)

(3) the closest point obtained by the MRP rule.

As proposed by Lu and Milios, the rotational and translational components are calculated from the MRP values and ICP values respectively. When the values given by the two components are close to the minimum (defined generally by a threshold), the translational component is calculated from the ICI values. We may note that we do not improve the convergence speed or region of convergence but the observability.

9.5.3 Experimental validation of laser-vision scan matching

We tested the algorithm for the direct estimation of the motion of the ANIS mobile robot on a corridor sequence with a loop. The only information used was the laser scan or the ELS. We decided to change

[4] we assume that the relative position between the laser and the camera is known

reference scan when the amount of points in correspondence between the reference scan and the current scan was under a predefined percentage threshold. The lower the threshold, the less information there is between the two poses. Figures 9.29 and 9.30 show the results with 80% overlap. The green triangles represent the different positions were the algorithm changed reference scan. Figures 9.31 and 9.32 correspond to the results on the same sequence with 60% overlap. These experiments show clearly the role of the visual information in constraining the translation. In figure 9.31 the estimation of the orientation is correct, only the translation is badly estimated.

Another test was done that details more precisely the effect of the visual information on the minimisation. Scan matching was undertaken on a loop closing problem obtained from real data. Figures 9.33 and 9.34 show the result after minimisation for the standard scan matching and for the ELS version respectively. The red plus sign correspond to the reference scan. The green points with circles show the current scan before minimisation and in magenta plus signs after minimisation. The current ELS has the same orientation as the standard current scan however it is shifted along the axis made by the corridor walls. The side views in figures 9.35 and 9.36 show the intensity on the z-axis. The standard scan in figure 9.35 is poorly aligned regarding the intensity compared to the case of ELS in figure 9.36. We may note however that a non-uniform change in intensity prevents the ELS from providing a perfect overlap.

Further improvements could be obtained by weighting the least-squares values according to a measure of uncertainty depending in particular on the incidence angle as in [Pfister et al., 2002].

The precision of scan matching makes it tempting to use directly as SLAM approach. However the cost of saving a scan at each iteration makes it less attractive for large scale mapping.

9.6 SLAM algorithm

We will now summarise the proposed SLAM strategy.

Salient laser-vision points were chosen to represent the environment. These can be extracted at a low computational cost from the laser and vision data. The landmark descriptor also provides reliable correspondences to be made using the standard nearest neighbour statistical gate.

During the metric map building process, we also build a topological representation that consists of key frames with image descriptors. These have a low memory consumption, each key frame contains an auto-correlogram of 256 values and 30 histograms with 64 intensity values. These values can be saved with 2 bytes (or 16 bites). Less than 50 Mb would be needed to represent 10^4 key frames. A query over a database of this size would require less than 200 ms which stays tractable in a "real-time" framework. The topological representation makes it possible to control landmark associations when the distance between frames becomes important and correspondences based on uncertainty could by unreliable.

The approach was tested in an indoor environment containing a loop. The odometry and the commands given to the robot were not used in this experiment. The sequence consists of 1500 images and laser scans acquired at 7 frames per second over a distance of 50 meters. This sequence is quite difficult because of vibrations that induced blurring and offsets in the omnidirectional image.

Figures 9.37 and 9.38 show the metric and topological map built during the exploration of the indoor sequence before loop closing. The green crosses indicate locations where a new key frame was added. We can see that the uncertainty has accumulated during the journey around the loop. The filter has become inconsistent as the 3σ uncertainty bound around the robot position does not include the previously explored region. The database query combined with the rotation estimation and scan matching was able to associate 9 features and enable loop closing. Figures 9.39 and 9.40 show the

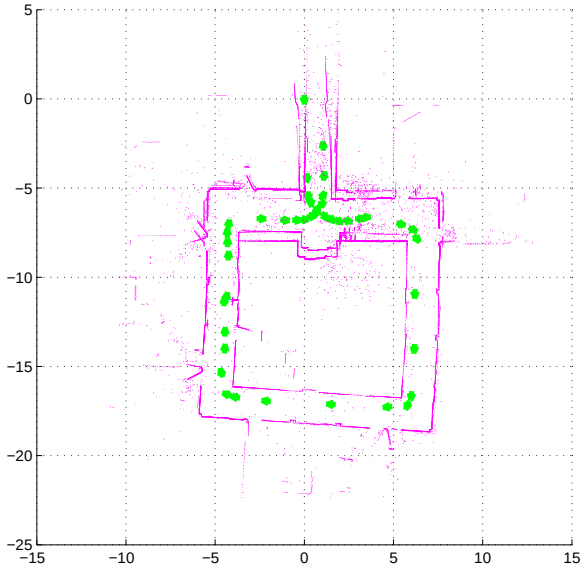


Figure 9.29: Motion estimation using only scan matching with 80% overlap between reference scans

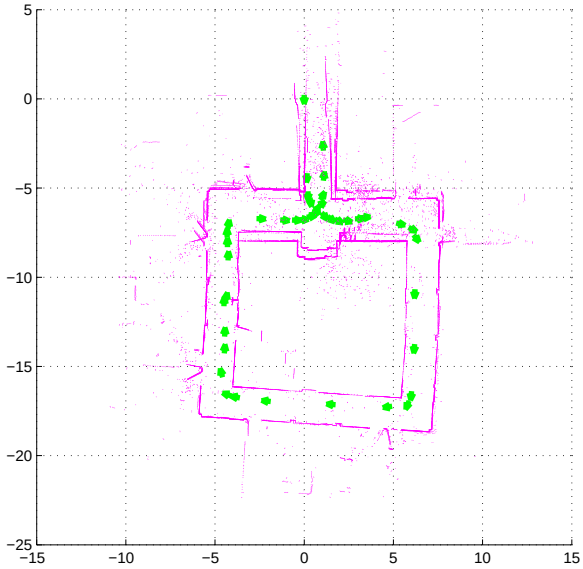


Figure 9.30: Motion estimation using ELS scan matching with 80% overlap between reference scans

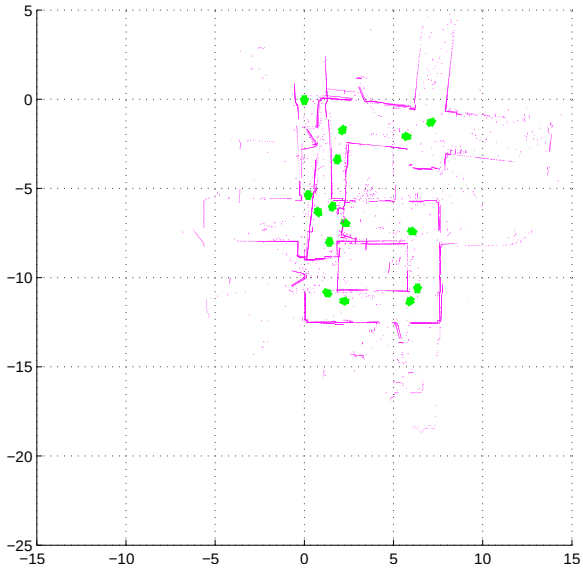


Figure 9.31: Motion estimation using only scan matching with 60% overlap between reference scans

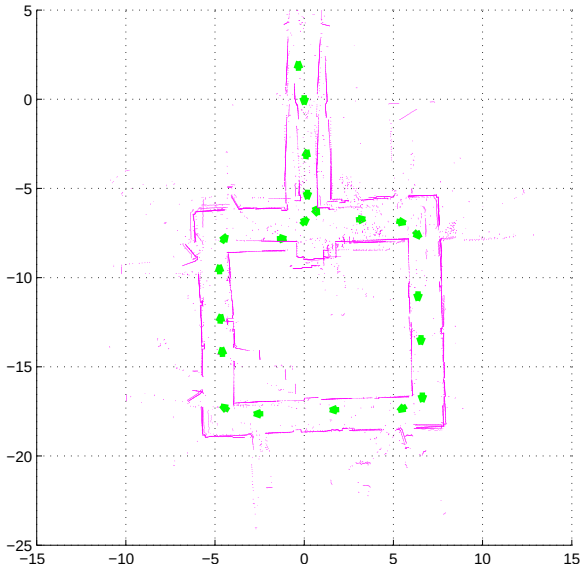


Figure 9.32: Motion estimation using ELS scan matching with 60% overlap between reference scans

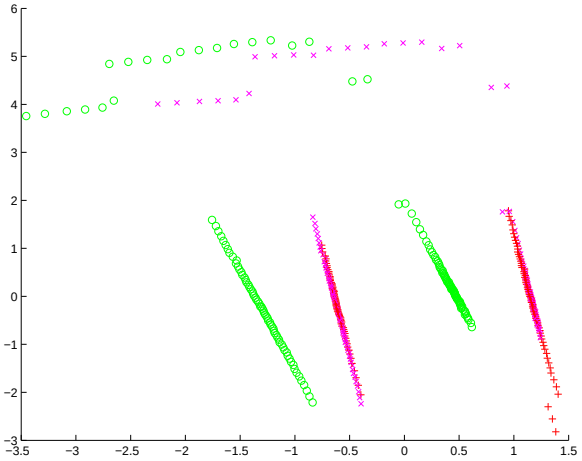


Figure 9.33: Standard scan matching in a loop closing situation

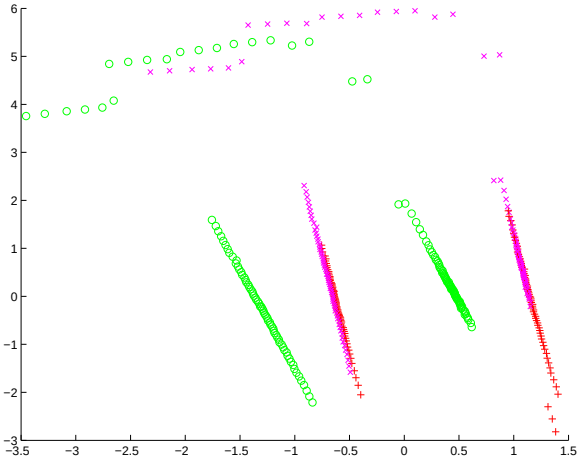


Figure 9.34: ELS scan matching in a loop closing situation

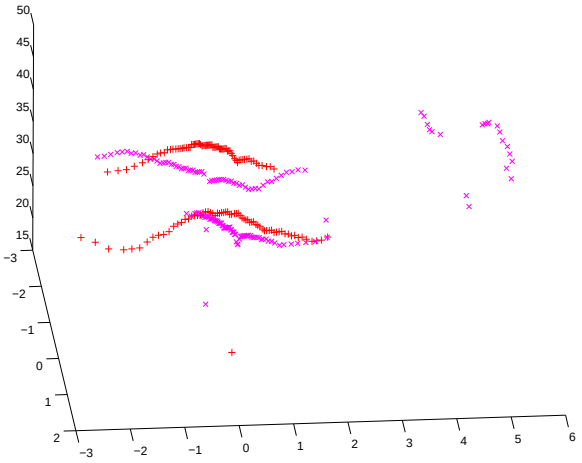


Figure 9.35: Side view of the standard scan matching in a loop closing situation

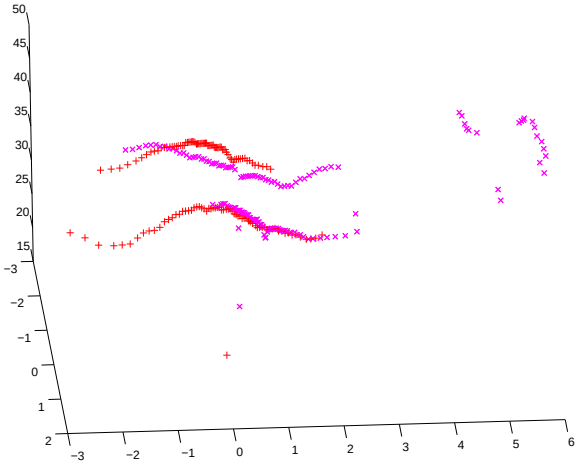


Figure 9.36: Side view of the ELS scan matching in a loop closing situation

results after the loop closure. The landmark uncertainty has been reduced and the map becomes more visually satisfying. If we had only used laser information, it would have been difficult not only to close the loop but also to estimate correctly the displacement along the corridor. In all, about 100 key frames were generated. At this rate, 5 km of corridors could be represented by 10^4 key frames.

The experiment shows that by combining vision and laser, a reliable and computationally efficient approach can be developed. Further experiments would however be needed to confirm the validity of the approach in different types of environments.

9.7 Conclusion

In this chapter, we described a mapping strategy combining vision and laser. We chose to represent the map by laser-vision features. Compared to laser edge points, these landmarks have the advantage of being located in regions with low clutter and a low incidence angle and can be matched reliably with a descriptor combining metrical and image intensity information. No assumptions were made on the type of environment being mapped and this mapping strategy could be used in an outdoor environment.

Inconsistencies are likely to occur in any real-world mapping situation. We reduced the impact by avoiding data associations to occur between "distant" points without explicitly launching a loop closure strategy. This approach combines a topological map with metric estimates to define a topological distance between points. Extensions to this work could consist in delaying the data association or using multiple hypothesis to improve the robustness.

The proposed strategy gave satisfying results on an indoor sequence but more thorough testing with different types of complex environments would be required to evaluate the robustness of the method.

