Etude du couvert forestier à partir d'un processus objet

Guillaume PERRIN

Rapport de stage de DEA - Septembre 2003
Encadrement : Xavier Descombes et Josiane Zerubia

Projet Ariana - CNRS/INRIA/UNSA

Table des matières

Introduction **5**

1 Etat de l'art en matière de détection de houppiers **7**
1.1 Introduction . . . 7
1.2 L'approche contour . . . 8
1.2.1 Le principe . . . 8
1.2.2 Les images de données . . . 8
1.2.3 Etapes de prétraitement . . . 9
1.2.4 Principe du suivi de vallées . . . 9
1.2.5 Règles d'obtention des contours : le rule-based system . . . 12
1.2.6 Résultats et interprétations . . . 15
1.3 L'approche objets . . . 16
1.3.1 Le principe . . . 16
1.3.2 Récapitulatif : l'algorithme de M. Larsen . . . 17

2 Processus objet : définitions et simulation **19**
2.1 Les processus objet : définitions et exemples . . . 19
2.1.1 Première approche des processus ponctuels . . . 19
2.1.2 Définition d'un processus ponctuel par une densité non normalisée . . . 24
2.1.3 Processus ponctuels de Markov . . . 27
2.2 Chaînes de Markov et Théorie ergodique . . . 32
2.2.1 Notions sur les chaînes de Markov . . . 32
2.2.2 Etude de la stabilité des chaînes de Markov . . . 36
2.3 Algorithme de simulation : Metropolis-Hastings-Green . . . 38
2.3.1 Présentation de l'algorithme . . . 38
2.3.2 Convergence de la chaîne de Markov considérée . . . 41
2.3.3 Quelques pistes à explorer . . . 48

3 Notre modèle pour la détection des houppiers **51**
3.1 Analyse des données : définition des objets d'intérêt . . . 51
3.1.1 Des sources d'images Très Haute Résolution . . . 51
3.1.2 Objectifs fixés . . . 53
3.1.3 Objets d'intérêt . . . 56
3.2 Modèle de la loi objectif . . . 58
3.2.1 Cadre Bayésien . . . 58
3.2.2 Terme a priori . . . 59
3.2.3 Terme d'attaches aux données . . . 60
3.2.4 Stabilité du processus . . . 64
3.2.5 Estimateurs . . . 64
3.3 Noyau de proposition adopté . . . 67
3.3.1 Mouvements simples . . . 67
3.3.2 Mouvements complexes . . . 72

3.4 Bilan : notre algorithme 73

4 Premiers résultats sur des images réelles et Interprétations **75**
4.1 Images étudiées 75
4.2 Résultats et Interprétations 77
4.2.1 Choix des paramètres 77
4.2.2 Exemples de résultats obtenus 78
4.2.3 Interprétations 78
4.3 Comparaison avec le suivi de vallées 84
4.3.1 Résultats obtenus avec le suivi de vallées 84
4.3.2 Tableau comparatif 85

Conclusion et Perspectives **87**

Introduction

Cadre de travail

Ce stage de DEA s'est déroulé au sein du ***projet Ariana***, à l'***Institut National de Recherche en Informatique et en Automatique*** (INRIA) Sophia-Antipolis. Créé en tant que projet commun CNRS/INRIA/UNSA en 1998, Ariana regroupe son activité autour de l'étude des problèmes inverses en observation de la Terre, par des approches variationnelle ou probabiliste.

Notre travail s'inscrit dans une collaboration avec l'***Inventaire Forestier National*** (IFN) et le ***Cemagref***, tous deux basés à Montpellier, dans le cadre d'un projet COLORS.

Besoins en géoréférencement

De nombreuses données sur les milieux forestiers sont nécessaires aux organismes de gestion des forêts (ONF,...) ou chargés d'en effectuer l'inventaire (IFN). Ces besoins comprennent aussi bien des besoins généraux de gestion des forêts, que des besoins plus particuliers comme celui de cartographier les types de peuplements en forêts de montagne par exemple.

En raison des progrès technologiques réalisés en matière de résolution des images (aériennes ou satellitaires), et de la baisse du coût d'obtention de telles images, de nouvelles perspectives s'ouvrent dans la connaissance des arbres et des forêts. Ces données pourraient en effet servir à évaluer certains paramètres de la biodiversité en fournissant des informations plus précises sur des arbres et leur situation (isolés, en rangées, en bosquets ou en peuplement) dans les paysages européens, ou à mieux évaluer le reboisement se produisant dans les secteurs sujets à l'abandon de terre, anciens pâturages ou terres agricoles moins productives en zones défavorisées, ou périurbaines.

Dans ce contexte, un outil pour aider les forestiers, les gestionnaires du territoire et les écologues du paysage dans l'inventaire des arbres isolés, en alignement, en bosquet ou en forêt, serait le bienvenu. Si cet outil pouvait servir à estimer d'autres paramètres, tels que la densité, le taux de couvert, la taille et le diamètre des arbres, jusqu'à présent renseignés uniquement par des travaux sur le terrain, cela s'avérerait fortement utile du point de vue économique.

Objectifs de notre étude

L'objectif de ce stage, qui se poursuivra par une thèse, est ainsi de parvenir à une identification semi-automatique des ***houppiers*** (partie supérieure d'un arbre, comprenant l'ensemble des branches et les feuilles) sur des images aériennes de peuplements forestiers.

Cet objectif doit être atteint avec différentes espèces d'arbre (bouleaux, chênes, pins, sapins, épicéas, ...), avec différentes formes de couronne, des conditions variables d'illumination, de pente et d'exposition.

Une nouvelle approche

Aucune information sur la répartition spatiale et les intéractions entre houppiers n'est prise en compte dans le cadre des approches usuelles d'analyse. La première partie de ce rapport sera d'ailleurs consacrée à un état de l'art des méthodes de détection de houppiers existantes.

L'approche que nous avons choisi d'adopter utilise des ***processus ponctuels marqués***. Leur utilisation en traitement d'image a été proposée par A. J. Baddeley et M.N.M. van Lieshout dans [35] sur un cas d'école très simple. Notre objectif est de pouvoir traiter des images réelles. Ces processus ponctuels marqués semblent être particulièrement bien adaptés à notre problème puisque :

- Les objets mis en jeu par ces processus peuvent facilement être nos objets d'intérêt, c'est à dire des arbres.

- L'approche est sur-pixélique : on se détache totalement de l'image pour se rapprocher du modèle de réalité physique.

- Les méthodes mises en jeu sont stochastiques, ce qui permet d'utiliser une panoplie d'outils classiques dans les problèmes inverses.

Les deuxième chapitre présentera le cadre mathématique de cette approche probabiliste, tandis que le troisième décrira le modèle que nous avons utilisé pour notre application à la détection des houppiers.

Enfin, nos premiers résultats, obtenus sur des images de forêts fournies par l'IFN, seront exposés, discutés et comparés avec les méthodes classiques dans une dernière partie.

CHAPITRE

1 Etat de l'art en matière de détection de houppiers

Différentes méthodes de détection de houppiers ont fait leur apparition ces dernières années. Deux grandes écoles en sont principalement à l'origine, toutes deux appartenant à des pays où il existe un réel besoin dans le domaine de la gestion des forêts : le Canada, et les pays scandinaves.

1.1. Introduction

Même si aucune méthode n'est totalement satisfaisante à ce jour en ce qui concerne la détection de houppiers, un certain nombre de travaux fournissent un cadre préliminaire.

Un premier type de méthodes consiste à détecter les maxima locaux de l'image. Un algorithme de suivi de vallées permet alors de déterminer les zones d'influence de ces maxima, chaque zone étant assimilée à un houppier. Ce genre de technique délimite donc les contours des arbres.

Pour des vignes, ou d'autres types de plantations présentant un alignement régulier, le comptage est aussi réalisable en travaillant sur la transformée de Fourier de l'image. Ce type d'approche est développé au Cemagref.

D'autres méthodes, introduites par K. Dralle et M. Rudemo, obtiennent de bons résultats pour les arbres situés proches du point Nadir (projeté orthogonal de la caméra sur le sol), en estimant les positions de ces arbres à l'aide d'un lissage de l'image par convolution avec un masque Gaussien. Ces mêmes chercheurs, aidés par M. Larsen, ont ensuite développé un modèle optique destiné à détecter les arbres éloignés du point Nadir. Il consiste à faire coïncider au mieux sur l'image des modèles optiques d'arbres ("template matching") : on peut alors déduire les positions des sommets des arbres, en trouvant les maxima de la fonction de corrélation ainsi calculée.

On peut distinguer enfin les approches fondées sur une analyse de la texture. Dans les milieux denses, ces approches ont tendance à fusionner les houppier proches les uns des autres. En outre, la précision géométrique de la délimitation des houppiers reste faible.

Nous allons présenter dans cette partie deux de ces techniques de détection, qui nous ont paru représentatives des deux grands familles d'approches existantes : l'approche contour avec le suivi de vallées, ainsi que l'approche objet, avec le "template matching".

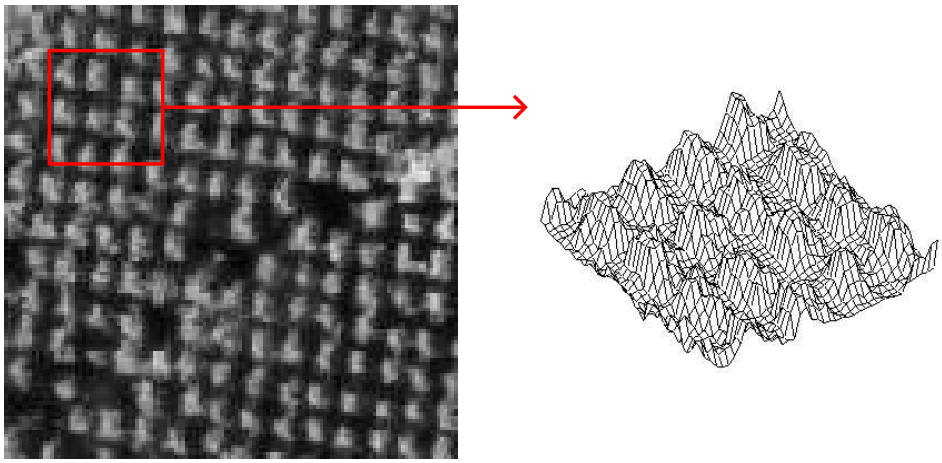

FIG. 1.1 – *A gauche : image de données ©IFN. A droite : extrait de sa carte topographique 3D, la troisième dimension étant les niveaux de gris.*

1.2. L'approche contour

Dans cette partie, nous présentons une méthode de délimitation automatique des couronnes d'arbres, proposée à l'origine par F.A. Gougeon dans [5]. Largement développée au Pacific Forestry Centre (Victoria, British Columbia, Canada), celle-ci fait partie d'un programme plus large de délimitation et de classification par espèces d'arbres. Aussi, nous retrouvons dans d'autres publications du même laboratoire des éléments sur ce sujet pouvant intéresser le lecteur. Celui-ci se reportera aux travaux de F.A. Gougeon et D.G. Leckie ([6], [7], [8], [9], [10], [18], [19], [20]), mais aussi à la thèse de P.S. Petersen [26], réalisée à Copenhague, au sein de la Royal Veterinary and Agricultural University.

1.2.1 Le principe

La base de cette approche réside dans l'analogie qui existe avec la topographie : l'idée générale est ainsi de se représenter l'image en niveaux de gris en trois dimensions, la troisième étant justement la valeur du niveau de gris en chaque pixel. L'image apparaît alors comme un paysage, où les arbres, plus lumineux et donc avec des niveaux de gris élevés, sont les ***montagnes***, et le reste de l'image, les ***vallées***. Une des étapes de l'algorithme consistera à propager ces vallées afin de délimiter les contours des montagnes. Il est à noter que, lors d'une détection manuelle des arbres, l'oeil humain réagit d'une façon similaire à celle utilisée par l'algorithme de suivi des vallées : ce sont en effet les fines bandes d'ombre entre les arbres qui lui permettent de les détecter.

L'algorithme proposé par F.A. Gougeon consiste, dans un premier temps, à isoler les couronnes entre elles et à les séparer de la végétation environnante, grâce à un programme de suivi de vallées (***valley-following algorithm***). Puis, dans un second temps, il s'intéresse aux contours de ces couronnes, obtenus en appliquant un certain nombre de règles prédéfinies (***rule-based system***).

Nous verrons par la suite qu'un prétraitement de l'image est indispensable pour obtenir de bons résultats.

1.2.2 Les images de données

Les images sur lesquelles F.A. Gougeon et D.G. Leckie ont effectué leurs tests étaient des images MEIS (Multispectral Electro-optical Imaging Sensor) [4]. Celles-ci font une meilleure utilisation du spectre des couleurs par rapport aux photographies aériennes, acquiérant jusqu'à 8 bandes

spectrales, et sont directement reliées à des coordonnées géographiques, facilitant par là-même l'interprétation des experts. Leur résolution peut être moyenne (1 à 10 mètres par pixel) ou haute (0.1 à 1 mètre par pixel) selon les besoins.

Cependant, une seule image en niveaux de gris est nécessaire pour l'algorithme de suivi de vallées. Les résolutions des images de tests furent 31 cm/pixel et 70 cm/pixel principalement, mais d'autres résolutions moins fines furent aussi étudiées afin de comparer les résultats obtenus.

Enfin, le choix du type de peuplement d'arbres est problématique, certains étant plus facilement détectables que d'autres. Les premiers résultats de F.A. Gougeon sont obtenus sur des peuplements de conifères.

1.2.3 Etapes de prétraitement

L'image de données, que nous noterons $\mathcal{I}$ par la suite, doit subir un prétraitement avant d'être injectée dans l'algorithme de suivi de vallées. Il convient tout d'abord de choisir une bande spectrale de l'image aérienne, avant de la convertir en image de niveaux de gris. Nous rappelons qu'une hypothèse de départ stipule que les arbres doivent être plus lumineux, en terme de niveaux de gris, que le reste de l'image. C'est pourquoi, et cela se généralise dans la plupart des images forestières, la bande spectrale élue est le proche infra-rouge. Cette bande est en effet sensible aux variations d'illumination et obtient de bonnes réponses de la part des matières végétales chlorophyliennes. Les arbres sans feuilles répondant moins bien, il faut adapter la bande de fréquence lorsque l'on souhaite les étudier, et choisir du visible de préférence, ce qui ne sera pas notre cas. Remarquons qu'il est aussi envisageable de combiner les informations comprises dans plusieurs bandes de fréquences.

Après l'obtention de notre image sur la bande de fréquence désirée, celle-ci est lissée afin d'y ôter le bruit à la fois dû à la discrétisation et au capteur. Ce lissage nous permet de faire disparaître de nombreuses erreurs de mesure, qui peuvent se révéler très génantes notamment si elles sont situées dans les vallées, car elles empêcheraient celles-ci de se rejoindre (voir plus bas le principe du suivi de vallées). Un filtre moyen de largeur $s > 0$ convient pour lisser $\mathcal{I}$. Les coefficients $(f_{i,j})$ de ce filtre, dans une matrice de taille $(2s+1) \times (2s+1)$ sont :

$$f_{i,j} = \frac{1}{(2s+1)^2}. \tag{1.1}$$

Enfin, un dernier traitement consiste à isoler la partie qui nous intéresse dans $\mathcal{I}$, à savoir la forêt, ceci pouvant s'effectuer manuellement par l'utilisateur, ou automatiquement par un processus de classification adéquat. L'objectif de cette dernière étape du prétraitement est de ne garder que les parties de l'image dans lesquelles les arbres sont séparés par une fine bande d'ombre, car ce sont ceux-la que l'on pourra délimiter. On peut par exemple appliquer un masque sur le reste de l'image.

1.2.4 Principe du suivi de vallées

Tout d'abord, nous commençons par chercher dans l'image $\mathcal{I}$ des pixels qui n'appartiennent pas aux couronnes des arbres, faisant partie de zones très ombragées (les vallées). Ces pixels sont supposés avoir un niveau de gris inférieur à un seuil θ, que l'on peut trouver en regardant l'histogramme de l'image. Tous les pixels concernés par ce seuillage sont marqués dans une image binaire $\mathcal{J}$ comme appartenant à la classe des vallées, l'autre classe étant bien entendu celle des arbres. Là encore, le choix du seuillage détermine les résultats que l'on obtiendra. Cependant il n'existe pas de méthode miracle pour trouver le θ optimal, il faut en tester sur les images différentes valeurs.