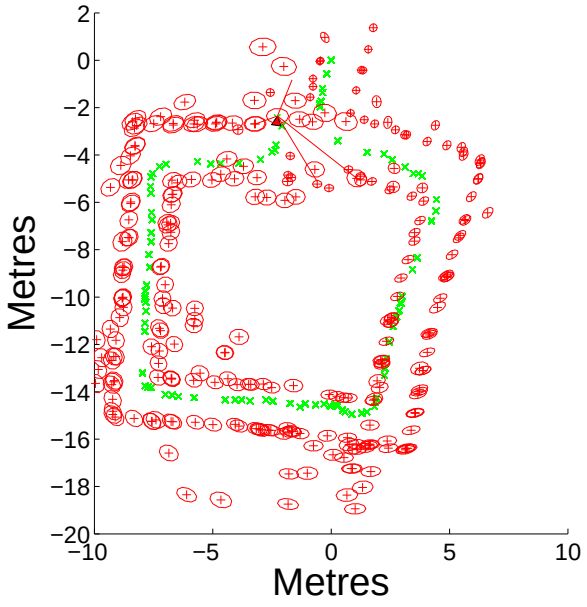


Figure 9.37: Metric map *before* loop closing

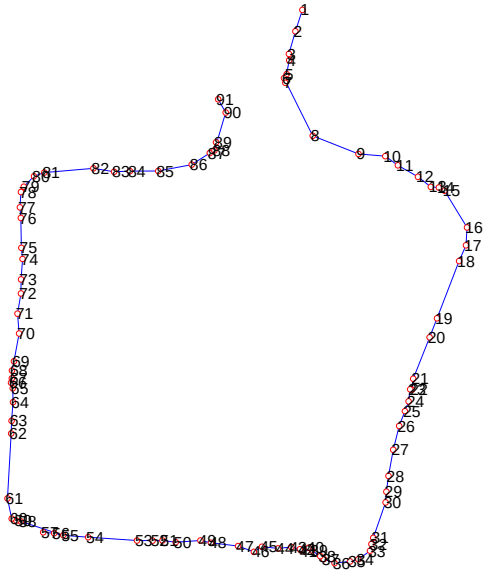


Figure 9.38: Topological map *before* loop closing

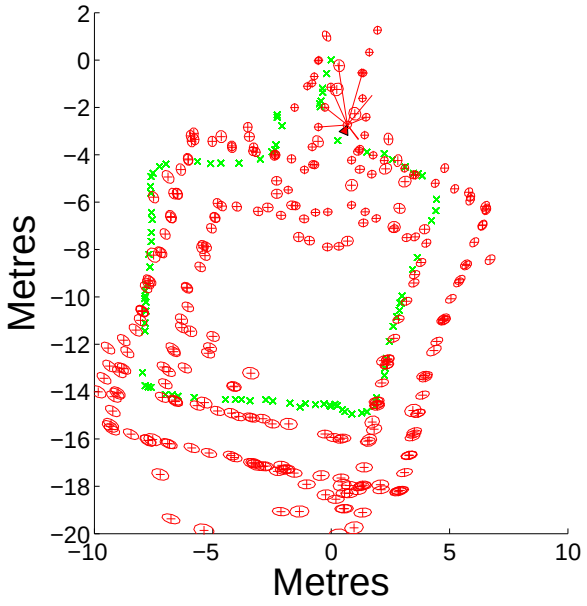


Figure 9.39: Metric map *after* loop closing

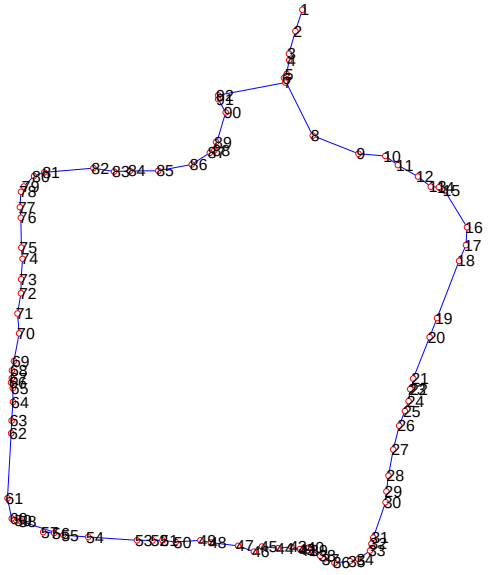


Figure 9.40: Topological map *after* loop closing

Chapter 10

Combining image features with laser readings for 6-DOF structure from motion

Contents

10.1 Introduction

In the previous chapter, we proposed a method to combine visual information and laser range data for planar simultaneous localisation and mapping (SLAM). The visual information improved the data association, localisation and observability that are difficult problems in SLAM. However planar motion estimation imposes strong constraints on the possible environments that can be explored. Estimating the full six degrees of freedom (6-DOF) is an active field of research. Recent approaches use either 3D laser range finders [Newman et al., 2006] and/or vision sometimes combined with interoceptive sensors (wheel encoders, inertial sensors, ...). 3D laser range finders are not currently compatible with real-time applications. Vision sensors alone introduce issues such as propagating correctly the scale factor, initialising the range in the monocular case or associating the data when using multiple cameras. We are not aware of any work combining 2D lasers with vision for 6-DOF motion estimation. This is somewhat surprising as the 2D lasers are common in robotic research, compatible with real-time and have been studied extensively over the past decade. The main motivations for combining the information is avoiding data association problems, removing the difficulty of propagating the scale factor and improving the robustness to outliers. In the first section, we describe the pre-processing step applied to the laser data to remove outliers and extract line segments. We will then present a vision-based tracking approach that includes explicitly the laser information to improve robustness and speed. In the last section, we will analyse briefly how lines can be parameterised to include range-bearing information.

10.2 Pre-processing the laser range measurements

The algorithms proposed in [Victorino, 2002, Chapter 2] were applied to pre-process the laser range measurements. We will summarise briefly the approach, further details can be found in the referenced work.

A filtering step is applied to the data in two passes:

1. rejection of local artifacts. A sliding window of three points is applied to the laser distance measurements. If the relative distance between points is greater than a pre-defined threshold, either we are in presence of an outlier or a new set of continuous points. This algorithm produces n point sets with at least three "close" points. Figure 10.1 shows the results obtained on laser readings made in a corridor. The yellow circles are the rejected points and the dots of different colours correspond to the different point sets generated by the algorithm.

2. local filtering. From the n sets obtained from the previous algorithm, we look for homogeneous sequences of points. The homogeneity criterion is the distance between the distances between adjacent points. Only sequences with a sufficient amount of points (eg. 10 points) are kept. In figure 10.2, the 11 homogeneous regions found by this algorithm are depicted with different colours and numbered in anti-clockwise order. The rejected points are drawn with yellow circles.

We found that these steps could lead to over-segmentation as a single artifact will separate a homogeneous region in two separate sets. In the same way as certain points are considered not to represent physical objects when they are isolated, we can also consider that small distances between segments are mainly due to noisy measurements. For this reason, we apply a merging step where segments that are separated by a distance under a pre-defined threshold are considered as belonging to a continuous surface. Figure 10.3 shows the result after merging the segments (with a threshold of 8 cm). The number of sets is reduced from 11 to 7.

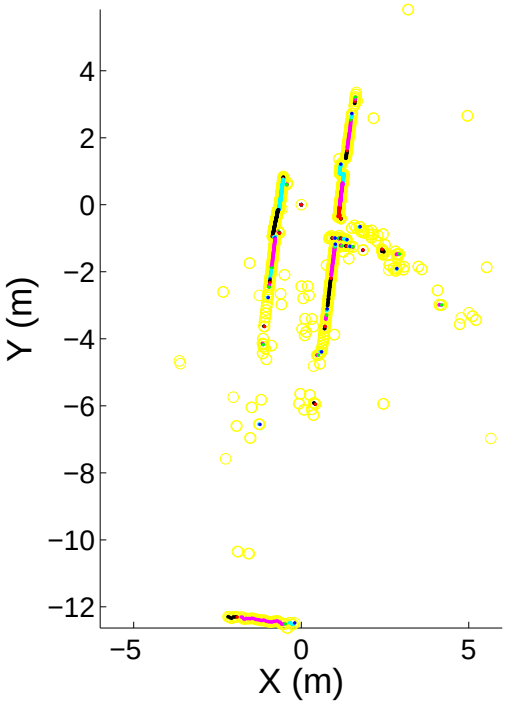


Figure 10.1: Reject local artifacts

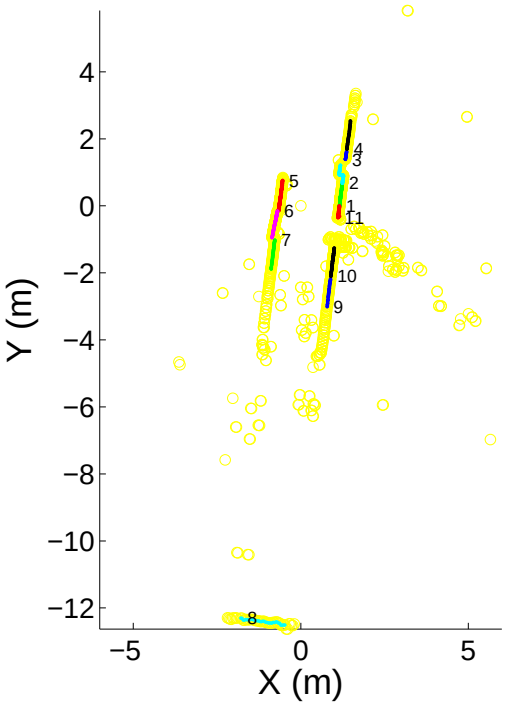


Figure 10.2: Filtered scan laser segmented in homogeneous regions

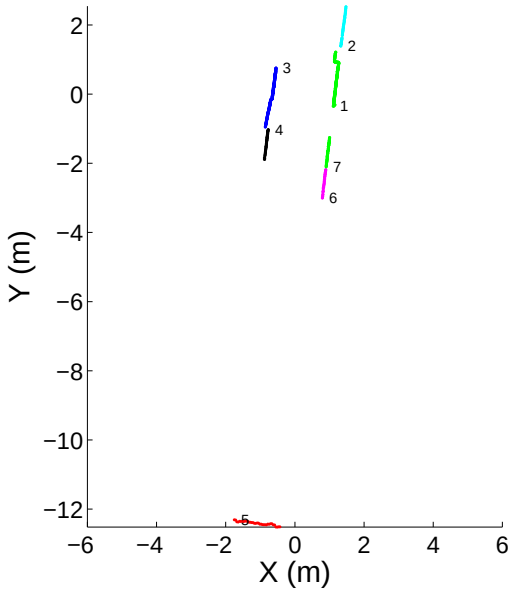


Figure 10.3: Filtered scan after merging close homogeneous point sets

For estimating the motion of planes, we will be interested in finding line segments in the laser scan. We apply the polygonal approximation from [Victorino, 2002] to the homogeneous regions obtained after filtering. Figures 10.4 and 10.5 show the polygonal approximation obtained respectively without and with the prior merging process. In the first case, 12 lines were found and in the second case, 10 lines were extracted.

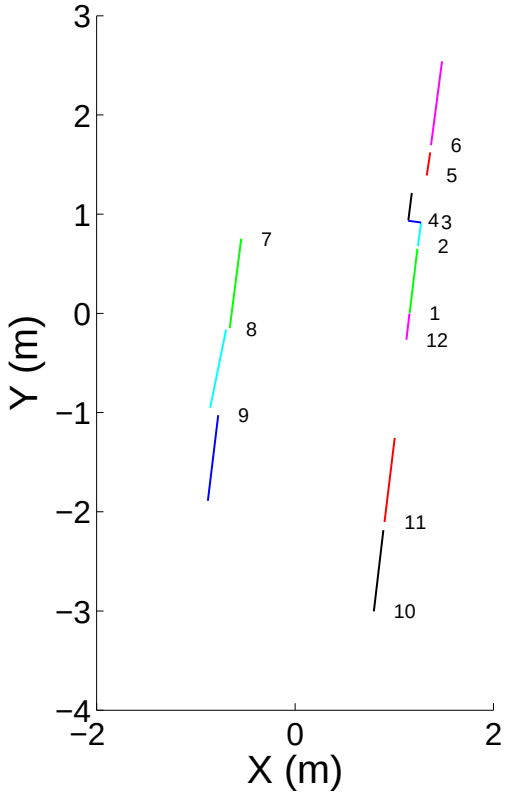


Figure 10.4: Polygonal approximation without prior merging

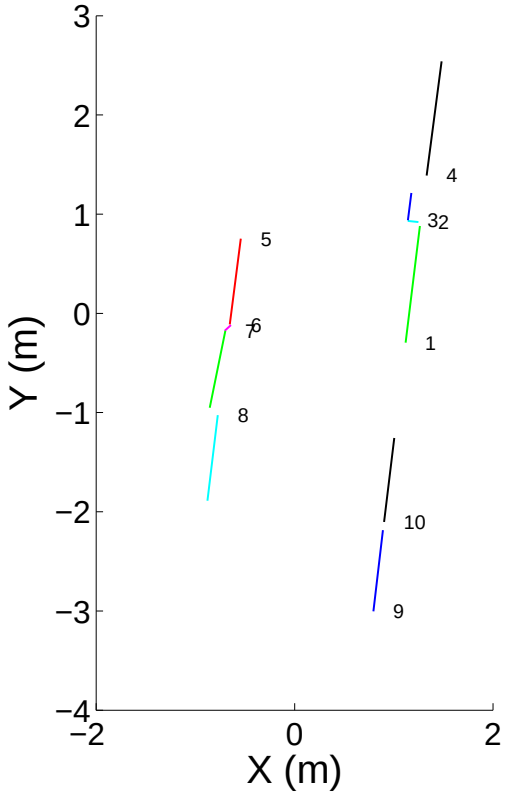


Figure 10.5: Polygonal approximation with prior merging

The merging step generally leads to fewer lines but with more supporting points. It is desirable to reduce the number of features in a SLAM framework to diminish the size of the state vector.

10.3 Combining vision and laser for motion estimation from planes

10.3.1 Plane parameterisation with laser information

In Chapter 6, we studied the registration of the images of planar surfaces. The motion of these image regions can be explained by planar homographies:

$$\mathbf{H} \sim \mathbf{R} + \mathbf{t}\mathbf{n}_d^{*\top} \tag{10.1}$$

where $\mathbf{R} \in \mathbb{SO}(3)$ is the rotation of the camera and $\mathbf{t} \in \mathbb{R}^3$ its translation, $\mathbf{n}_d^* = \mathbf{n}^*/d^*$ is the ratio between the normal vector to the plane $\mathbf{n}^*$ (a unit vector) and the distance d^* of the plane to the origin of the reference frame.

Tracking multiple planes from the image information can be done by minimising the following image-based function:

$$\begin{cases} F(\mathbf{x}) = \frac{1}{2}\sum_{j=1}^{m}\sum_{i=1}^{q_j}\|\mathbf{f}_{ij}\|^2 \\ \mathbf{f}_{ij} = \mathcal{I}\left(\Pi_S\left(\mathbf{w}<\mathbf{H}(\mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}, \widehat{\mathbf{n}}_d^j + \mathbf{n}_d^j(\mathbf{x}))><\boldsymbol{\mathcal{X}}_s^{ij*}>\right)\right) - \mathcal{I}^*(\mathbf{p}_{ij}^*) \end{cases}$$

with $\mathbf{T}$ the transformation matrix, $\widehat{\ }$ corresponds to the estimates and the functions of $\mathbf{x}$ are the increments.