L'idéal serait en fait d'employer à la place de ce seuillage uniforme un seuillage adaptatif où θ serait dépendant de la position dans l'image. En effet, selon l'inclinaison du terrain et la position du soleil par rapport à la caméra, certaines zones risquent d'être mieux éclairées que d'autres et le seuillage uniforme en sera d'autant moins efficace. Les auteurs mentionnent ce fait, mais se limitent

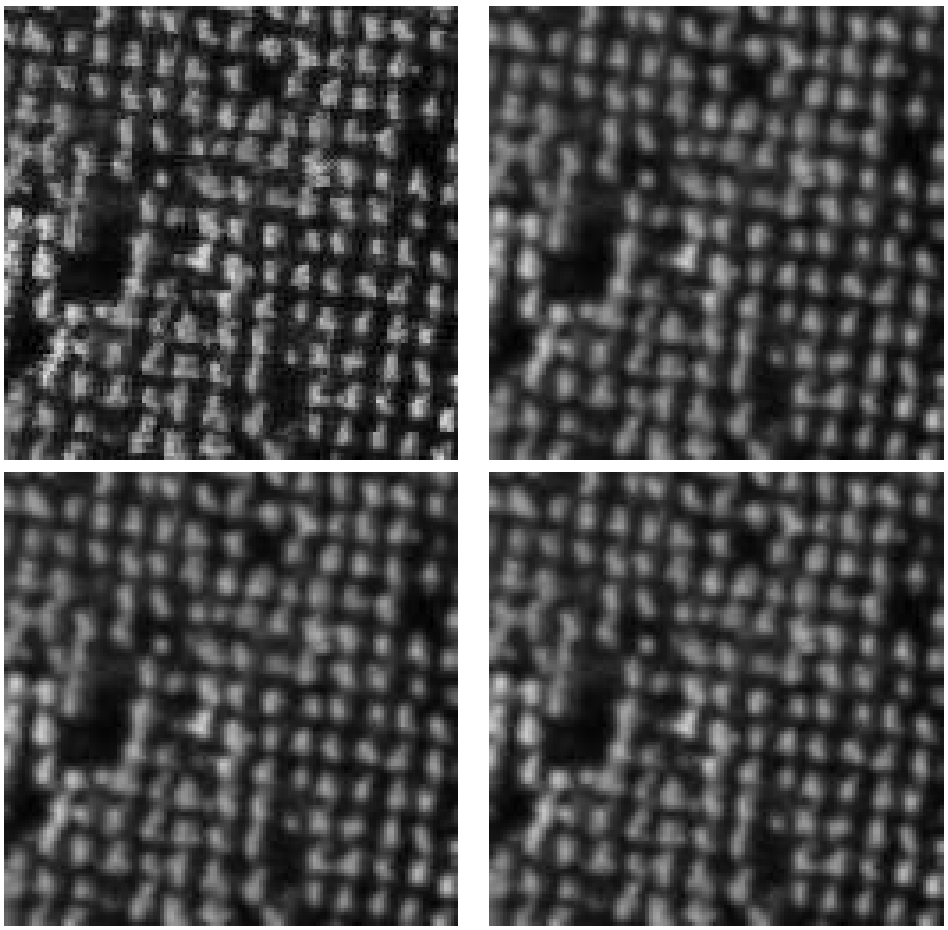

FIG. 1.2 – *Lissage de l'image. En haut à gauche : $\mathcal{I}$ ©IFN. En haut à droite : filtrage avec une fenêtre moyenne de largeur 1. En bas à gauche : filtrage avec une fenêtre moyenne de largeur 3. En bas à droite : filtrage avec une fenêtre moyenne de largeur 5.*

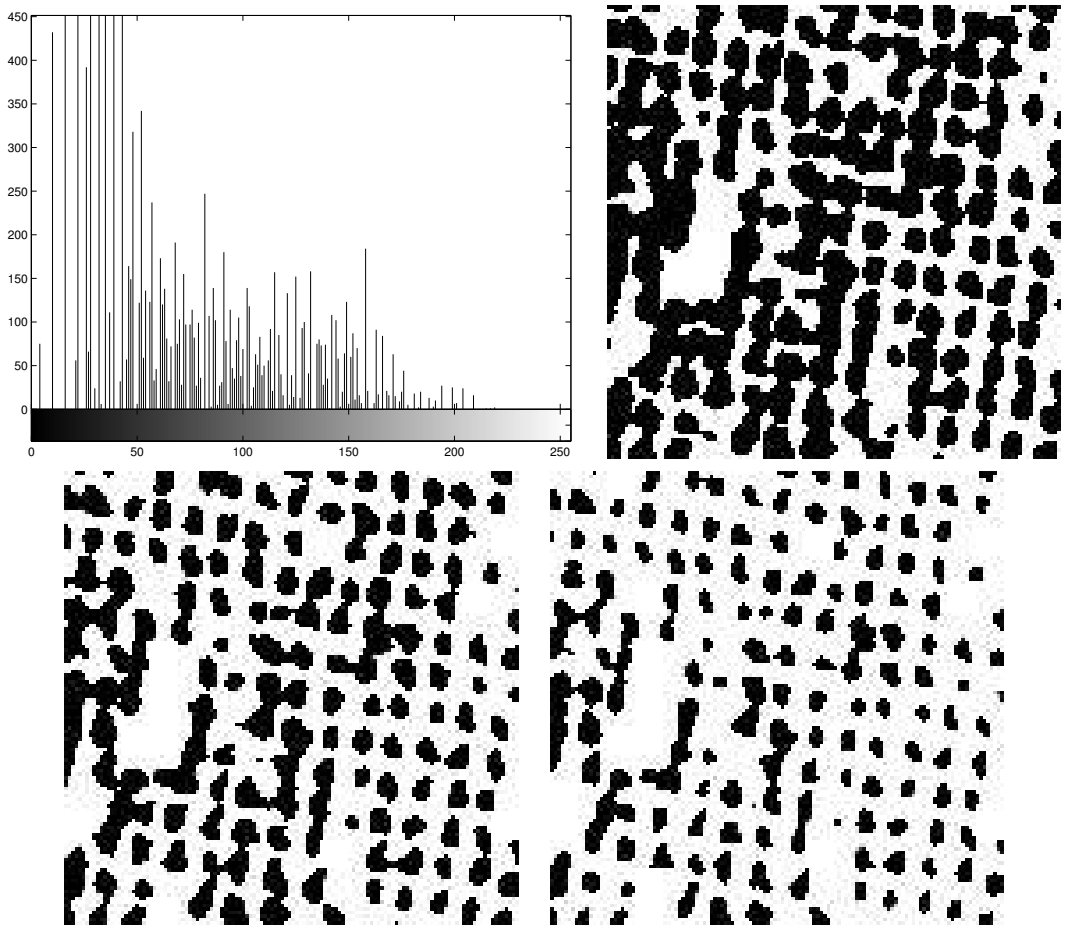

FIG. 1.3 – *Seuillage de l'image filtrée (en blanc les vallées, en noir les montagnes) avec une fenêtre moyenne de largeur 5. En haut à gauche : histogramme de $\mathcal{I}$. En haut à droite : $\mathcal{J}$ obtenue avec $\theta = 50$. En bas à gauche : $\mathcal{J}$ obtenue avec $\theta = 70$. En bas à droite : $\mathcal{J}$ obtenue avec $\theta = 90$.*

dans une première approche à un seuillage uniforme.

On a donc simplement :

$$\mathcal{I}_{i,j} < \theta \Rightarrow \mathcal{J}_{i,j} = 1. \tag{1.2}$$

Puis, un second passage sur les pixels de l'image qui n'ont pas été marqués recherche les minima locaux, en comparant la valeur en niveau de gris de chaque pixel avec celles de ses 8 voisins (ie 8-connexité). Là encore, les pixels vérifiant ce test sont marqués dans l'image $\mathcal{J}$ comme étant des vallées :

$$si\, \mathcal{J}_{i,j} \neq 1 : min\{\mathcal{I}_{i+\alpha,j+\beta}\} = \mathcal{I}_{i,j}, avec\, \alpha = \{-1,0,1\}\, et\, \beta = \{-1,0,1\} \Rightarrow \mathcal{J}_{i,j} = 1. \tag{1.3}$$

La prochaine étape consiste à élargir ces vallées. Nous parcourons l'image $\mathcal{J}$ dans ses 4 directions successivement (suivant ses lignes et ses colonnes) et, dès que l'on rencontre un pixel de $\mathcal{J}$ qui se situe dans une vallée, on explore certains de ses voisins (suivant le sens du parcours) dans l'image $\mathcal{I}$ afin d'y détecter des vallées de largeur 3, 2, ou 1. Nous allons maintenant décrire plus précisément cette étape, en prenant par exemple le cas d'un parcours des pixels de l'image de haut en bas et de gauche à droite (schéma de gauche dans Fig. (1.4)).

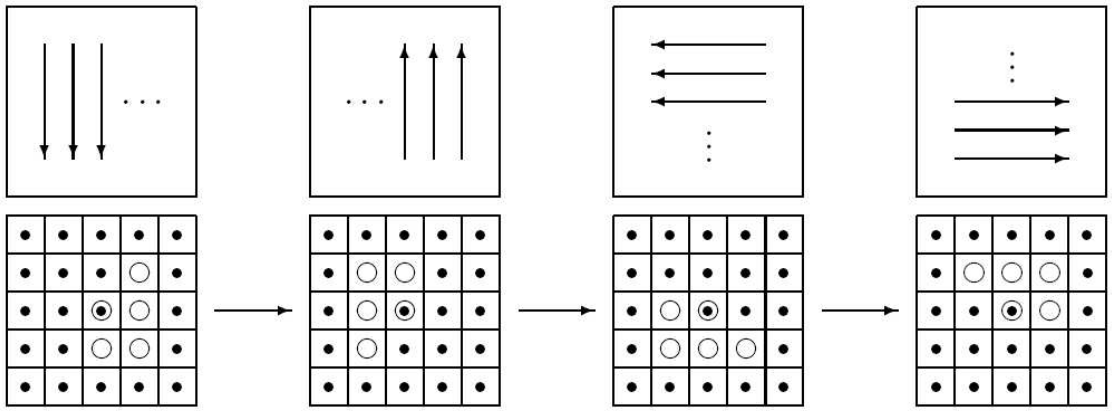

FIG. 1.4 – *Les 4 sens de parcours de l'image. Le pixel sur lequel on s'arrête est le point encerclé, ses voisins sont les 4 cercles, et les pixels que l'on sera amené à étudier pour détecter de nouvelles vallées sont les points.*

Nous nous arrêtons donc sur un pixel (i,j) tel que $\mathcal{J}_{i,j} = 1$, supposé non près du bord pour que les pixels représentés dans Fig. (1.4) existent dans les images $\mathcal{I}$ et $\mathcal{J}$. Ses 4 voisins qui nous intéressent sont dans ce cas les pixels $(i-1,j+1)$,$(i,j+1)$,$(i+1,j+1)$ et $(i+1,j)$. En premier lieu, il faut regarder si les 3 pixels de la colonne $j+1$ constituent une vallée de largeur 3, ce qui est vérifié si

$$\mathcal{I}_{i+\alpha,j+1} < \mathcal{I}_{i+\beta,j+1}, avec\, \alpha = \{-1,0,1\}\, et\, \beta = \{-2,2\}. \tag{1.4}$$

Dans ce cas ces 3 pixels sont marqués dans $\mathcal{J}$ comme appartenant à la classe des vallées. Sinon, nous cherchons des vallées de largeur 2 parmi les couples de pixels suivants : $\{(i-1,j+1),(i,j+1)\}$, $\{(i,j+1),(i+1,j+1)\}$ et $\{(i+1,j),(i+1,j+1)\}$. Le principe est le même que pour les vallées de largeur 3, il suffit que ces pixels aient un niveau de gris moins élevé que les 2 pixels situés à leurs extrémités pour qu'ils soient acceptés. Voici un exemple des inégalités à vérifier pour le couple $\{(i-1,j+1),(i,j+1)\}$:

$$\mathcal{I}_{i+\alpha,j+1} < \mathcal{I}_{i+\beta,j+1}, avec\, \alpha = \{-1,0\}\, et\, \beta = \{-2,1\}. \tag{1.5}$$

Enfin, nous finissons par une recherche des vallées de largeur 1. Chacun des 4 voisins est étudié de façon similaire : si par exemple

$$\mathcal{I}_{i-1,j+1} < \mathcal{I}_{i+\beta,j+1}, avec\, \beta = \{-2,0\}, \tag{1.6}$$

alors le pixel $(i-1,j+1)$ est marqué dans $\mathcal{J}$, les mêmes tests s'appliquant aux trois autres.

Nous répétons ces manipulations pour chacun des 4 sens de parcours de l'image (formant un cycle) jusqu'à ce qu'aucun nouveau pixel de vallées ne soit découvert au cours d'un cycle entier. La première partie de l'algorithme est alors achevée, mais il est nécessaire de délimiter plus précisément les couronnes qui pour la plupart sont mal formées, les vallées n'étant pas assez connectées entre elles : ce sera le rôle du "rule-based system".

1.2.5 Règles d'obtention des contours : le rule-based system

Les couronnes obtenues précédemment étant incomplètes, et regroupant parfois même plusieurs arbres, un traitement après le suivi de vallées est indispensable. Au cours de cette seconde partie

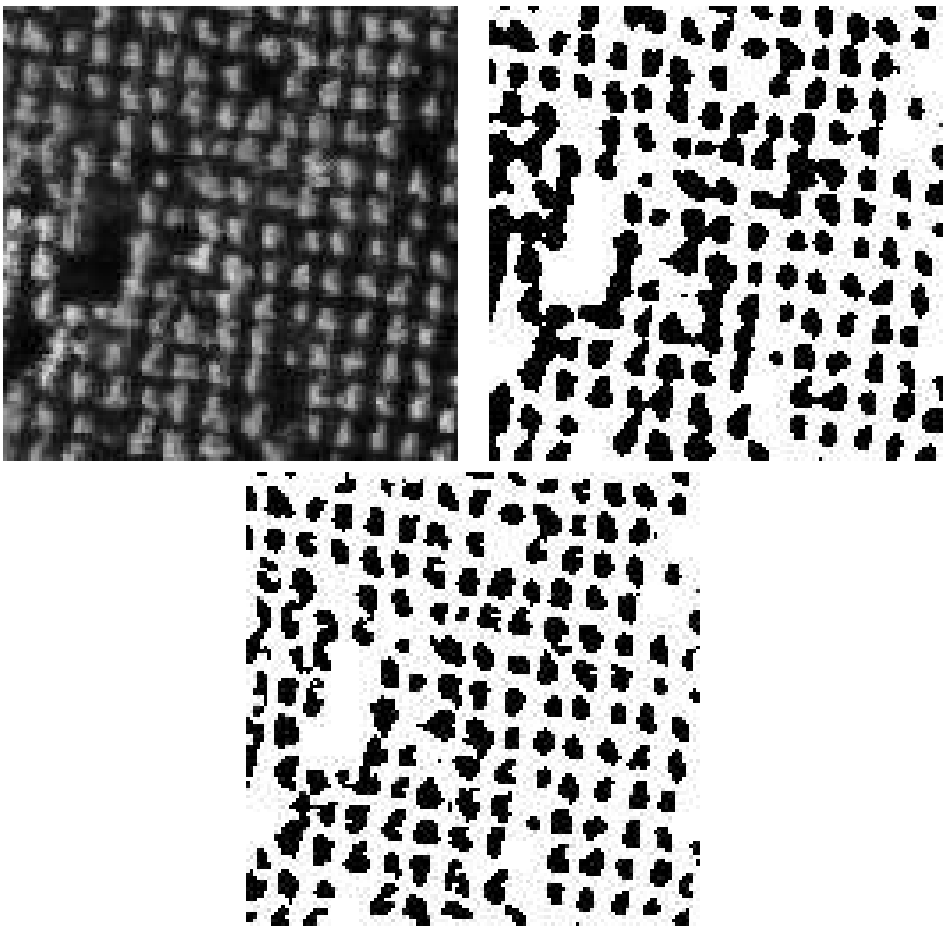

FIG. 1.5 – *Résultat du suivi de vallées. En haut à gauche : $\mathcal{I}$ ©IFN. En haut à droite : $\mathcal{J}$ après filtrage et seuillage, $s = 3$ et $\theta = 70$. En bas, $\mathcal{J}$ après les cycles du suivi de vallées.*