Most work combining vision and laser are laser-based, in other words the visual information is only used to improve the laser estimation or provide a 3D reconstruction. We propose a vision-based approach where the range information from the laser helps constrain the tracking and simultaneous 3D reconstruction and 6-DOF motion estimation.

There are several ways to combine the vision and depth information. Let $\mathbf{s}_1^j, \mathbf{s}_2^j, \ldots, \mathbf{s}_r^j$ correspond to the laser points in the camera reference frame that intersect plane j parameterised by $\mathbf{n}_d^j$. We could for example add the following error to the minimisation (algebraic error) that expresses that the laser points belong to the j^{th} plane:

$$\frac{1}{2}\sum_{k=1}^{r}\|(\mathbf{n}_d^j)^\top \mathbf{s}_k^j - 1\|^2 \tag{10.2}$$

An alternative is to include directly the equation of the line segment extracted from the laser scan into the plane parameterisation. We may note that this does not give the Maximum Likelihood based on the laser and camera uncertainty. However laser measurements are generally more precise than the range observations deduced from vision. This approach has several advantages:

- robustness to noise. By reducing the number of estimated parameters, the cost function is less likely to fall in local minima and robust approaches (such as M-estimators) will lead to better results,

- computational speed. We estimate a single parameter instead of three, so the pseudo-inversion of the Jacobian requires less computation.

Let $\mathbf{P}_1$ and $\mathbf{P}_2$ be two distinct end-points of a line segment associated to a plane of parameter $\mathbf{n}_d$. From the properties of the laser, $\mathbf{P}_1$ and $\mathbf{P}_2$ are not aligned with the origin. Thus the set of possible values for $\mathbf{n}_d$ defines a pencil of planes $\boldsymbol{\mathcal{D}}$ of dimension 1. We have the following properties:

$$\mathbf{P}_{2\times 3}\mathbf{n}_d = \mathbf{1}_{2\times 1} \ , \text{ with } \ \mathbf{P} = [\mathbf{P}_1\ \mathbf{P}_2]^\top \text{ and } \ \mathbf{1} = [1\ 1]^\top$$

Let $\mathbf{n}_b$ be in the kernel of $\mathbf{P} - \mathbf{1}$ and such that $\mathbf{P}\mathbf{n}_b = \mathbf{1}$. Let $\mathbf{n}_{Ker}$ be an element of the kernel of $\mathbf{P}$. These values can be obtained by singular value decomposition or more simply by cross product in the 3D case. Algorithm 4 describes the approach (we add the extra constraint that $\mathbf{n}_b$ and $\mathbf{n}_{Ker}$ are orthogonal).

$$\boldsymbol{\mathcal{D}} = \{\mathbf{n}_b + \lambda\mathbf{n}_{Ker} | \lambda \in \mathbb{R}\}$$

Algorithm 4: Parameterising the pencil of planes passing through two points

Data: Two end-points $\mathbf{P}_1$ and $\mathbf{P}_2$ of a line extracted from a laser scan.

Result: $\mathbf{n}_b$ and $\mathbf{n}_{Ker}$ such that $\boldsymbol{\mathcal{D}} = \{\mathbf{n}_b + \lambda \mathbf{n}_{Ker} | \lambda \in \mathbb{R}\}$ is the pencil of planes containing $\mathbf{P}_1$ and $\mathbf{P}_2$

$\mathbf{n}_{Ker} = \mathbf{P}_1 \times \mathbf{P}_2$
$\mathbf{n}_b = (\mathbf{P}_1 - \mathbf{1}) \times (\mathbf{P}_2 - \mathbf{1})$
$\mathbf{n}_b = \mathbf{n}_b / (\mathbf{1}^\top \mathbf{n}_b)$
$\mathbf{n}_b = \mathbf{n}_b - \frac{\mathbf{n}_b^\top \mathbf{n}_{Ker}}{\|\mathbf{n}_{Ker}\|^2} \mathbf{n}_{Ker}$ `// Impose` $\mathbf{n}_b^\top \mathbf{n}_{Ker} = 0$ `(optional)`

Figure 10.6 illustrates the parameterisation. The line extracted from the laser scan is depicted in a red dotted line between $\mathbf{P}_1$ and $\mathbf{P}_2$. Two planes are drawn for $\lambda = 0.1$ and $\lambda = 0.5$. The points $\mathbf{P}_\perp$ correspond to the closest points to the origin belonging to the planes ($\mathbf{P}_\perp = \frac{\mathbf{n}_d}{\|\mathbf{n}_d\|^2}$). The homography, parameterised by the transformation and the parameter λ can then be written:

$$\mathbf{H} \sim \mathbf{R} + \mathbf{t} \left(\mathbf{n}_b + \lambda \mathbf{n}_{Ker}\right)^\top \tag{10.3}$$

The degenerate case where the plane passes through the origin, $\mathbf{n} = \mathbf{n}_{Ker}$ is rejected to infinity.

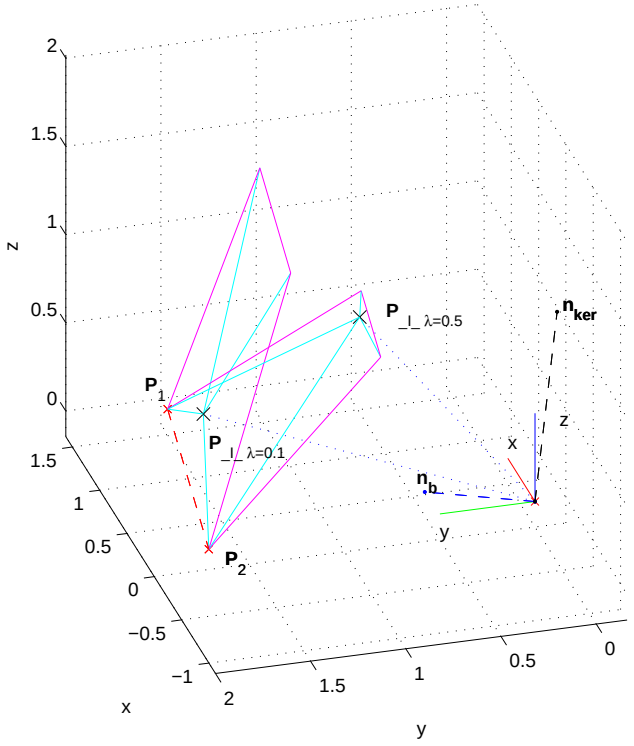


Figure 10.6: Two planes generated by the parameterisation

The least-square system corresponding to the registration of the images becomes:

$$\begin{cases} F(\mathbf{x}, \Lambda) = \frac{1}{2} \sum_{j=1}^{m} \sum_{i=1}^{q_j} \|\mathbf{f}_{ij}\|^2 \\ \mathbf{f}_{ij} = \mathcal{I}\left(\Pi_S\left(\mathbf{w}<\mathbf{H}(\mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}, \widehat{\lambda}_j + \lambda_j)><\boldsymbol{\mathcal{X}}_s^{ij*}>\right)\right) - \mathcal{I}^*(\mathbf{p}_{ij}^*) \end{cases}$$

with $\mathbf{x}$ the parameterisation of the transformation and $\Lambda = (\lambda_1, \ldots, \lambda_m)$ the values defining the m planes. The number of unknowns is $6 + m$ instead of $6 + 3 \times m - 1$ for the full estimation of the plane equations.

The Jacobian needed for the minimisation is similar to the case analysed in Appendix C (we use the same notations).

Let $\widehat{\mathbf{n}} = \mathbf{n}_b + \widehat{\lambda}\mathbf{n}_{Ker}$:

$$\mathbf{J}_{H_\lambda} = \left[\nabla_\lambda \ \mathbf{H}(\widehat{\mathbf{T}}, \widehat{\lambda})^{-1}\mathbf{H}(\widehat{\mathbf{T}}, \lambda)\right]_{\lambda=\widehat{\lambda}} \tag{10.4}$$

$$\mathbf{J}_\lambda(0) = \left[\nabla_\lambda \ \lambda + \widehat{\lambda}\right]_{\lambda=0} = 1 \tag{10.5}$$

we can then show that:

$$\mathbf{J}_{H_\lambda\lambda} = \mathbf{J}_{H_\lambda}\mathbf{J}_\lambda(0) = \begin{bmatrix} (\widehat{\tau}_x\widehat{\mathbf{n}} + \mathbf{b}_x)^\top\widehat{\mathbf{R}}^\top\widehat{\mathbf{t}}\mathbf{n}_{Ker} \\ (\widehat{\tau}_y\widehat{\mathbf{n}} + \mathbf{b}_y)^\top\widehat{\mathbf{R}}^\top\widehat{\mathbf{t}}\mathbf{n}_{Ker} \\ (\widehat{\tau}_z\widehat{\mathbf{n}} + \mathbf{b}_z)^\top\widehat{\mathbf{R}}^\top\widehat{\mathbf{t}}\mathbf{n}_{Ker} \end{bmatrix} \tag{10.6}$$

We should note that this approach only uses the laser information when initialising the plane parameterisation. The information from the laser acquired at each time step was not used in the minimisation, it could however be included using equation (10.2). Another use of the laser data could be occlusion detection.

10.3.2 Initialisation

Finding planes in an unknown environment can be a difficult task using only visual information. Laser data can simplify this step. If we extract line segments in the laser scan as explained in Section 10.2, we know that there exists a region around the line that is planar. The size of this region is unknown. We choose to rely on robust methods (M-estimators) to reject image regions that do not correspond to planar surfaces. We thus make the assumption that the regions chosen are "mainly" planar.

Figure 10.7 shows line segments extracted from a laser scan. Figure 10.8 illustrates image templates assumed to be mainly planar and of an arbitrary size. The reprojection of the laser segments are represented with the reprojection of the bounding boxes.

The minimisation of the reprojection error requires to initialise the plane normal $\mathbf{n}_d$ and more specifically λ with $\mathbf{n}_d = \mathbf{n}_b + \lambda\mathbf{n}_{Ker}$. The fact that we can observe the planes indicates that the plane is more likely to be oriented towards the camera center. This choice corresponds to $\lambda = 0$ (as we chose $\mathbf{n}_b$ orthogonal to $\mathbf{n}_{Ker}$), the corresponding 3D reconstruction is shown in Figure 10.9. In the case of a mobile robot with a camera approximately vertical, we could also initialise the values assuming verticality of the planes in the camera frame as in figure 10.10. This corresponds to $\lambda = -\mathbf{n}_b(3)/\mathbf{n}_{Ker}(3)$. In the experiments, we did not make this assumption to show the validity of the approach even when the initial orientation of the planes are unknown and imprecise.

In the initialisation step, we also use a criteria of entropy to reject planes that do not have sufficient information to enable robust tracking. We also reject planes that correspond to small regions in the environment (we chose to reject lines segments of less than 40 cm) and planes that reproject to small regions in the image (less than 40 pixels across). Figure 10.11 shows the final planes selected for the tracking.