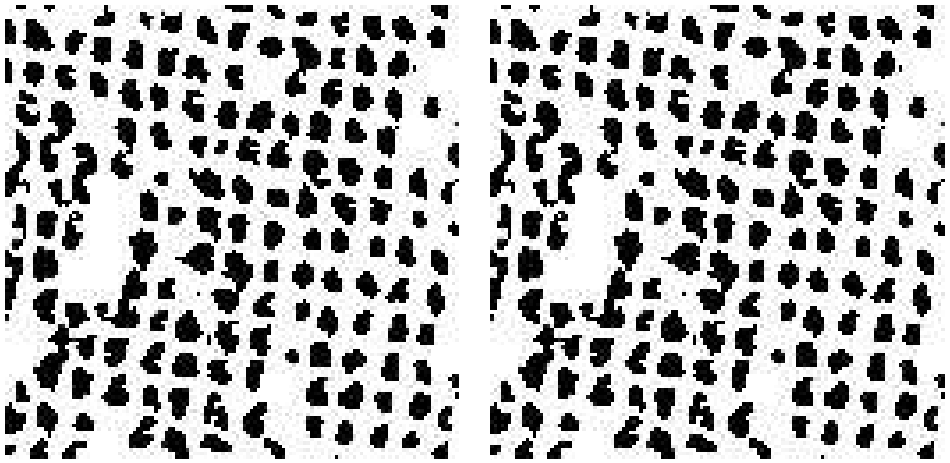

FIG. 1.6 – *Résultat du suivi de vallées avec les paramètres* $s = 3$ *et* $\theta = 70$. *A gauche, avant la fermeture. A droite, après la fermeture.*

de l'algorithme, les contours des couronnes d'arbres sont extraits l'un après l'autre. Auparavant, on applique une fermeture à l'image (dilatation puis érosion) à l'aide d'un élément structurant de taille 3×3. Cette fermeture a pour effet de faire disparaître tous les éléments des couronnes de dimension inférieure à 3 pixels, comme le montrent les images de la Fig. (1.6).

Tout d'abord, nous plaçons l'image $\mathcal{J}$ dans un cadre plus grand, en lui rajoutant 2 lignes et 2 colonnes à ses extrémités. Nous aurons besoin de ce cadre pour ne pas sortir de l'image dans les prochaines étapes. Nous parcourons alors $\mathcal{J}$ dans une des 4 directions possibles, par exemple de haut en bas et de gauche à droite, à la recherche de pixels (i,j) non marqués tels qu'aucun de leurs voisins ne soit marqué :

$$\sum_{r=-1}^{1} \sum_{s=-1}^{1} \mathcal{J}_{i+r,j+s} = 0. \tag{1.7}$$

Puis, à partir d'un pixel (i,j) vérifiant (1.7), nous nous déplaçons vers la gauche, jusqu'à trouver un pixel de $\mathcal{J}$ appartenant à une vallée. Ce pixel appartiendra au contour de l'arbre sur lequel nous nous sommes arrêtés. Il sera le premier maillon d'une chaîne qui va contourner l'arbre.

A partir de là, on applique des règles, au nombre de 5, décrites par F.A. Gougeon dans [5]. Chacune de ces 5 règles contient des critères sur des pixels de l'image binaire, situés dans une direction privilégiée selon le numéro de la règle. Dans le cas où l'un de ces critères est vérifié, le ou les prochains pixels du contour sont déterminés : ils sont par construction reliés avec celui de départ. Puis on recommence cette recherche à partir du dernier maillon de la chaîne.

Ces règles sont appliquées dans l'ordre, de la première à la cinquième, et proposent donc des directions de recherche particulières, s'éloignant progressivement de celle donnée par les deux derniers pixels du contour (de 0 à 360 degrés dans le sens des aiguilles d'une montre par rapport à cette direction). Par exemple, la règle 1 propose des pixels dans deux directions, 0 et 45 degrés, par rapport à l'ancienne direction. On obtiendrait un contour convexe si cette règle était toujours vérifiée, les contours concaves étant quant à eux autorisés par les autres.

Une fois que le contour se referme sur lui-même, au niveau du pixel de départ ou de tout autre pixel, nous contrôlons sa longueur pour éviter que ne se produisent des erreurs : il ne doit être ni trop petit ni trop grand. De proche en proche, nous parvenons ainsi à délimiter chaque couronne.

Par contre, si il ne se referme pas et que sa longueur excède une longueur donnée, on stoppe la

propagation du contour, et on cherche à nouveau un pixel de départ.

F.A. Gougeon propose de répéter ce processus 3 fois afin de délimiter les couronnes ne pouvant être isolées qu'une fois les contours adjacents établis. En effet, on remplit le contenu de ces contours dans l'image binaire $\mathcal{J}$, ce qui apporte de nouvelles informations. Il faut attendre qu'un parcours complet de l'image sans aucun nouveau contour détecté ne soit effectué, avant de finir l'algorithme.

1.2.6 Résultats et interprétations

Un premier résultat, obtenu par F.A. Gougeon sur des images de plantations coniférienne (épicéas rouge, blanc et de Norvège, pin rouge) d'une résolution de 31 cm/pixel, avance un dénombrement des couronnes se situant à 7.7% près de celui réalisé par des experts au sol. Mieux encore, le dénombrement fait par photo-interprétation, par des experts, s'est révélé être à seulement 18.1% près. Une grande partie des couronnes d'arbres a donc été correctement détectée, très exactement 81% de ces contours sont semblables aux contours manuels dessinés par des experts sur le terrain.

Les différentes espèces d'arbres parmi les conifères ne donnent pourtant pas le même résultat : F.A. Gougeon a ainsi dû ne pas tenir compte d'une des parcelles coniférienne : les épicéas de Norvège. Ayant un diamètre beaucoup plus gros que les autres, ceux-ci sont coupés en de nombreuses couronnes au cours de l'algorithme (138 détectés au lieu de 39 sur le terrain). Mais des tests supplémentaires effectués sur des images à résolution plus faible (70 cm/pixel) font disparaître ces inconvénients, de même qu'un lissage plus conséquent. En outre, certaines parcelles de pins, plus éclairées que les autres, ont posé de grandes difficultés tant à l'algorithme qu'aux experts humains. Dans ce cas, le programme ne sépare pas assez les arbres entre eux, car les vallées tranchent moins par rapport aux montagnes à cause de cette illumination. Une analyse plus poussée des erreurs révèle d'ailleurs que la plus commune fut l'omission d'une des couronnes, dû à l'amalgame avec une autre. Ceci arriva pour 15.3% des couronnes.

D.G. Leckie, dans [18], a quant à lui travaillé sur des images de résolution 36 cm/pixel représentant des parcelles de "Tsuga heterophylla" (western hemlock), conifères typiques de l'Amérique du Nord, et de cèdres, sur un site qui regroupait des peuplements d'arbres d'âges différents. Il commence par décrire des degrés de précision de recouvrement des arbres détectés par l'algorithme sur les arbres existants, qui l'aident à mieux quantifier l'exactitude des résultats qu'il obtient. Il dégage une interprétation intéressante de ceux-ci en fonction de la densité des troncs d'arbres par hectare et de leur âge. Ainsi, l'algorithme de détection des contours fonctionne mieux sur des vieux arbres, plus gros, pour lesquels le recouvrement est pour 60% d'entre eux parfait (selon ses critères d'évaluation, voir [18] pour plus de détails), alors qu'il est rare de trouver des arbres jeunes bien délimités. En ce qui concerne la densité enfin, il remarque que pour des densités supérieures à 900 arbres/ha, le nombre d'omission augmente de manière conséquente, et la sous-etimation du nombre total de couronnes par là-même.

En conclusion de ces travaux, les conditions pour lesquelles cet algorithme fonctionne très bien sont :

- un soleil bas, car une trop grande luminosité sur les parcelles diminue le contraste entre les vallées et les montagnes.

- des peuplements d'arbres appropriés, à savoir des conifères, dont la forme conique est favorable.

- des densités inférieures à 900 arbres/ha, au-delà desquelles le nombre d'omissions augmente considérablement.

- une résolution de l'image adaptée à la taille des arbres que l'on doit détecter, pour éviter le morcellement des grosses couronnes.

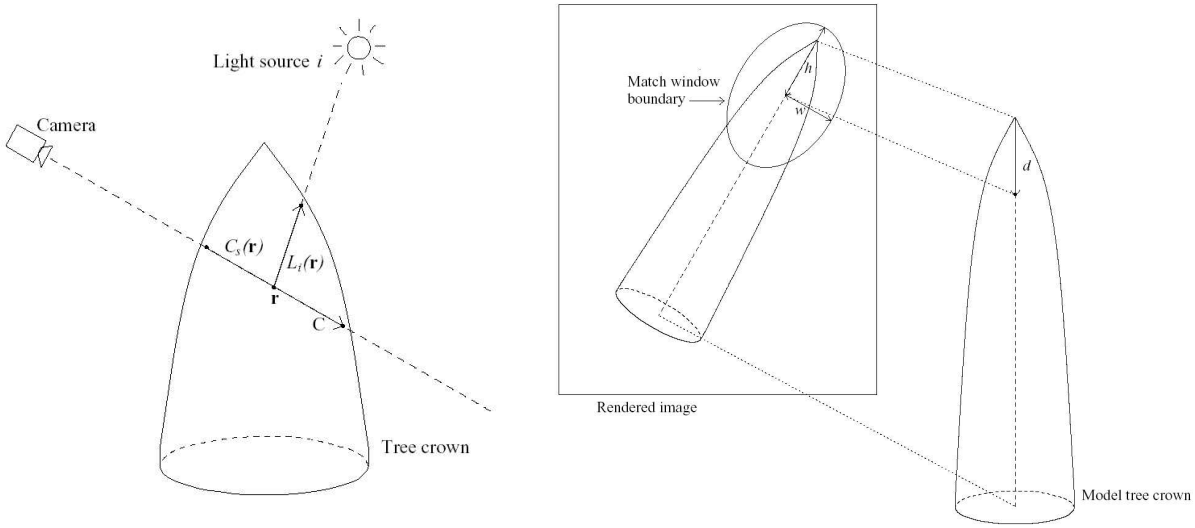

FIG. 1.7 – *Modèle optique de Larsen, obtention du patron (fenêtre elliptique) par un lancer de rayons.*

1.3. L'approche objets

M. Larsen, dans [15], [16] et [17], propose une autre approche au problème de détection des houppiers. Des patrons (template), obtenus grâce à un modèle optique des arbres, lui permettent de trouver les positions des arbres sur des images aériennes. Cette méthode a pour nom le ***template matching***, et est principalement développée au sein du Danish Agricultural and Veterinary Research Council.

1.3.1 Le principe

▸ Définition d'un patron (template)

Toute la difficulté de cette méthode consiste à définir le patron que l'on essaiera d'assortir dans l'image. Ce patron est en fait une image représentant un unique houppier obtenu grâce à un modèle optique de la scène. Il est calculé à partir d'un modèle géométrique de l'arbre (ellipsoïde), et d'un lancer de rayon initialisé grâce à des informations que donne l'utilisateur au programme, comme l'inclinaison du soleil à l'instant de la photographie, et la position de la caméra.

Ce patron est alors coupé par une fenêtre elliptique, puisque les arbres ne sont pas visibles entièrement sur une image aérienne, à cause de leurs voisins. Les paramètres de forme de cette fenêtre peuvent être renseignés par des connaissances a priori sur la hauteur des arbres, ou leur espacement. La Fig. (1.7) illustre ces propos, quant à la Fig. (1.8), elle donne un exemple de patron que M. Larsen a utilisé pour ses images.

▸ Positionnement des sommets des arbres

On transpose alors cette fenêtre dans l'image aérienne, calculant en chaque pixel sa corrélation avec l'image. A partir des maxima locaux dans l'image de corrélation résultante, on peut déduire les positions des arbres.

La Fig. (1.9) nous donne un résultat obtenu par M. Larsen pour une parcelle d'épicéas de Norvège.

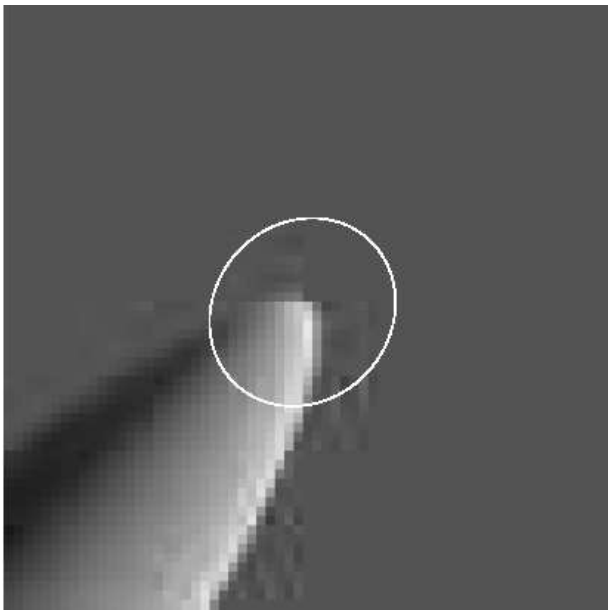

FIG. 1.8 – *Exemple de patron obtenu par M. Larsen.*

Les sommets des arbres sont calculés à partir du patron de la Fig. (1.8).

▸ Résultats

Avec des paramètres optimaux pour la fenêtre elliptique, M. Larsen obtient une reconnaissance de près de 90% des sommets des arbres pour les parcelles les plus favorables. Ces résultats peuvent être altérés dans le cas où aucune connaissance a priori sur les arbres du peuplement forestier n'est disponible.

M. Larsen propose alors de faire varier la taille de la fenêtre elliptique, et de calculer plusieurs images de corrélation. Les positions des arbres seront les pixels de l'image qui ont le plus souvent été cités par les différentes fenêtres. Pour cela, on attribue autant de votes aux pixels d'une image de votes, de même taille que l'image aérienne, que de fenêtres qui y détectent un arbre. Les maxima locaux de l'image de votes ainsi obtenue, une fois lissée, représentent les positions des houppiers. Cette technique est appelée le "template voting".

1.3.2 Récapitulatif : l'algorithme de M. Larsen

L'algorithme de M. Larsen comporte donc 3 étapes :

1. modélisation du patron grâce au lancer de rayons. Idéalement, ce patron doit être calculé en chaque pixel de l'image, situé plus ou moins loin du point Nadir. Dans la pratique, un même patron est utilisé pour des zones de l'image regroupant plusieurs pixels.

2. pour chaque pixel de l'image aérienne, on calcule la corrélation de l'image avec le patron.

3. recherche des maxima locaux de l'image de corrélation résultante. Si l'on connaît a priori le nombre d'arbres à détecter, il suffit alors de les prendre dans l'ordre décroissant, en ajoutant une règle qui interdit que deux positions d'arbres ne soient trop proches. Sinon, il est conseillé d'utiliser le principe des votes, en reproduisant la démarche précédente avec plusieurs fenêtres.

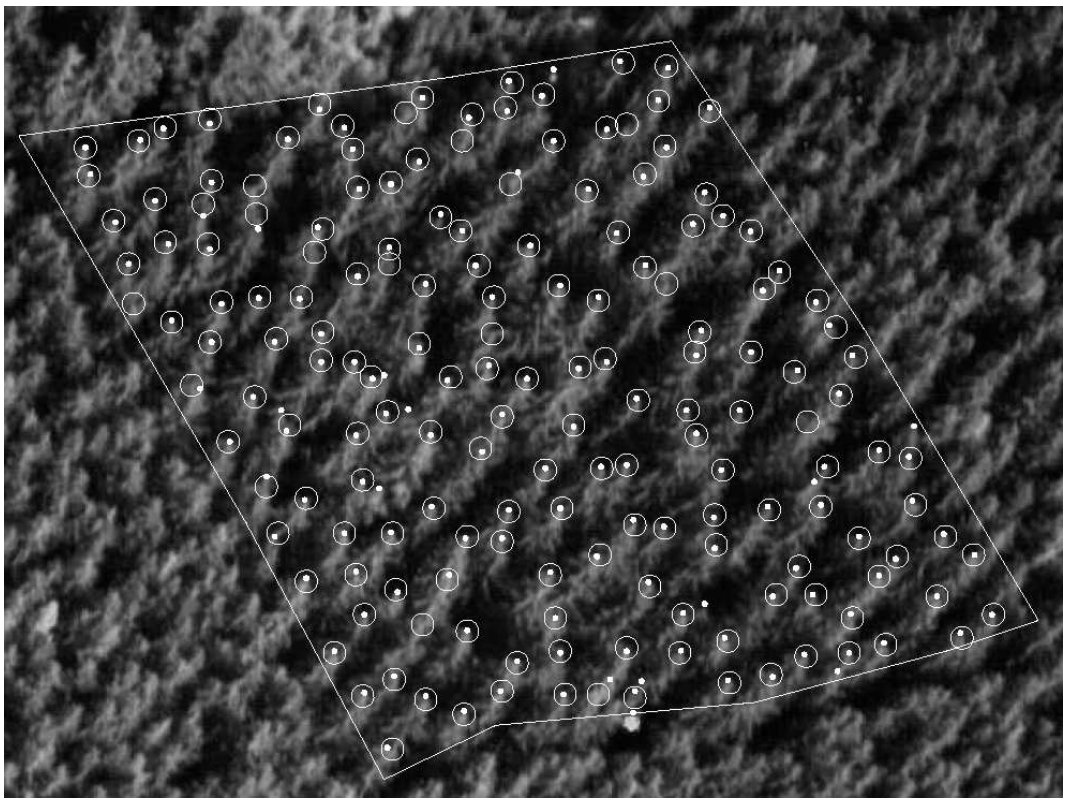

FIG. 1.9 – *Résultats du template matching sur une parcelle d'épicéas de Norvège.*

CHAPITRE

2 Processus objet : définitions et simulation

Dans de nombreux domaines de la recherche et de l'industrie, la résolution de problèmes nécessite l'utilisation de modèles géométriques, et souvent une analyse statistique pour les simuler. Ces problèmes relèvent de la ***Géométrie Stochastique***.