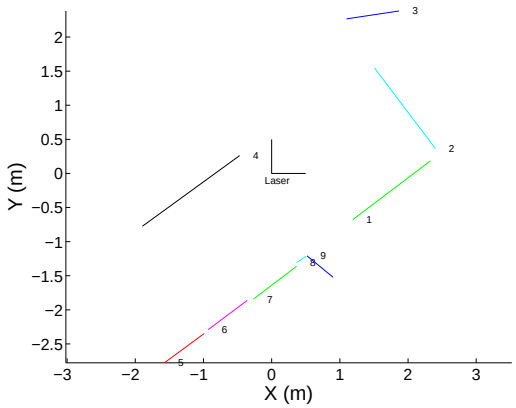


Figure 10.7: Laser segments extracted from a laser scan

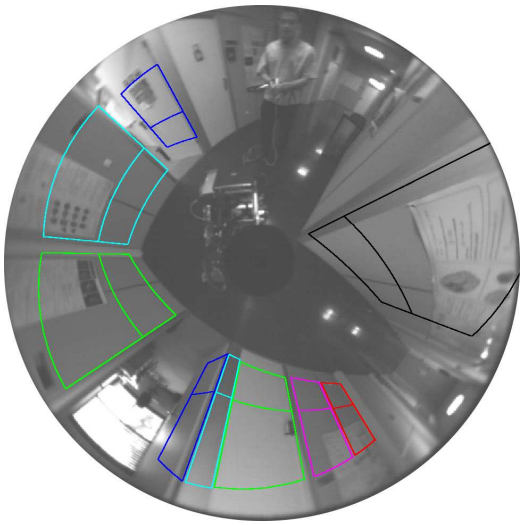


Figure 10.8: Planes chosen from laser scan segments

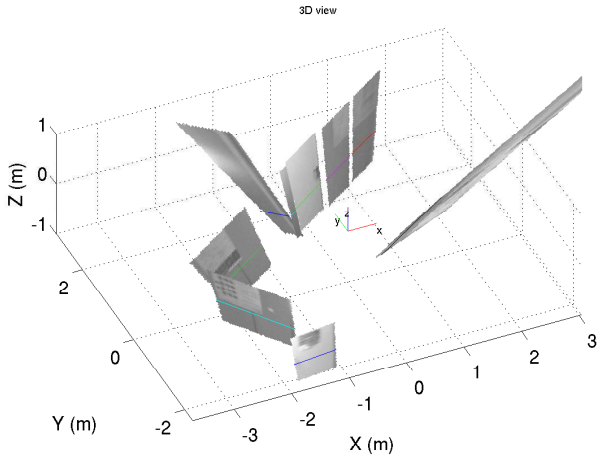


Figure 10.9: Plane equation initialisation to maximise the viewing angle

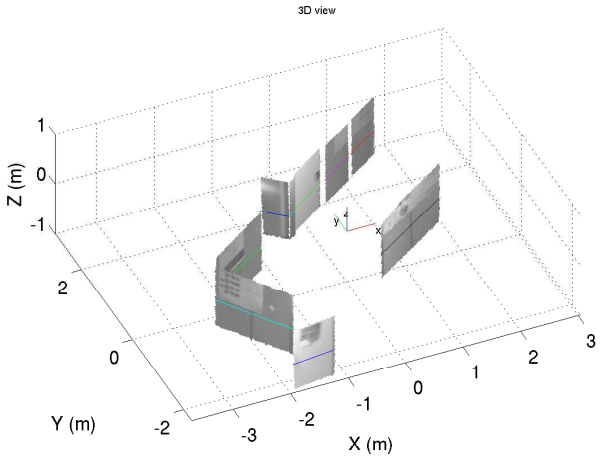


Figure 10.10: Plane equation initialisation assuming verticality of the planes

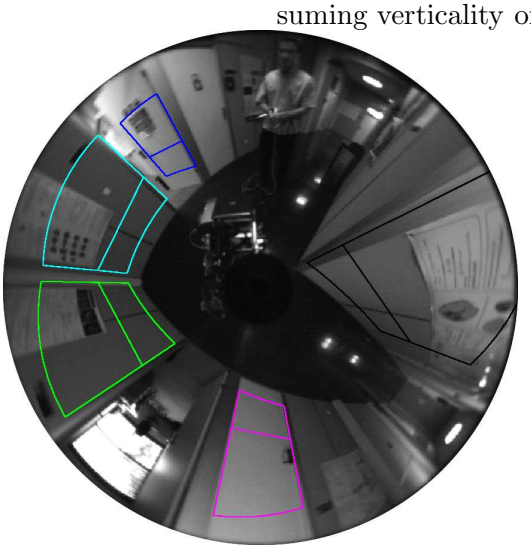


Figure 10.11: Planes selected for the tracking

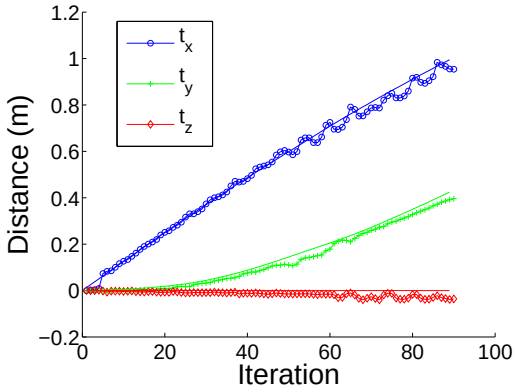


Figure 10.12: Translation estimation

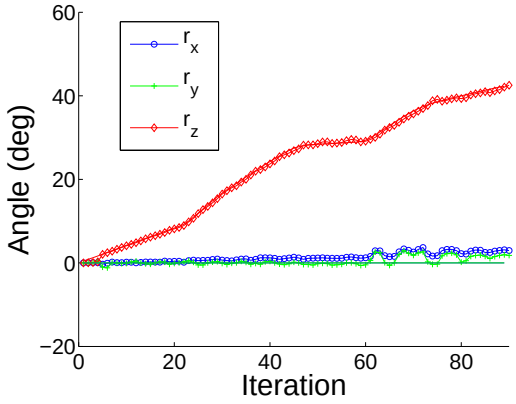


Figure 10.13: Rotation estimation

10.3.3 Experimental validation

The 6-DOF motion estimation and reconstruction was applied to a sequence of 90 omnidirectional images and laser scans. This sequence was used previously for the validation of the omnidirectional tracking in Chapter 6.

The planes were extracted automatically as explained in the previous section. Figures 10.14 and 10.15 represent the planes chosen for the tracking and illustrate the initial values given to the normals.

Figures 10.16, 10.18 and 10.20 show the planes tracked during the sequence and figures 10.17, 10.19 and 10.21 the associated 3D reconstructions. Figures 10.12 and 10.13 show the translation and rotation estimates respectively. The plane contours are the reprojection of the reference contours in the current image using the motion estimation. This explains why the depicted motion for the plane on the right in the images appears highly distorted.

This sequence is interesting because strong occlusion and specularities affect the observed planes. The M-estimators are sufficient to reject the outliers and the 6-DOF motion estimation stays precise. Further improvements could be obtained through filtering and by the use of the laser readings that can guide the rejection of occluded sections of the image. An analysis should also be done to know the domain of validity of the tracking when part of the homography is fixed by the laser data.

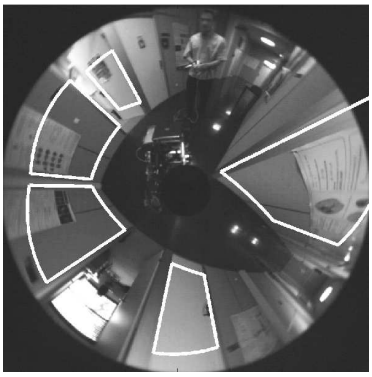


Figure 10.14: Initial tracked planes

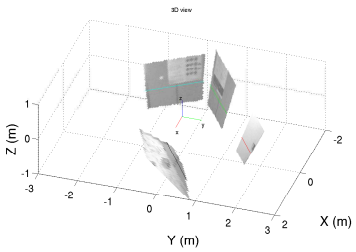


Figure 10.15: Initial 3D reconstruction

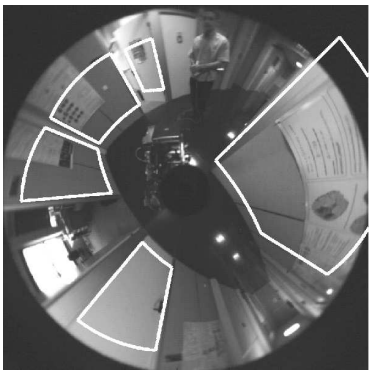


Figure 10.16: Tracked planes after image 30

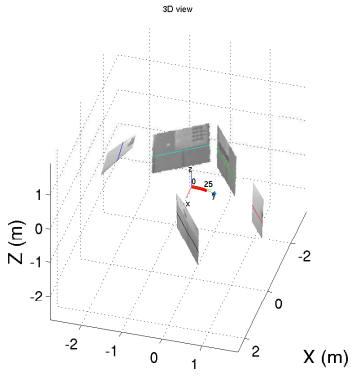


Figure 10.17: 3D reconstruction and motion after image 30

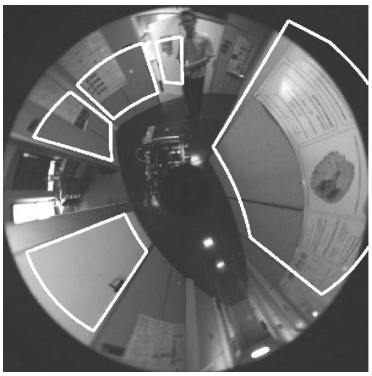


Figure 10.18: Tracked planes after image 60

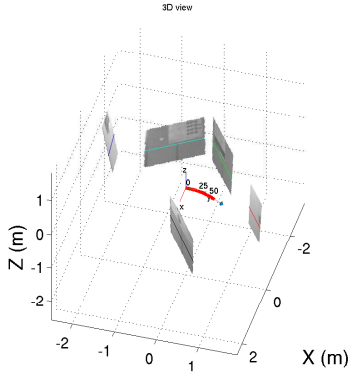


Figure 10.19: 3D reconstruction and motion after image 60

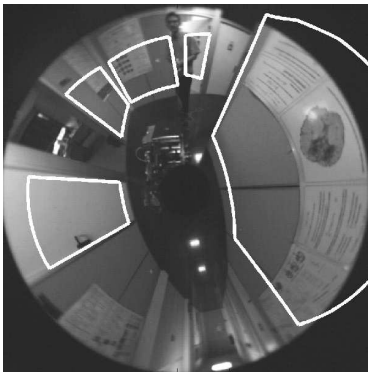


Figure 10.20: Tracked planes after image 90

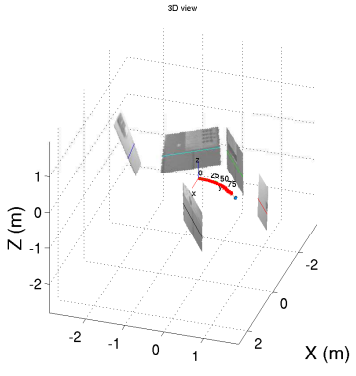


Figure 10.21: 3D reconstruction and motion after image 90

10.4 Conclusion and perspectives

In this chapter, we proposed an approach to combine 3D planes and 2D lines extracted from 2D laser data. We also discussed how laser data could also be included in structure from motion from lines. We showed that by considering the problem from a vision perspective, we could estimate the full 6-DOF motion of the robot. Experimental results on this preliminary work are encouraging and give an insight into the possibilities offered by this combination of sensors. Further developments would be needed to estimate the covariance associated to the 3D reconstruction and build a stochastic map.

Chapter 11

Conclusion and future research

Research in the field of simultaneous localisation and mapping has recently provided important improvements from the point of view of filtering to build maps of large-scale environments. However the complexity of the mapped environments has been limited by the sensors used. Most large scale mapping implementations were made using 2D lasers or millimetre wave radars that do not enable reliable data association or 6-DOF motion estimation.

This thesis attempted to address some of the issues in map building and motion estimation by a novel combination of sensors: an omnidirectional camera and a 2D laser range finder. The visual information helps recognise previously explored regions. The laser gives precise range-bearing measurements. Combined, they can provide efficient and robust 3-DOF or 6-DOF motion estimation.

This chapter summarises the contributions of this work and future directions of research.

11.1 Summary

This dissertation covered some of the essential components of a working system combining omnidirectional vision and 2D lasers:

- the omnidirectional calibration process: how to find a precise relationship between the image points and the projective rays,
- how to find the position between the laser sensor and the camera,
- omnidirectional motion estimation from planes: how to track planes efficiently and take into account the image distortion,
- omnidirectional motion estimation from lines,
- laser-vision coupling: how to combine omnidirectional vision and laser range-bearing for 3-DOF SLAM and 6-DOF motion estimation.

We started by analysing the projection model associated to omnidirectional sensors. We showed that the standard planar projection model was not adapted to take into account the wide field of view of the sensor and that the spherical perspective projection model solved the issues regarding cheirality and discontinuities in the image processing.

The unified projection model by Geyer and Barreto was then adapted to enable the calibration of sensors with distortion errors in the projection model such as parabolic mirrors with telecentric lenses.

We also devised a methodology to enable the calibration of a wide variety of sensors used in robotics and computer vision. This lead to a calibration toolbox made available over the Internet.

Combining vision and laser also requires finding the relative position between the sensors. The case of a laser range finder with a visible laser beam and the case when the beam is invisible were studied. This step makes it possible to relate the intensity or color information in the image to the laser measurements. The laser scan combined with vision, that we named Enriched Laser Scan (ELS), reduces the problem of observability common to laser scan matching in corridor-like environments. The advantages of using this combination of sensors were shown on real data. The wide field of view of the omnidirectional sensor made it possible to localise the robot independently of the motion estimates and enabled loop closing.

A chapter was dedicated to iterative minimisation over groups using Lie algebras. Without being original as such, the approach is not well known and many problems in computer vision and robotics could benefit from this method. In particular, the estimation of the uncertainty needed in SLAM frameworks requires minimal parameterisations. The case of 3D lines is a typically example where a minimal representation should be used for SLAM but apparently this has never been the case.

This thesis also analysed motion estimation using only omnidirectional vision. We showed how visual tracking could be adapted to single viewpoint sensors by using parametric models (in this case homographies) combined with the spherical perspective model. We also contributed by improving the computational cost of the efficient second order minimisation algorithm. Comparison with the inverse compositional shows a systematic gain in time thanks to the better convergence rate. We then derived a constrained tracking method to estimate the camera and plane positions simultaneously. How to remove outliers and specularities was also discussed.

Planes or points do not always provide a sufficient amount of information for structure and motion, so we studied line features. We analysed how they could be parameterised in a way compatible with extraction, tracking and bundle adjustment for omnidirectional sensors.

Finally this thesis proposed two approaches for laser-vision structure and motion. The first relies on the extraction of novel salient laser-vision features that can be included in a standard 3-DOF EKF framework. Combined with a localisation approach using omnidirectional vision, loop closing was made possible and the overall approach provided a complete SLAM method that does not require the standard assumption of a piecewise linear environment. This technique could ultimately be applied to outdoor environments. The second approach provides 6-DOF motion estimation through the combination of laser lines and 3D planes. Results were shown on a 6-DOF motion estimation sequence. Further research would be needed to include this work in a full SLAM framework.

11.2 Future research

11.2.1 Extensions to the current work

In this thesis, we were not able to explore in detail the problem of identifying previously observed scenes. However we used an approach from image indexing that has seemingly rarely been used for robot localisation. We believe that combining auto-correlograms with Haar invariants over local attributes could be a way of improving localisation. Combining local metric information could also help distinguish between visually similar regions.

The loop closing procedure based on a qualitative estimation followed by scan matching proved sufficient in our indoor environment but in a more general and complex situation, deciding in a single step could lead to incorrect loop closing. Delaying the decision as is done regularly in a SLAM framework would be required for a real-world working system.

In Chapter 6, we discussed efficient ways of tracking planar features. SSD is however limited to small inter-frame motion. Ways of initialising the tracking, extending the convergence domain and recovering from total occlusion are important steps for applying the approach in general real-world situations. Some possible solutions would consist in finding invariant descriptors that could enable an efficient search in the image such as histograms or image invariants. Correlation is a standard way of initialising image tracking but is sensitive to changes in shape and is computationally expensive. How to adapt the initialisation to the variable image resolution in omnidirectional vision is also an open question. Kalman filtering or particle filtering with a motion model can also improve the initialisation.

Auto-calibration was only briefly discussed in this thesis. However it is well known that the projective properties of panoramic sensors are strongly related to the intrinsic parameters. Visual tracking for example could provide a very convenient way of calibrating the sensor.

11.2.2 Longer term developments

Plane tracking has many advantages in a SLAM framework in particular the strong constraints it imposes on the motion with few parameters. However extracting planes in a general situation can be a complex task and was not discussed in this work. The notion of plane itself depends on the distance to the landmark and few articles have studied the problem of on-line probabilistic planar structure estimation.

For vision sensors, observability poses difficult challenges. Identifying that the features do not constrain the motion can be done theoretically by looking at the covariance matrix but in practice the noise makes the identification harder and can render the map inconsistent. If we manage to identify this situation, a new map could for example be initialised and we could later try and fuse the maps.

Cameras provide a rich description of the environment and often only part of this information is needed to estimate the motion. Active vision, that chooses where to look for worthwhile data based on maximising the information gain is an interesting topic for longer term developments.

Another topic that was not covered in this thesis was the combination of vision and laser for dynamic environments. Previous work has studied the use of 2D lasers to track moving objects or people but vision could undoubtedly improve the robustness of these methods and help re-identify previously observed dynamic features. Building semantic representations by identifying objects such as chairs or doors can also help improve the quality of the maps. Classification and object recognition represents an important part of the vision literature and could benefit robot mapping.

Appendix A

Jacobian of the projection function

A.1 Changing frame

$$W(\boldsymbol{\mathcal{X}}, V^1) = \mathbf{R}_{\boldsymbol{\mathcal{X}}}(\mathbf{Q}') + \mathbf{t}$$

$$\frac{\partial W}{\partial \mathbf{Q}}_{3\times4} = \frac{\partial W}{\partial \mathbf{Q}'}_{3\times4} \frac{\partial \mathbf{Q}'}{\partial \mathbf{Q}}_{4\times4}$$

$$\frac{\partial W}{\partial \mathbf{Q}'}_{3\times4} \left[\begin{array}{cccc} \left(\frac{\partial W}{\partial q'_0}\right) & \left(\frac{\partial W}{\partial q'_1}\right) & \left(\frac{\partial W}{\partial q'_2}\right) & \left(\frac{\partial W}{\partial q'_3}\right) \end{array} \right]$$

$$\frac{\partial W}{\partial q'_0} = 2 \left[\begin{array}{c} q'_0 x - q'_3 y + q'_2 z \\ q'_3 x + q'_0 y - q'_1 z \\ -q'_2 x + q'_1 y + q'_0 z \end{array} \right] , \quad \frac{\partial W}{\partial q'_1} = 2 \left[\begin{array}{c} q'_1 x + q'_2 y + q'_3 z \\ q'_2 x - q'_1 y - q'_0 z \\ q'_3 x + q'_0 y - q'_1 z \end{array} \right]$$

$$\frac{\partial W}{\partial q'_2} = 2 \left[\begin{array}{c} -q'_2 x + q'_1 y + q'_0 z \\ q'_1 x + q'_2 y + q'_3 z \\ -q'_0 x + q'_3 y - q'_2 z \end{array} \right] , \quad \frac{\partial W}{\partial q'_3} = 2 \left[\begin{array}{c} -q'_3 x - q'_0 y + q'_1 z \\ q'_0 x - q'_3 y + q'_2 z \\ q'_1 x + q'_2 y + q'_3 z \end{array} \right]$$

$$\frac{\partial \mathbf{Q}'}{\partial \mathbf{Q}}_{4\times4} = \frac{1}{\mathbf{Q}'^3} \left[\begin{array}{cccc} q_1^2 + q_2^2 + q_3^2 & -q_0 q_1 & -q_0 q_2 & -q_0 q_3 \\ -q_0 q_1 & q_0^2 + q_2^2 + q_3^2 & -q_1 q_2 & -q_1 q_3 \\ -q_0 q_2 & -q_1 q_2 & q_0^2 + q_1^2 + q_3^2 & -q_2 q_3 \\ -q_0 q_3 & -q_1 q_3 & -q_2 q_3 & q_0^2 + q_1^2 + q_2^2 \end{array} \right]$$

$$\frac{\partial W}{\partial \mathbf{t}}_{3\times3} = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right]$$

Finally:

$$\frac{\partial W}{\partial V^1}_{3\times7} = \left[\begin{array}{cc} \left(\frac{\partial W}{\partial \mathbf{Q}'} \frac{\partial \mathbf{Q}'}{\partial \mathbf{Q}}\right)_{3\times4} & \left(\frac{\partial W}{\partial \mathbf{t}}\right)_{3\times3} \end{array} \right]$$

A.2 Mirror transformation

$H = \hbar \circ S$ with $r = \sqrt{x^2 + y^2 + z^2}$:

$$\frac{\partial H}{\partial \boldsymbol{\mathcal{X}}}_{2\times 3} = \frac{1}{r(z+\xi r)^2} \begin{bmatrix} rz + \xi(y^2+z^2) & -\xi xy & x(-\xi z - r) \\ -\xi xy & rz + \xi(x^2+z^2) & y(-\xi z - r) \end{bmatrix}$$

$$\frac{\partial H}{\partial V^2}_{2\times 1} = -\frac{r}{(z+\xi r)^2} \begin{bmatrix} x \\ y \end{bmatrix}$$

A.3 Distortion

With $\rho_2 = x^2 + y^2$:

$$D(\boldsymbol{\mathcal{X}}, V^3) = \begin{bmatrix} x(1 + k_1\rho_2 + k_2\rho_2^2) + 2p_1xy + p_2(\rho_2 + 2x^2) \\ y(1 + k_1\rho_2 + k_2\rho_2^2) + p_1(\rho_2 + 2y^2) + 2p_2xy \end{bmatrix}$$

$$\frac{\partial D}{\partial V^3}_{2\times 4} = \begin{bmatrix} x\rho_2 & x\rho_2^2 & 2xy & \rho_2 + 2x^2 \\ y\rho_2 & y\rho_2^2 & \rho_2 + 2y^2 & 2xy \end{bmatrix}$$

$$\frac{\partial D}{\partial \boldsymbol{\mathcal{X}}}_{2\times 2} = \begin{bmatrix} 1 + k_1(\rho_2 + 2x^2) + k_2\rho_2(\rho_2 + 4x^2) + p_1 2y + p_2 6x & 1 + k_1 2xy + k_2 4\rho_2 xy + p_1 2x + p_2 2y \\ 1 + k_1 2xy + k_2 4\rho_2 xy + p_1 2x + p_2 2y & 1 + k_1(\rho_2 + 2y^2) + k_2\rho_2(\rho_2 + 4y^4) + p_1 6y + p_2 2x \end{bmatrix}$$

A.4 Generalised projection matrix

$$k(\boldsymbol{\mathcal{X}}, V^4) = \begin{bmatrix} \gamma_1(x + \alpha y) + c_1 \\ \gamma_2 y + c_2 \end{bmatrix}$$

$$\frac{\partial k}{\partial V^4}_{2\times 5} = \begin{bmatrix} \gamma_1 y & x + \alpha y & 0 & 1 & 0 \\ 0 & 0 & y & 0 & 1 \end{bmatrix}, \quad \frac{\partial k}{\partial \boldsymbol{\mathcal{X}}}_{2\times 2} = \begin{bmatrix} \gamma_1 & \gamma_1\alpha \\ 0 & \gamma_2 \end{bmatrix}$$

Appendix B

Jacobian for tracking a single plane

Current Jacobian

We will write the current Jacobian as the product of five different Jacobians:

$$\begin{aligned}\mathbf{J}(\mathbf{0}) &= \left[\nabla_{\mathbf{x}}\, \mathcal{I}\left(\Pi(\mathbf{w}<\widehat{\mathbf{H}}\mathbf{H}(\mathbf{x})><\boldsymbol{\mathcal{X}}_s^*>)\right) - \mathcal{I}^*(\mathbf{p}^*)\right]_{\mathbf{x}=\mathbf{0}} \\ &= \mathbf{J}_{\mathcal{I}}\, \mathbf{J}_{\Pi}\, \mathbf{J}_w\, \mathbf{J}_{\mathbf{H}_{\mathbf{x}}}(\mathbf{0})\end{aligned}$$

Noting that:

$$\begin{aligned}\mathbf{w}<\widehat{\mathbf{H}}\mathbf{H}(\mathbf{x})><\boldsymbol{\mathcal{X}}_s^*> &= \mathbf{w}<\widehat{\mathbf{H}}><\mathbf{w}<\mathbf{H}(\mathbf{x})><\boldsymbol{\mathcal{X}}_s^*>> \\ &= \mathbf{w}<\widehat{\mathbf{H}}><\Pi^{-1}(\mathbf{q})>\end{aligned}$$

with $\mathbf{q} = \Pi(\mathbf{w}<\mathbf{H}(\mathbf{x})><\boldsymbol{\mathcal{X}}_s^*>)$. The first Jacobian $\mathbf{J}_{\mathcal{I}}$ is:

$$\begin{aligned}\mathbf{J}_{\mathcal{I}} &= \left[\nabla_{\mathbf{q}}\, \mathcal{I}\left(\Pi(\mathbf{w}<\widehat{\mathbf{H}}><\Pi^{-1}(\mathbf{q})>)\right)\right]_{\mathbf{q}=\mathbf{t}} \\ &\quad \text{with}: \mathbf{t} = \Pi(\mathbf{w}<\mathbf{H}(\mathbf{0})><\boldsymbol{\mathcal{X}}_s^*>)\end{aligned}$$

$\Pi(\mathbf{w}<\mathbf{H}(\mathbf{0})><\boldsymbol{\mathcal{X}}_s^*>) = \Pi(\mathbf{w}<\mathbf{I}><\boldsymbol{\mathcal{X}}_s^*>) = \mathbf{p}$ so $\mathbf{J}_{\mathcal{I}}$ is the jacobian of the current image. The Jacobian of Π is detailed in Appendix A.

$$\mathbf{J}_w = [\nabla_{\mathbf{H}}\, \mathbf{w}<.><\boldsymbol{\mathcal{X}}_s^*>]_{\mathbf{H}=\mathbf{H}(\mathbf{0})=\mathbf{I}} = \begin{bmatrix} \boldsymbol{\mathcal{X}}_s^{*\top} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\mathcal{X}}_s^{*\top} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\mathcal{X}}_s^{*\top} \end{bmatrix}_{3\times 9}$$

$$\mathbf{J}_{\mathbf{H}}(\mathbf{0}) = [\nabla_{\mathbf{x}}\, \mathbf{H}]_{\mathbf{x}=\mathbf{0}} = [\text{flat}(\mathbf{A}_1)^\top\ \text{flat}(\mathbf{A}_2)^\top \cdots \text{flat}(\mathbf{A}_8)^\top]_{9\times 8}^\top$$

$$\text{with}: \text{flat}(\mathbf{M}_{n\times m}) = [m_{11}\ m_{12}\ \cdots\ m_{1m}\ m_{21}\ m_{22}\ \cdots m_{nm}]^\top$$

Reference Jacobian

The reference Jacobian $\mathbf{J}(\mathbf{x}_0)$ can be written as:

$$\begin{aligned}\mathbf{J}(\mathbf{x}_0) &= \left[\nabla_{\mathbf{x}}\, \mathcal{I}\left(\Pi(\mathbf{w}<\widehat{\mathbf{H}}\mathbf{H}(\mathbf{x})><\boldsymbol{\mathcal{X}}_s^*>)\right) - \mathcal{I}^*(\mathbf{p}^*)\right]_{\mathbf{x}=\mathbf{x}_0} \\ &= \mathbf{J}_{\mathcal{I}^*}\, \mathbf{J}_{\Pi}\, \mathbf{J}_w\, \mathbf{J}_{(\overline{\mathbf{H}}^{-1}\widehat{\mathbf{H}}\mathbf{H}_{\mathbf{x}})}(\mathbf{x}_0)\end{aligned}$$

Noting that:

$$\begin{aligned}\mathbf{w}<\widehat{\mathbf{H}}\mathbf{H}(\mathbf{x})><\boldsymbol{\mathcal{X}}_s^*> &= \mathbf{w}<\overline{\mathbf{H}}><\mathbf{w}<\overline{\mathbf{H}}^{-1}\widehat{\mathbf{H}}\mathbf{H}(\mathbf{x})><\boldsymbol{\mathcal{X}}_s^*>> \\ &= \mathbf{w}<\overline{\mathbf{H}}><\Pi^{-1}(\mathbf{q})>\end{aligned}$$

with $\mathbf{q} = \Pi(\mathbf{w}<\overline{\mathbf{H}}^{-1}\widehat{\mathbf{H}}\mathbf{H}(\mathbf{x})><\boldsymbol{\mathcal{X}}_s^*>)$, the Jacobian $\mathbf{J}_{\mathcal{I}^*}$ is:

$$\begin{aligned}\mathbf{J}_{\mathcal{I}^*} &= \left[\nabla_{\mathbf{q}}\,\mathcal{I}\left(\Pi(\mathbf{w}<\overline{\mathbf{H}}><\Pi^{-1}(\mathbf{q})>)\right)\right]_{\mathbf{q}=\mathbf{t}} \\ &\quad \text{with : } \mathbf{t} = \Pi(\mathbf{w}<\overline{\mathbf{H}}^{-1}\widehat{\mathbf{H}}\mathbf{H}(\mathbf{x}_0)><\boldsymbol{\mathcal{X}}_s^*>)\end{aligned}$$

$\Pi(\mathbf{w}<\overline{\mathbf{H}}^{-1}\widehat{\mathbf{H}}\mathbf{H}(\mathbf{x}_0)><\boldsymbol{\mathcal{X}}_s^*>) = \Pi(\mathbf{w}<\mathbf{I}><\boldsymbol{\mathcal{X}}_s^*>) = \mathbf{p}$ so $\mathbf{J}_{\mathcal{I}^*}$ is the jacobian of the reference image.

$\mathbf{J}_\Pi$ and $\mathbf{J}_w$ are the same as for the current Jacobian.

Thanks to the Lie group parameterization, $\mathbf{J}_{(\overline{\mathbf{H}}^{-1}\widehat{\mathbf{H}}\mathbf{H}_\mathbf{x})}(\mathbf{x}_0)\mathbf{x}_0 = \mathbf{J}_{\mathbf{H}_\mathbf{x}}(\mathbf{0})\mathbf{x}_0$.

Appendix C

Jacobian for tracking multiple planes

For clarity, we will no longer indicate i, j and d.

The translation can only be obtain up to a scale factor so to avoid over-parameterising the system, the "first plane" can be estimated differently:

$$\mathbf{H}_1 = \mathbf{R} + \frac{\mathbf{t}}{d_1}\mathbf{n}_1^\top \tag{C.1}$$

$$\mathbf{H}_2 = \mathbf{R} + \frac{\mathbf{t}}{d_1}\left(\frac{d_1}{d_2}\mathbf{n}_2^\top\right) \tag{C.2}$$

$$\vdots \quad \vdots$$

$$\mathbf{H}_m = \mathbf{R} + \frac{\mathbf{t}}{d_1}\left(\frac{d_1}{d_m}\mathbf{n}_m^\top\right) \tag{C.3}$$

In other words, we can only estimate two parameters for $\mathbf{n}_1$ (for example by normalising) and three for $\frac{d_1}{d_i}\mathbf{n}_i$, $i > 1$.