Les processus objet font partie intégrante de la géométrie stochastique : ce sont des modèles probabilistes permettant d'étudier des structures géométrique complexes, à partir d'éléments de bases simples.

Leurs applications dans le domaine du traitement d'images sont nombreuses. Comme premiers exemples, le lecteur pourra se reporter à [14] et [33], où l'on extrait à partir d'images aériennes les segments de droite constituant le réseau linéique routier, à [2] et [24], où l'on extrait les caricatures rectangulaires des bâtiments, mais aussi à [13], qui applique ces méthodes à la segmentation d'images.

L'objectif de ce chapitre est de présenter les points qui nous ont paru essentiels autour de la théorie des processus objet. Le cadre mathématique donnant naissance à ces processus sera dans un premier temps exposé, avec des exemples à la clé. Puis, dans un deuxième temps, nous aborderons la question de la simulation des processus ponctuels.

2.1. Les processus objet : définitions et exemples

Les éléments de théorie que nous exposons ici ne comprennent que le nécessaire à l'élaboration de notre modèle, le lecteur pouvant trouver des compléments en consultant [34] et [35].

2.1.1 Première approche des processus ponctuels

▸ L'espace objet U

Chaque objet du processus que l'on souhaite simuler appartient à un même espace que nous appelons l'espace objet, noté U. Dans la pratique, et surtout en traitement d'images, ce sera souvent

un sous-ensemble compact de $\mathbb{R}^n$.

Cet espace U doit contenir toute l'information sur les objets, à savoir leur position, leurs paramètres de forme, puisque ce sont des objets géométriques, et tout autre paramètre servant à les déterminer de façon unique.

Nous munissons U d'une tribu $\mathcal{B}(U)$ ainsi que d'une mesure $\lambda(.)$ telle que $0 < \lambda(U) < \infty$.

▸ L'espace des configurations $\mathbf{\Psi}$

Définition 2.1

> Une ***configuration*** $\mathbf{x}$ est un ensemble dénombrable, non ordonné, de points de U.
>
> $$\mathbf{x} = \{x_1, \ldots, x_n, \ldots\}, x_i \in U \, \forall i.$$

Par exemple, $\{-3, 5.2, \pi, 177\}$ est une configuration de points de $\mathbb{R}$.

L'espace des configurations $\mathbf{\Psi}$ sur U est alors défini comme la réunion de toutes les configurations $\mathbf{x}$.

Cet espace de décompose sous la forme d'une partition $\mathbf{\Psi} = \bigcup_n \mathbf{\Psi}_n$, où $\mathbf{\Psi}_n$ contient l'ensemble des configurations $\mathbf{x}$ composées exactement de n objets non nécessairement distincts deux à deux. On note $\mathbf{\Psi}_0 = \{\emptyset\}$ la configuration vide.

Le lien entre l'espace objet U et l'espace des configurations $\mathbf{\Psi}$ s'obtient par l'intermédiaire de projections de l'espace produit U^n, constitué de tous les n-uplets ordonnés d'éléments non nécessairement distincts deux à deux, sur l'espace des configurations $\mathbf{\Psi}_n$:

$$\omega_n : \begin{array}{c} U^n \to \mathbf{\Psi}_n \\ (x_1, \ldots, x_n) \to \{x_1, \ldots, x_n\}. \end{array} \tag{2.1}$$

A un élément de $\mathbf{\Psi}_n$ correspondent donc $n!$ éléments de U^n. Les fonctions ω_n sont surjectives.

Définition 2.2

> Une configuration $\mathbf{x}$ est dite ***localement finie*** si pour tout borélien $A \subseteq U$, le nombre de points de $\mathbf{x}$ dans A est fini. On note $\mathbf{\Psi}^{lf}$ le sous ensemble de $\mathbf{\Psi}$ formé des configurations localement finies.
> Pour tout n, les configurations $\mathbf{x} \in \mathbf{\Psi_n}$ sont par définition finies.

Définition 2.3

> Une configuration $\mathbf{x}$ est dite ***simple*** si elle ne contient que des points deux à deux distincts. On note $\mathbf{\Psi}_s$ le sous ensemble de $\mathbf{\Psi}$ formé des configurations simples.

Par la suite, nous travaillerons sur des configurations localement finies et simples. Afin d'alléger les notations, nous remplacerons abusivement l'écriture de $\mathbf{\Psi}^{lf} \cap \mathbf{\Psi}_s$ par celle de $\mathbf{\Psi}$.

▸ Définition d'un processus ponctuel

Une fois les configurations d'objets définies, nous pouvons alors aborder la notion de processus ponctuel.

Définition 2.4

> Un ***processus ponctuel*** sur U est une application d'un espace probabilisé $(\Omega,\mathcal{A},\mathbb{P})$ dans $\mathbf{\Psi}^{lf}$, tel que pour tout borélien borné $A \subseteq U$, le nombre $N(A)$ de points dans A soit une variable aléatoire.

Définition 2.5

> Un ***processus ponctuel marqué***, ou ***processus objet***, sur $U = \mathcal{P} \times \mathcal{K}$ est un processus ponctuel sur $U = \mathcal{P} \times \mathcal{K}$ dont les positions sont dans $\mathcal{P}$ et les marques dans $\mathcal{K}$, tel que le processus des points non marqués soit un processus ponctuel bien défini sur $\mathcal{P}$.

D'après [35], pour définir un processus ponctuel il suffit de connaître ses probabilités vides :

Proposition 2.6 *:* **Probabilités vides**

> Un processus ponctuel est entièrement déterminé par les probabilités vides des boréliens bornés : $\{\mathbb{P}(N(A)) = 0 : A \in U\}$.

Cette proposition peut se révéler utile en pratique comme nous le verrons dans le cadre du processus ponctuel de Poisson défini ci-dessous.

▸ Premiers exemples de processus ponctuels

L'exemple le plus simple qui nous vient à l'esprit est celui d'un processus ponctuel ne contenant qu'un point du plan, ξ, uniformément distribué sur un compact $U \in \mathbb{R}^2$. En munissant l'espace objet U de la mesure de Lebesgue $\lambda(.)$ de $\mathbb{R}^2$, ξ est tel que

$$P(\xi \in A) = \frac{\lambda(A)}{\lambda(U)},$$

pour tout borélien $A \in U$.

D'autres exemples méritent d'être abordés.

▷ Processus ponctuel binomial

Nous pouvons généraliser l'exemple précédent à des processus ponctuels composés de n points uniformément distribués sur U. On parle dans ce cas de processus ponctuel binomial. La Fig. (2.1) montre un exemple de simulation d'un tel processus, avec $n = 50$ et $U = [0,100]^2$.

Ce type de processus est appelé binomial puisque pour tout borélien $A \in U$, la variable aléatoire $N(A)$ du nombre de points dans A suit une loi binomiale de paramètres n et $\frac{\lambda(A)}{\lambda(U)}$.