Current Jacobian

We will decompose the current Jacobian as such:

$$\begin{aligned}\mathbf{J}(\mathbf{0}) &= \left[\nabla_{\mathbf{x}}\, \mathcal{I}\left(\Pi(\mathbf{w}<\mathbf{H}(\mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}, \widehat{\mathbf{n}} + \mathbf{n}(\mathbf{x}))><\boldsymbol{\mathcal{X}}_s^*>)\right) - \mathcal{I}^*(\mathbf{p}^*)\right]_{\mathbf{x}=\mathbf{0}} \\ &= \mathbf{J}_{\mathcal{I}}\mathbf{J}_{\Pi}\, \mathbf{J}_w[\mathbf{J}_{H_T T}(\mathbf{0}) \;\; \mathbf{J}_{H_n n}(\mathbf{0})] \qquad \text{(C.4)} \\ & \qquad \text{(C.5)}\end{aligned}$$

Noting that:

$$\mathbf{w}<\mathbf{H}(\mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}, \widehat{\mathbf{n}} + \mathbf{n}(\mathbf{x}))><\boldsymbol{\mathcal{X}}_s^*> = \mathbf{w}<\mathbf{H}(\widehat{\mathbf{T}}, \widehat{\mathbf{n}})><\Pi^{-1}(\mathbf{q})>$$

with:

$$\mathbf{q} = \Pi(\mathbf{w}<\mathbf{H}(\widehat{\mathbf{T}}, \widehat{\mathbf{n}})^{-1}\mathbf{H}(\mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}, \widehat{\mathbf{n}} + \mathbf{n}(\mathbf{x}))><\boldsymbol{\mathcal{X}}_s^*>) \tag{C.6}$$

In $\mathbf{0}$ this leads to $\mathbf{q} = \Pi(\mathbf{w}<\mathbf{I}><\boldsymbol{\mathcal{X}}_s^*>)$ so the first Jacobian $\mathbf{J}_{\mathcal{I}}$ is:

$$\mathbf{J}_{\mathcal{I}} = \left[\nabla_{\mathbf{q}}\, \mathcal{I}\left(\Pi(\mathbf{w}<\mathbf{H}(\widehat{\mathbf{T}}, \widehat{\mathbf{n}})><\Pi^{-1}(\mathbf{q})>)\right)\right]_{\mathbf{q}=\mathbf{p}} \tag{C.7}$$

which is the jacobian of the current image.
$\mathbf{J}_\Pi$ and $\mathbf{J}_w$ are the same as for the single plane tracking.

$$\mathbf{J}_{H_T} = \left[\nabla_{\mathbf{T}}\ \mathbf{H}(\widehat{\mathbf{T}}, \widehat{\mathbf{n}})^{-1}\mathbf{H}(\mathbf{T}\widehat{\mathbf{T}}, \widehat{\mathbf{n}})\right]_{\mathbf{T}=\mathbf{I}} \tag{C.8}$$

$$\mathbf{J}_T(\mathbf{0}) = \left[\nabla_{\mathbf{x}}\ \mathbf{T}(\mathbf{x})\right]_{\mathbf{x}=\mathbf{0}} \tag{C.9}$$

If we write $\widehat{\mathbf{T}}$ as:

$$\widehat{\mathbf{T}} = \begin{bmatrix} \widehat{\mathbf{R}} & \widehat{\mathbf{t}} \\ \mathbf{0} & 1 \end{bmatrix} \tag{C.10}$$

and let $\widehat{\boldsymbol{\tau}} = (\widehat{\tau}_x, \widehat{\tau}_y, \widehat{\tau}_z)$ be the (3×1) vector:

$$\widehat{\boldsymbol{\tau}} = \frac{-\widehat{\mathbf{R}}^\top\widehat{\mathbf{t}}}{1 + \widehat{\mathbf{n}}^\top\widehat{\mathbf{R}}^\top\widehat{\mathbf{t}}} \tag{C.11}$$

if we write $\mathbf{J}_{H_T T} = \mathbf{J}_{H_T}\mathbf{J}_T$ it can be shown that:

$$\mathbf{J}_{H_T T} = \begin{bmatrix} \widehat{\mathbf{n}}\,(\widehat{\tau}_x\widehat{\mathbf{n}} + \mathbf{b}_x)^\top\widehat{\mathbf{R}}^\top & \widehat{\mathbf{H}}^\top\,[\widehat{\tau}_x\widehat{\mathbf{n}} + \mathbf{b}_x]_\times\,\mathbf{I} \\ \widehat{\mathbf{n}}\,(\widehat{\tau}_y\widehat{\mathbf{n}} + \mathbf{b}_y)^\top\widehat{\mathbf{R}}^\top & \widehat{\mathbf{H}}^\top\,[\widehat{\tau}_y\widehat{\mathbf{n}} + \mathbf{b}_y]_\times\,\mathbf{I} \\ \widehat{\mathbf{n}}\,(\widehat{\tau}_z\widehat{\mathbf{n}} + \mathbf{b}_z)^\top\widehat{\mathbf{R}}^\top & \widehat{\mathbf{H}}^\top\,[\widehat{\tau}_z\widehat{\mathbf{n}} + \mathbf{b}_z]_\times\,\mathbf{I} \end{bmatrix} \tag{C.12}$$

$$\mathbf{J}_{H_n} = \left[\nabla_{\mathbf{n}}\ \mathbf{H}(\widehat{\mathbf{T}}, \widehat{\mathbf{n}})^{-1}\mathbf{H}(\widehat{\mathbf{T}}, \mathbf{n})\right]_{\mathbf{n}=\widehat{\mathbf{n}}} \tag{C.13}$$

$$\mathbf{J}_n(\mathbf{0}) = \left[\nabla_{\mathbf{x}}\ \widehat{\mathbf{n}} + \mathbf{n}(\mathbf{x})\right]_{\mathbf{x}=\mathbf{0}} = \mathbf{I} \tag{C.14}$$

we can then also show that:

$$\mathbf{J}_{H_n n} = \mathbf{J}_{H_n}\mathbf{J}_n(\mathbf{0}) = \begin{bmatrix} (\widehat{\tau}_x\widehat{\mathbf{n}} + \mathbf{b}_x)^\top\widehat{\mathbf{R}}^\top\widehat{\mathbf{t}}\mathbf{J}_i \\ (\widehat{\tau}_y\widehat{\mathbf{n}} + \mathbf{b}_y)^\top\widehat{\mathbf{R}}^\top\widehat{\mathbf{t}}\mathbf{J}_i \\ (\widehat{\tau}_z\widehat{\mathbf{n}} + \mathbf{b}_z)^\top\widehat{\mathbf{R}}^\top\widehat{\mathbf{t}}\mathbf{J}_i \end{bmatrix} \tag{C.15}$$

with $\mathbf{J}_i = \left[\nabla_{\mathbf{n}}\ \ \frac{\widehat{\mathbf{n}}+\mathbf{n}}{\|\widehat{\mathbf{n}}+\mathbf{n}\|}\right]_{\mathbf{n}=\mathbf{0}}$ if $i = 1$ and $\mathbf{J}_i = \mathbf{I}_3$ for $i > 1$ (to take into account the normalisation of $\mathbf{n}_1$).

Reference Jacobian

We will decompose the reference Jacobian as such:

$$\begin{aligned} \mathbf{J}(\mathbf{x}_0) &= \left[\nabla_{\mathbf{x}}\ \mathcal{I}\left(\Pi(\mathbf{w}{<}\mathbf{H}(\mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}, \widehat{\mathbf{n}} + \mathbf{n}(\mathbf{x})){>}{<}\boldsymbol{\mathcal{X}}_s^*{>})\right)\right]_{\mathbf{x}=\mathbf{x}_0} \\ &= \mathbf{J}_{\mathcal{I}^*}\ \mathbf{J}_\Pi\ \mathbf{J}_w\ [\mathbf{J}_{H_T^* T^*}(\mathbf{x}_0)\ \ \mathbf{J}_{H_n^* n^*}(\mathbf{x}_0)] \end{aligned} \tag{C.16}$$

Noting that:

$$\mathbf{w}{<}\mathbf{H}(\mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}, \widehat{\mathbf{n}} + \mathbf{n}(\mathbf{x})){>}{<}\boldsymbol{\mathcal{X}}_s^*{>} = \mathbf{w}{<}\mathbf{H}(\overline{\mathbf{T}}, \overline{\mathbf{n}}){>}{<}\Pi^{-1}(\mathbf{q}){>}$$

with:

$$\mathbf{q} = \Pi(\mathbf{w}{<}\mathbf{H}(\overline{\mathbf{T}}, \overline{\mathbf{n}})^{-1}\mathbf{H}(\mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}, \widehat{\mathbf{n}} + \mathbf{n}(\mathbf{x})){>}{<}\boldsymbol{\mathcal{X}}_s^*{>}) \tag{C.17}$$

In $\mathbf{x}_0$ this leads to $\mathbf{q} = \Pi(\mathbf{w}<\mathbf{I}><\boldsymbol{\mathcal{X}}_s^*>)$ so the first Jacobian $\mathbf{J}_{\mathcal{I}^*}$ is:

$$\mathbf{J}_{\mathcal{I}^*} = \left[\nabla_{\mathbf{q}} \mathcal{I}\left(\Pi(\mathbf{w}<\mathbf{H}(\overline{\mathbf{T}}, \overline{\mathbf{n}})><\Pi^{-1}(\mathbf{q})>)\right)\right]_{\mathbf{q}=\mathbf{p}} \tag{C.18}$$

which is the jacobian of the reference image.

$\mathbf{J}_\Pi$ and $\mathbf{J}_w$ are the same as for the single plane tracking.

$$\mathbf{J}_{H_T^*} = \left[\nabla_{\mathbf{T}} \mathbf{H}(\overline{\mathbf{T}}, \overline{\mathbf{n}})^{-1}\mathbf{H}(\mathbf{T}\overline{\mathbf{T}}, \widehat{\mathbf{n}})\right]_{\mathbf{T}=\mathbf{I}} \tag{C.19}$$

$$\mathbf{J}_{T^*}(\mathbf{x}_0) = \left[\nabla_{\mathbf{x}} \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}\overline{\mathbf{T}}^{-1}\right]_{\mathbf{x}=\mathbf{x}_0} \tag{C.20}$$

$$\mathbf{J}_{H_n^*} = \left[\nabla_{\mathbf{T}} \mathbf{H}(\overline{\mathbf{T}}, \overline{\mathbf{n}})^{-1}\mathbf{H}(\widehat{\mathbf{T}}, \mathbf{n})\right]_{\mathbf{n}=\widehat{\mathbf{n}}} \tag{C.21}$$

$$\mathbf{J}_{n^*}(\mathbf{x}_0) = \left[\nabla_{\mathbf{x}} \widehat{\mathbf{n}} + \mathbf{n}(\mathbf{x})\right]_{\mathbf{x}=\mathbf{x}_0} = \mathbf{I} = \mathbf{J}_n(\mathbf{0}) \tag{C.22}$$

Trivially we have $\mathbf{J}_n(\mathbf{0})\mathbf{x}_0 = \mathbf{J}_{n^*}(\mathbf{x}_0)\mathbf{x}_0$ but also, thanks to the Lie group parameterization, it can be shown that $\mathbf{J}_{T^*}(\mathbf{x}_0)\mathbf{x}_0 = \mathbf{J}_T(\mathbf{0})\mathbf{x}_0$. If we now make the approximation that $\widehat{\mathbf{T}} \approx \overline{\mathbf{T}}$ and $\widehat{\mathbf{n}} \approx \overline{\mathbf{n}}$, $\mathbf{J}_{H_T^*} \approx \mathbf{J}_{H_T}$ and $\mathbf{J}_{H_n^*} \approx \mathbf{J}_{H_n}$ and we obtain equation 6.18.

Appendix D

The Kalman filter

The Kalman filter was used for the experiments of Chapter 9, we will now describe the related equations.

D.1 Discrete Kalman Filter (KF)

The discrete Kalman filter addresses the problem of trying to estimate the discrete time-controlled process governed by the equations:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \mathbf{w}_{t-1}$$

with a measurement $\mathbf{z}$ that can be written:

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t$$

- $\mathbf{w}_t$ represents the process noise, $\mathbf{w} \sim N(0, \mathbf{Q})$.
- $\mathbf{v}_t$ represents the measurement noise, $\mathbf{v} \sim N(0, \mathbf{R})$.

Prediction and time update equations

$$\mathbf{x}_{t+1/t} = \mathbf{A}\mathbf{x}_{t/t} + \mathbf{B}\mathbf{u}_t$$
$$\mathbf{P}_{t+1/t} = \mathbf{A}\mathbf{P}_{t/t}\mathbf{A}^\top + \mathbf{Q}_t$$

Measurement update equations

$$\mathbf{x}_{t+1/t+1} = \mathbf{x}_{t+1/t} + \mathbf{W}_{t+1}\mathbf{v}_{t+1}$$
$$\mathbf{P}_{t+1/t+1} = \mathbf{P}_{t+1/t} - \mathbf{W}_{t+1}\mathbf{S}_{t+1}\mathbf{W}_{t+1}^\top$$

The measurement update equations add the information from the new measurements to correct the estimate from the model. $\mathbf{v}$ is called the "innovation" and corresponds to the amount of "unpredicted" information obtained from the new measurement. $\mathbf{W}$ is the Kalman gain and expresses how much trust we can have in the measurement.

$$\mathbf{v}_{t+1} = \mathbf{z}_{t+1} - \mathbf{H}\mathbf{x}_{t+1/t}$$

$$\mathbf{W}_{t+1} = \mathbf{P}_{t+1/t}\mathbf{H}_{t+1/t}^{\top} S_{t+1}^{-1}$$

$$\mathbf{S}_{t+1} = \mathbf{H}_{t+1/t}\mathbf{P}_{t+1/t}\mathbf{H}_{t+1/t}^{\top} + \mathbf{R}_{t+1}$$

$\mathbf{R}$ is the measurement noise covariance.

D.2 Extended Kalman Filter (EKF)

The state transition and measurement equations are often non-linear. The Extended Kalman Filter (EKF) is an extension of the Kalman filter to cope with these non-linearities. The mathematical simplifications involves come however at a price: the distributions are not correctly modelled and the linearisations will lead to inconsistencies. In practice however, the results obtained are often satisfactory.

Prediction and time update equations

$$\mathbf{x}_{t+1/t} = f(\mathbf{x}_{t/t}, \mathbf{u}_t)$$

$$\mathbf{P}_{t+1/t} = (\nabla_{\mathbf{x}} f)_{t/t}\mathbf{P}_{t/t}(\nabla_{\mathbf{x}} f)_{t/t}^{\top} + \mathbf{Q}_t$$

f is the state update equation.
$\mathbf{x}_{t/t}$ is the state estimate at time k based on the information at time k.
$\mathbf{x}_{t+1/t}$ is the state estimate at time k+1 based on the time update model (ie without integrating the measurement information).
$\mathbf{P}$ correspond to the covariance matrices.
$\mathbf{Q}$ is the process noise covariance.

Measurement update equations

$$\mathbf{x}_{t+1/t+1} = \mathbf{x}_{t+1/t} + \mathbf{W}_{t+1}\mathbf{v}_{t+1}$$

$$\mathbf{P}_{t+1/t+1} = \mathbf{P}_{t+1/t} - \mathbf{W}_{t+1}\mathbf{S}_{t+1}\mathbf{W}_{t+1}^{\top}$$

The measurement update equations add the information from the new measurements to correct the estimate from the model. $\mathbf{v}$ is called the "innovation" and corresponds to the amount of "unpredicted" information obtained from the new measurement. $\mathbf{W}$ is the Kalman gain and expresses how much trust we can have in the measurement.

$$\mathbf{v}_{t+1} = \mathbf{z}_{t+1} - h(\mathbf{x}_{t+1/t})$$

$$\mathbf{W}_{t+1} = \mathbf{P}_{t+1/t}(\nabla_{\mathbf{x}} h)_{t+1/t}^{\top}\mathbf{S}_{t+1}^{-1}$$

$$\mathbf{S}_{t+1} = (\nabla_{\mathbf{x}} h)_{t+1/t}\mathbf{P}_{t+1/t}(\nabla_{\mathbf{x}} h)_{t+1/t}^{\top} + \mathbf{R}_{t+1}$$

$\mathbf{R}$ is the measurement noise covariance.

Bibliography

Migratory Birds, volume 179. Bird Talk Magazine, 2005.

Ercan U. Acar, Howie Choset, Alfred A. Rizzi, Prasad N. Atkar, and Douglas Hull. Morse decompositions for coverage tasks. *International Jounal of Robotics Research*, 21(4):331–344, 2002.

Nicolas Andreff, Bernard Espiau, and Radu Horaud. Visual servoing from lines. *International Journal of Robotics Research*, 21(8):679–700, August 2002.

N. Ayache and O. D. Faugeras. Building, registrating, and fusing noisy visual maps. *International Journal of Robotics Research*, 7(6):45–65, 1988.

Tim Bailey and Hugh Durrant-Whyte. Simultaneous localisation and mapping (slam): Part ii - state of the art. *Robotics and Automation Magazine*, 2006.

Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the ekf-slam algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006a.

Tim Bailey, Juan Nieto, and Eduardo Nebot. Consistency of the fastslam algorithm. In *IEEE International Conference on Robotics and Automation*, 2006b.

S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *CVPR*, pages 1090–1097, 2001.

S. Baker, R. Patil, K.M. Cheung, and I. Matthews. Lucas-kanade 20 years on: Part 1. Technical report, Robotics Institute, Carnegie Mellon University, 2004.

Simon Baker and Shree K. Nayar. A theory of catadioptric image formation. In *ICCV*, pages 35–42, 1998.

Simon Baker, Ankur Datta, and Takeo Kanade. Parameterizing homographies. Technical report, Robotics Institute, Carnegie Mellon University, March 2006.

Hynek Bakstein and Tomás Padjla. An overview of non-central cameras. In *Computer Vision Winter Workshop*, 2001.

J. Barreto, F. Martin, and R. Horaud. Visual servoing/tracking using central catadioptric cameras. In *International Symposium on Experimental Robotics*, Advanced Robotics Series, 2002.

J. P. Barreto and H. Araujo. Direct least square fitting of paracatadioptric line images. In *OMNIVIS*, 2003.

João P. Barreto. *General Central Projection Systems, modeling, calibration and visual servoing.* PhD thesis, Department of electrical and computer engineering, 2003.

A. Bartoli. Groupwise geometric and photometric direct image registration. In *British Machine Vision Conference*, 2006.

A. Bartoli and P. Sturm. The 3d line motion matrix and alignement of line reconstructions. *IJCV*, 57 (3):159–178, 2004.

A. Bartoli and P. Sturm. Structure from motion using lines: Representation, triangulation and bundle adjustment. *CVIU*, 100(3):416–441, 2005.

S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE International Conference on Intelligent Robots and Systems*, 2004.

S. Benhimane and E. Malis. Homography-based 2d visual tracking and servoing. *Joint Issue of the International Journal of Computer Vision and the International Journal of Robotic Research, 2006.*, 2006.

R. Benosman and S. B. Kang. A brief historical perspective on panorama. In *Panoramic Vision*, pages 5–20. Apr 2001.

P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

Peter Biber and Wolfgang Strasser. nscan-matching: Simultaneous matching of multiple scans and application to slam. In *IEEE International Conference on Robotics and Automation*, 2006.

Peter Biber, Sven Fleck, and Wolfgang Straßer. A probabilistic framework for robust and accurate matching of point clouds. In *26th Pattern Recognition Symposium (DAGM 04)*, 2004.

R. Biswas, B. Limketkai, Sanner S., and S. Thrun. Towards object mapping in non-stationary environments with mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

M. Bosse, R. Rikoski, J. Leonard, and S. Teller. Vanishing points and 3d lines from omnidirectional video. In *ICIP*, 2002.

M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *IEEE International Conference on Robotics and Automation*, 2003.

P. Bouthemy. A maximum likelihood framework for determining moving edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):499–511, 1989.

C. Brauer-Burchardt and K. Voss. A new algorithm to correct fish-eye- and strong wide-angle-lens-distortion from single images. In *International Conference on Image Processing*, volume 1, pages 225–228, Oct 2001.

J. E. Bresenham. Algorithm for computer control of a digital plotter. *IDM Systems Journal*, 4(1), 1965.

T.J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, 26(4):639–656, 1990.

J. M. Buenaposada and L. Baumela. Real-time tracking and estimation of planar pose. In *ICPR*, pages 697–700, 2002.

Wolfram Burgard, Dieter Fox, Hauke Jans, Christian Matenar, and Sebastian Thrun. Sonar-based mapping of large-scale mobile robot environments using em. In *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*, pages 67–76, 1999.

N.A. Carlson. Federated square root filters for decentralized parallel processes. *IEEE Transactions on Aerospace and Electronic Systems*, 26(3), 1990.

C. Charron, O. Labbani-Igbida, and El Mustapha Mouaddib. Qualitative localization using omnidirectional images and invariant features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

Cyril Charron, Ouiddad Labbani-Igbida, and El Mustapha Mouaddib. *On Building Omnidirectional Image Signatures Using Haar Invariant Features: Application to the Localization of Robots*, volume 4179 of *Lecture Notes in Computer Science*, pages 1099–1110. Springer Berlin / Heidelberg, 2006.

R. Chatila and J.-P. Laumond. Position referencing and consistent world modeling for mobile robots. In *IEEE International Conference on Robotics and Automation*, 1985.

Y. Chen and Medioni. Object modeling by registration of multiple range images. In *IEEE International Conference on Robotics and Automation*, 1991.

K. Chia, A. Cheok, and S. Prince. Online 6 dof augmented reality registration from natural features. In *IEEE International Symposium on Mixed and Augmented Reality*, 2002.

A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

H. Choset and K. Nagatani. Topological simultaneous localization and mapping (slam): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2): 125–137, 2001.

Robert T. Collins, Yanxi Liu, and Marius Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1631–1643, 2005.

D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575, 2003.

Michael Csorba. *Simultaneous Localisation and Map Building*. PhD thesis, Department of Engineering Science, University of Oxford, 1997.

A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *International Conference on Computer Vision*, 2003.

A.J. Davison, Y. González Cid, and N. Kita. Real-time 3D SLAM with wide-angle vision. In *IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.

Andrew J. Davison. Active search for real-time vision. In *International Conference on Computer Vision*, 2005.

M. Deans and M. Hebert. Experimental comparison of techniques for localization and mapping using a bearings only sensor. In *Seventh International Symposium on Experimental Robotics*, 2000.

L Delahoche, C Pégard, B Marhic, and P Vasseur. A navigation system based on an omnidirectional vision sensor. In *IEEE International Conference on Intelligent Robots and Systems*, 1997.

Cédric Demonceaux, Pascal Vasseur, and Claude Pégard. Robust attitude estimation with catadioptric vision. In *IEEE International Conference on Intelligent Robots and Systems*, 2006.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:185–197, 1977.

Frederic Devernay and Olivier D. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24, 2001.

G. Dissanayake, P. Newman, H. F. Durrant-Whyte, S. Clark, , and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

Arnaud Doucet, Nando de Freitas, Kevin Murphy, and Stuart Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Uncertainty in Artificial Intelligence*, 2000.

T.W. Drummond and R. Cipolla. Application of lie algebras to affine invariant visual servoing. *Int. Journal of Computer Vision*, 37(1):21–41, 2000.

T.W. Drummond and R. Cipolla. Visual tracking and control using lie algebras. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 1999.

Romain Dupont, Renaud Keriven, and Philippe Fuchs. An improved calibration technique for coupled single-row telemeter and ccd camera. *3DIM*, 00:89–94, 2005. ISSN 1550-6185.

Hugh Durrant-Whyte. Localisation, mapping and the simultaneaous localisation and mapping (slam) problem. SLAM Summer School, 2002.

Hugh Durrant-Whyte and Tim Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *Robotics and Automation Magazine*, 2006.

E. D. Eade and T. W. Drummond. Edge landmarks in monocular slam. In *British Machine Vision Conference*, 2006.

A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, 1988.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

Andrew W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *CVPR*, 2001.

D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.

Jean Gallier. *Geometric Methods and Applications For Computer Science and Engineering*. Springer-Verlag, 2001.

C. Geyer and K. Daniilidis. Catadioptric projective geometry. *IJCV*, 45(3):223–243, 2001.

Christopher Geyer. *Catadioptric Projective Geometry : theory and applications*. PhD thesis, University of Pennsylvania, 2003.

Christopher Geyer and Konstantinos Daniilidis. A unifying theory for central panoramic systems and practical applications. In *European Conference on Computer Vision*, pages 445–461, 2000.

C. Giovannangeli, Ph. Gaussier, and J.-P. Banquet. Robustness of visual place cells in dynamic indoor and outdoor environment. *International Journal of Advanced Robotic Systems*, 3(2):115–124, 2006.

José-Joel Gonzalez-Barbosa and Simon Lacroix. Rover localization in natural environments by indexing panoramic images. In *IEEE International Conference on Robotics and Automation*, 2002.

Paul Graham and Thomas S. Collett. View-based navigation in insects: how wood ants (formica rufa l.) look at and are guided by extended landmarks. *The Journal of Experimental Biology*, (205): 2499–2509, 2002.

J. E. Guivant and E. M. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotic and Automation*, 17(3): 242–257, 2000.

J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1999.

H. Hadj-Abdelkader, Y. Mezouar, N. Andreff, and P. Martinet. 2 1/2 d visual servoing with central catadioptric cameras. In *IEEE International Conference on Intelligent Robots and Systems*, 2005.

G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

D. Hähnel, W. Burgard, D. Fox, and S. Thrun. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003a.

Dirk Hähnel, Rudolph Triebel, Wolfram Burgard, and Sebastian Thrun. Map building with mobile robots in dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003b.

Brian C. Hall. An elementary introduction to groups and representations, 2000.

R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, 1973.

Richard Hartley and Andrew Zisserman. *Multiple View geometry in Computer vision*. Cambridge university press, 2000.

Janne Heikkilä. Geometric camera calibration using circular control points. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

J. Hong, X. Tan, B. Pinette, R. Weiss, and E.M. Riseman. Image-based homing. In *IEEE International Conference on Robotics and Automation*, 1991.

Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART C: APPLICATIONS AND REVIEWS*, 34(3), 2004.

Jing Huang, M.and Wei-Jing Zhu Kumar, S.R.and Mitra, and R. Zabih. Image indexing using color correlograms. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

Takahiro Ishikawa, Iain Matthews, and Simon Baker. Efficient image alignment with outlier rejection. Technical report, Robotics Institute, Carnegie Mellon University, 2002.

Simon J. Julier and Jeffery K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management*, 1997.

S.J. Julier and J.K. Uhlmann. Simultaneous localization and map building using split covariance intersection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001.

K. Konolige. Slam via variable reduction from constraint maps. In *IEEE International Conference on Robotics and Automation*, 2005.

K. Konolige, S. Gutmann, D. Guzzoni, R. Ficklin, and K. Nicewarner. Mobile robot sense net. In *Sensor Fusion and Decentralized Control in Robotic Systems*, 1999.

Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63, 1991.

N. M. Kwok and G. Dissanayake. An efficient multiple hypothesis filter for bearing-only slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

D. Lambrinos, R. oller, T. Labhart, R. Pfeifer, and R. Wehner. A mobile robot employing insect strategies for navigation. *Robotics and Autonomous Systems, special issue: Biomimetic Robots*, 1999.

P. Y. Lee and J. B. Moore. Pose estimation via a gauss-newton-on-manifold approach. In *16th International Symposium on Mathematical Theory of Network and System (MTNS)*, 2004.

Thomas Lemaire and Simon Lacroix. Slam with panoramic vision. Technical report, LAAS-CNRS, 2006. submitted to the Journal for Fields Robotics in the special issue SLAM in the Fields.

J. Leonard and H. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In D. Koditschek J. Hollerbach, editor, *International Symposium on Robotics Research*, 1999.

J. Leonard and P. Newman. Consistent, convergent, and constant-time slam. In *International Joint Conference on Artificial Intelligence*, 2003.

J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, 1991.

V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, October 2005.

V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Computer Vision and Pattern Recognition*, June 2005.

Shih-Schön Lin and Ruzena Bajcsy. Single-view-point omnidirectional catadioptric cone mirror imager. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):840–845, May 2006.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60):91–110, 2004.

F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997a.

Feng Lu and Evangelos Milios. Robot pose estimation in unknown environments by matching 2d range scans. *Journal of Intelligent and Robotic Systems*, 18(3):249–275, 1997b.

Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

Yi Ma, Stefano Soatto, Jana Kosecka, and S. Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer-Verlag, 2003. ISBN 0387008934.

K. Madsen, H. Nielsen, and O. Tingleff. Methods for non-linear least squares problems. Technical report, Technical University of Denmark, 2004.

E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *IEEE International Conference on Robotics and Automation*, 2004.

J. Mallon and P. Whelan. Precise radial un-distortion of images. In *ICPR*, 2004.

E. Marchand and F. Chaumette. Feature tracking for visual servoing purposes. *Robotics and Autonomous Systems*, 52(1):53–70, 2005.

J. Matas and O. Chum. Randomized ransac with t(d,d) test. In *British Machine Vision Conference*, 2002.

C. Mei and E. Malis. Fast central catadioptric line extraction, estimation, tracking and structure from motion. In *IEEE International Conference on Intelligent Robots and Systems*, October 2006.

C. Mei and P. Rives. Calibrage non biaise d'un capteur central catadioptrique. In *RFIA*, January 2006a.

C. Mei and P. Rives. Calibration between a central catadioptric camera and a laser range finder for robotic applications. In *IEEE International Conference on Robotics and Automation*, May 2006b.

C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, April 2007.

C. Mei, S. Benhimane, E. Malis, and P. Rives. Homography-based tracking for central catadioptric cameras. In *IEEE International Conference on Intelligent Robots and Systems*, October 2006a.

C. Mei, S. Benhimane, E. Malis, and P. Rives. Constrained multiple planar template tracking for central catadioptric cameras. In *British Machine Vision Conference*, September 2006b.

Y. Mezouar, H. Haj Abdelkader, P. Martinet, and F. Chaumette. Central catadioptric visual servoing from 3d straight lines. In *IEEE International Conference on Intelligent Robots and Systems*, 2004.

Branislav Micusik. *Two-view geometry of Omnidirectional Cameras*. PhD thesis, Center for Machine Perception, Czech Technical University, 2004.

K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

N. Molton, A. Davison, and I. Reid. Locally planar patch features for real-time structure from motion. In *British Machine Vision Conference*, 2004.

Michael Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.

H. Moravec and M. Martin. Robot spatial perception by stereoscopic vision and 3d evidence grids. Technical report, Mobile Robot Laboratory, Robotics Institute, Carnegie Mellon University, 1996.

H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, 1988.

E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real-time localization and 3d reconstruction. In *IEEE Conference of Vision and Pattern Recognition*, 2006.

Philippe Moutarlier and Raja Chatila. An experimental system for incremental environment modelling by an autonomous mobile robot. In *International Symposium on Experimental Robotics*, 1989.

S.K. Nayar and V.N. Peri. Folded Catadioptric Cameras. In *Panoramic Vision*, pages 103–119. Apr 2001.

Eduardo Nebot, Favio Masson, Jose Guivant, and H. Durrant-Whyte. *Experimental Robotics VIII*, volume 5, chapter Robust Simultaneous Localization and Mapping for Very Large Outdoor Environments, pages 200–209. Springer Berlin / Heidelberg, 2003.

J. Neira and J.D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.

P. Newman, J. Leonard, J. D. Tardos, and J. Neira. Explore and return: experimental validation of real-time concurrent mapping and localization. In *IEEE International Conference on Robotics and Automation*, 2002.

Paul Newman, David Cole, and Kin Ho. Outdoor slam using visual appearance and laser ranging. In *IEEE International Conference on Robotics and Automation*, 2006.

J. Nieto, T. Bailey, and E. Nebot. Recursive scan-matching slam. *Journal of Robotics and Autonomous Systems*, 2006. In press.

David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1), 2006.

T. Pajdla. Stereo with oblique cameras. *International Journal of Computer Vision*, 47(1):161–170, 2002.

Tomas Pajdla and Vaclav Hlavac. Zero phase representation of panoramic images for image vased localization. In *Computer Analysis of Images and Patterns*, pages 550–557, 1999.

Mark A. Paskin. Thin junction tree filters for simultaneous localization and mapping. Computer Science Division Technical Report CSD-02-1198, University of California, Berkeley, September 2002.

S. Pfister, K. Kriechbaum, S. Roumeliotis, and J. Burdick. Weighted range sensor matching algorithms for mobile robot displacement estimation. In *IEEE International Conference on Robotics and Automation*, 2002.

J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *Computer Vision and Pattern Recognition*, June 2005.

P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *European Conference on Computer Vision*, 2002.

D. W. Rees. Panoramic television viewing system. United States Patent No. 3, 505, 465, April 1970.

A. Remazeilles, F. Chaumette, and P. Gros. 3d navigation based on a visual memory. In *IEEE International Conference on Robotics and Automation*, 2006.

Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *IEEE International Conference on Computer Vision*, 1998.

Bernt Schiele and James L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, 2000.

J.G. Semple and G.T. Kneebone. *Algebraic Projective Geometry*. Clarendon Press, 1979.

Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

H. Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *International Journal on Computer Vision*, 16(1):63–84, 2000.

S. Siggelkow and H. Burkhardt. Improvement of histogram-based image retrieval and classification. In *International Conference on Pattern Recognition*, 2002.

G. Silveira, E. Malis, and P. Rives. An efficient direct method for improving visual slam. In *IEEE International Conference on Robotics and Automation*, 2007. Submitted.

Geraldo Silveira, Ezio Malis, and Patrick Rives. Visual servoing over unknown, unstructured, large-scale scenes. In *IEEE International Conference on Robotics and Automation*, 2006.

G. Simon, A. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *IEEE International Symposium on Mixed and Augmented Reality*, 2000.

Sudipta N Sinha, Jan-Michael Frahm, Marc Pollefeys, and Yakup Genc. Gpu-based video feature tracking and matching. In *EDGE 2006, workshop on Edge Computing Using New Commodity Architectures*, 2006.

P. Smith. *Edge-based Motion Segmentation*. PhD thesis, Cambridge University Engineering Department, 2001.

P. Smith, T. Drummond, and R. Cipolla. Layered motion segmentation and depth ordering by tracking edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):479–494, 2004.

P. Smith, I. Reid, and A. J. Davison. Real-time monocular slam with straight lines. In *British Machine Vision Conference*, 2006.

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag, Berlin-Heidelberg, 1990.

Randall Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5:56–68, 1986.

J. Solà, M. Devy, A. Monin, and T. Lemaire. Undelayed initialization in bearing only slam. In *IEEE International Conference on Intelligent Robots and Systems*, 2005.

Peter Sturm. Multi-view geometry for general camera models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 206–212, June 2005.

Peter Sturm, Srikumar Ramalingam, and Suresh Lodha. On calibration, structure from motion and multi-view geometry for generic camera models. In Kostas Daniilidis and Reinhard Klette, editors, *Imaging Beyond the Pinhole Camera*, volume 33 of *Computational Imaging and Vision Series*. Springer, 2006.

Tomás Svoboda, Tomás Pajdla, and Václav Hlavác. Epipolar geometry for panoramic cameras. *Lecture Notes in Computer Science*, 1406, 1998.

R. Swaminathan, M. D. Grossberg, and S. K. Nayar. Non-Single Viewpoint Catadioptric Cameras: Geometry and Analysis. *International Journal of Computer Vision*, 66(3):211–229, Mar 2006.

Rahul Swaminathan, Michael Grossberg, and Shree Nayar. Caustics of catadioptric cameras. In *International Conference On Computer Vision*, 2001.

Jean-Philippe Tardif, Peter Sturm, and Sébastien Roy. Self-calibration of a general radially symmetric distortion model. In *European Conference on Computer Vision*, 2006.

J. Tardos, J. Neira, P. Newman, and J. Leonard. Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotics Research*, 21(4):311–330, 2002.

C. J. Taylor and D. J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1021–1032, 1995.

S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.

S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000.

S. Thrun, C. Martin, Y. Liu, D. fihnel, R. Emery-Muntemerlo, D. Chakrabarti, and W. Burgard. A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor en- vironmcnts with mobile robots. *IEEE Transactions on Robotics and Automation*, 2003.

S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7/8), 2004.

Sebastian Thrun. Robotic mapping: A survey. Technical report, Carnegie Mellon University, 2002.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1-3):29–53, 1998.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.

B. Tordoff, W. Mayol, T. de Campos, and D. Murray. Head pose estimation for wearable robot control. In *British Machine Vision Conference*, 2002.

Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms*, Computer Science. Springer Berlin / Heidelberg, 1999.

I. Ulrich and I Nourbakhsh. Appearance-based place recognition for topological localization. In *IEEE International Conference on Robotics and Automation*, 2000.

J. Valls Miro, W. Zhou, and G. Dissanayake. Towards vision based navigation in large indoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

V.S. Varadarajan. *Lie Groups, Lie Algebras, and Their Representations*. Springer-Verlag, 1974.

P. Vasseur and E. M. Mouaddib. Central catadioptric line detection. In *British Machine Vision Conference*, 2004.

Alessandro Victorino. *La commande référencée capteur: une approche robuste au problème de navigation, localisation et cartographie simultanées pour un robot mobile d'extérieur*. PhD thesis, Université de Nice, INRIA Sophia Antipolis, 2002.

M. Walter, R. Eustice, and J. Leonard. A provably consistent method for imposing exact sparsity in feature-based slam information filters. In *Proceedings of the 12th International Symposium of Robotics Research (ISRR)*, 2005.

C.-C. Wang, C. Thorpe, and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *IEEE International Conference on Robotics and Automation*, 2003.

K. Weber, S. Venkatesh, and M. Srinivasan. Insect-inspired robotic homing. *Adaptive Behavior*, 1998.

J. Weng, P. Cohen, and M Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, 1992.

S. Williams and I. Mahon. Simultaneous localisation and mapping on the great barrier reef. In *IEEE International Conference on Robotics and Automation*, 2004.

Denis Wolf and Gaurav S. Sukhatme. Online simultaneous localization and mapping in dynamic environments. In *IEEE International Conference on Robotics and Automation*, 2004.

L. Xu and E. Oja. Randomized hough transform (rht) : Basic mechanisms, algorithms, and computational complexities. *Computer Vision, Graphics and Image Processing*, 57(2):131–154, 1993.

Y. Yagi and S. Kawato. Panoramic scene analysis with conic projection. In *International Conference on Robots and Systems*, 1990.

Y Yagi and M Yachida. Real-time generation of environmental map and obstacle avoidance using omnidirectional image sensor with conic mirror. In *CVPR*, 1991.

Yasushi Yagi. Omnidirectional sensing and its applications. *IEICE Trans, on Information and Systems*, E82-D(3):568–579, 1999.

K. Yamazawa, Y. Yagi, and M. Yachida. Omnidirectional imaging with hyperboloidal projection. In *Internation Conference on Robotics and Automation*, 1993.

Y. Yamazawa, K.and Yagi and M. Yachida. 3d line segment reconstruction by using hyperomni vision and omnidirectional hough transforming. In *ICPR*, 2000.

Gehua Yang, Charles V. Stewart, Michal Sofka, and Chia-Ling Tsai. Registration of challenging image pairs: initialization, estimation, and decision. Technical report, Department of Computer Science, Rensselaer Polytechnic Institute, 2006. URL `http://www.vision.cs.rpi.edu/gdbicp/`.

Xianghua Ying and Zhanyi Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model ? In *European Conference on Computer Vision*, 2004.

Q. Zhang and R. Pless. Extrinsic calibration of a camera and laser range finder. In *IEEE International Conference on Intelligent Robots and Systems*, 2004.

List of Figures

List of Tables

Résumé

Estimer le mouvement d'un robot et construire en même temps une représentation de l'environnement (problème SLAM: Simultaneous Localisation And Mapping) est souvent considéré comme un problème essentiel pour développer des robots pleinement autonomes qui ne nécessitent pas de connaissances a priori de l'environnement pour réaliser leurs tâches.

L'évolution du SLAM est très liée aux capteurs utilisés. Les sonars couplée avec l'odométrie sont souvent présentés comme les premiers capteurs ayant fourni des résultats convaincants. Depuis, les lasers 2D ont souvent remplacés ces capteurs pour des raisons de précision et de rapport signal/bruit. Néanmoins les lasers 2D permettent uniquement d'estimer des mouvements planaires et ne donnent pas des informations perceptuelles suffisantes pour identifier de manière fiable des régions précédemment explorées.

Ces observations nous ont amenés à explorer à travers cette thèse comment combiner un capteur omnidirectionnel avec un télémètre laser pour effectuer la localisation et cartographie simultanée dans des environnements complexes et de grandes tailles.

Les contributions de cette thèse concernent l'étalonnage des capteurs centraux catadioptriques (avec le développement d'un logiciel opensource disponible sur le site internet de l'auteur) et la recherche de la position relative entre un capteur omnidirectionnel et un télémètre laser. Des approches efficaces pour estimer le mouvement 3D du capteur en utilisant des droites et des plans sont détaillées. Enfin deux méthodes sont proposées combinant laser et vision pour effectuer du SLAM planaire mais aussi pour estimer la position 3D du robot ainsi que la structure de l'environnement.

Mots clefs : vision omnidirectionnelle, télémétrie laser, cartographie, estimation du mouvement, suivi basé vision

Abstract

The problem of estimating the motion of a robot and simultaneously building a representation of the environment (known as SLAM: Simultaneous Localisation And Mapping) is often considered as an essential topic of research to build fully autonomous systems that do not require any prior knowledge of the environment to fulfill their tasks.

The evolution of SLAM is closely linked to the sensors used. Sonars with odometry are often presented as the first sensors having led to convincing results. Since then, 2D laser range finders have often replaced sonars when possible because of the higher precision and better signal to noise ratio. However 2D lasers alone limit SLAM to planar motion estimation and do not provide sufficiently rich information to reliably identify previously explored regions.

These observations have led us to explore throughout this thesis how to combine an omnidirectional camera with a laser range finder to help solve some of the challenges of SLAM in large-scale complex environments.

The contributions of this thesis concern a method to calibrate central catadioptric cameras (with the development of an opensource toolbox available on the author's website) and find the relative position between an omnidirectional sensor and a laser range finder. How to represent lines and planes for motion estimation is also studied with the use of Lie algebras to provide a minimal parameterisation. Finally we will detail how laser and vision can be combined for planar SLAM and 3D structure from motion.

Key words: omnidirectional vision, laser range finder, mapping, SLAM, motion estimation, visual tracking