▷ Processus ponctuel de Poisson

Un autre exemple de processus ponctuels, sans doute le plus utilisé en simulation, est celui du processus ponctuel de Poisson de loi d'intensité $\nu(.)$. Dans ce cas, la variable aléatoire $N(A)$ définie

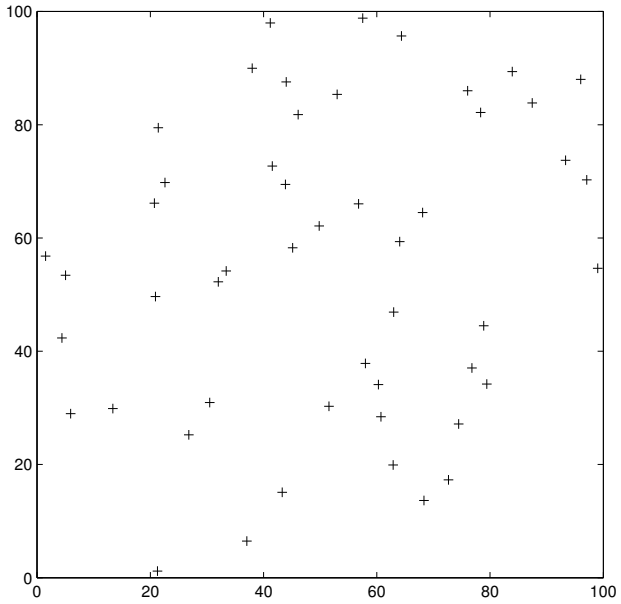

FIG. 2.1 – *Simulation d'un processus binomial de 50 points sur* $[0,100]^2$

plus haut suit une loi de Poisson d'espérance $\nu(A)$ pour tout borélien $A \in U$.

Une façon de les simuler consiste à tout d'abord générer le nombre de points n sur U du processus, selon la distribution

$$\mathbb{P}(n) = e^{-\nu(U)} \frac{\nu(U)^n}{n!}$$

d'espérance $\nu(U)$, puis de choisir ces n points sur U indépendamment suivant la loi $\frac{\nu(.)}{\nu(U)}$. On parle de processus de Poisson homogène lorsque $\nu(.)$ est proportionnelle à la mesure de Lebesgue $\lambda(.)$ sur U.

Proposition 2.7 *:* **Processus de Poisson conditionné = Processus binomial**

> Soit $\mathbf{X}$ un processus de Poisson de loi d'intensité $\nu(.)$. Pour tout borélien borné $A \in U$, conditionnellement à $\{N(A) = n\}$, la restriction du processus $\mathbf{X}$ à A est un processus ponctuel binomial de points ν-uniformément distribués.

♢ **Démonstration :**
Il suffit de montrer que les probabilités vides de ces deux processus sont égales, d'après la proposition (2.1.1), ce qui est fait dans [34] ou [35].

♢

La Fig. (2.2) nous donne un exemple de processus de Poisson homogène, toujours sur $U = [0,100]^2$, de paramètre de proportionnalité $a = 0.005$. Dans ce cas, on vérifie aisément que l'espérance du nombre de points dans U, $\mathbb{E}[N(U)]$, est

$$\mathbb{E}[N(U)] = a.\lambda(U) = 50. \tag{2.2}$$

Enfin, la Fig. (2.3) nous donne une réalisation d'un processus de Poisson accordant 3 fois plus de poids au sous ensemble $U' = [0,50]^2$. Un tel processus peut par exemple être simulé par la loi

$$\nu(A) = a.\left[\lambda\left(A \cap (U/U')\right) + 3\lambda\left(A \cap (U')\right)\right].$$

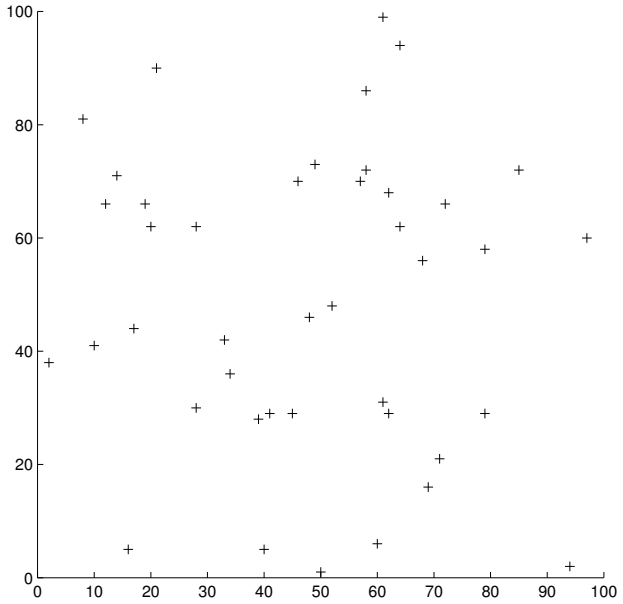

FIG. 2.2 – *Réalisation d'un processus de Poisson homogène, a=0.005.*

Avec le même paramètre $a = 0.005$, l'espérance du nombre de points d'une réalisation d'un tel processus est $\nu(U) = 75$.

▸ Tribu et mesure de l'espace des configurations

Pour tout $n > 0$, munissons l'espace produit U^n de la tribu produit $\mathcal{B}^n$ déduite de $\mathcal{B}(U)$, ainsi que $\mathbf{\Psi}_0$ de la tribu triviale $\{\{\emptyset\},\mathbf{\Psi}\}$. Chaque projection ω_n permet d'obtenir une famille $\omega_n(\mathcal{B}^n)$ de parties de $\mathbf{\Psi}_n$. Nous choisissons alors comme tribu sur $\mathbf{\Psi}$ la tribu notée $\mathcal{B}(\mathbf{\Psi})$ et engendrée par :

$$\omega_0(\mathcal{B}^0), \omega_1(\mathcal{B}^1), \ldots, \omega_n(\mathcal{B}^n), \ldots$$

On considère alors comme mesure de référence la mesure $\mu(.)$ définie pour tout $B \in \mathcal{B}(\mathbf{\Psi})$ par :

$$\mu(B) = e^{-\lambda(U)} \left(\mathbf{1}_{[\emptyset \in B]} + \sum_{n=1}^{\infty} \frac{\mu_n(B)}{n!} \right) \tag{2.3}$$

avec :

$$\mu_n(B) = \int \ldots \int \mathbf{1}_{[\{u_1,\ldots,u_n\} \in B]} \lambda(du_1) \ldots \lambda(du_n). \tag{2.4}$$

Cette mesure est une mesure de probabilité, qui correspond à la distribution d'un processus de Poisson homogène X sur U de mesure d'intensité $\lambda(.)$.

Ainsi, le nombre d'objets $n(X)$ de la configuration suit une loi de Poisson de paramètre $\lambda(U)$. De plus, comme tout processus de Poisson, conditionnellement à l'événement $\{n(X) = n\}$, la loi

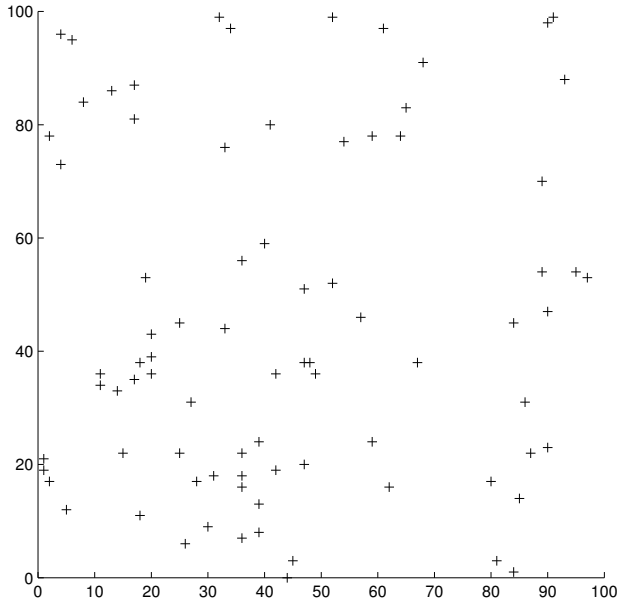

FIG. 2.3 – *Réalisation d'un processus de Poisson non homogène, donnant trois fois plus de poids au coin en bas à droite.*

$\mu(.|n(X)=n)$ suit une loi binomiale à n points sur U de densité $\lambda(.)/\lambda(U)$. Elle est donc la superposition de n points indépendants et uniformément répartis dans U suivant la loi $\lambda(.)/\lambda(U)$.

L'existence de l'espace probabilisé de l'ensemble des configurations, $(\mathbf{\Psi},\mathcal{B}(\mathbf{\Psi}),\mu(.))$, nous permet alors de définir un processus ponctuel par sa densité par rapport à la loi de Poisson de référence sur cet espace.

2.1.2 Définition d'un processus ponctuel par une densité non normalisée

▸ Processus ponctuel défini par une loi par rapport à un processus de Poisson de référence

Sur un espace probabilisé de référence $(\mathbf{\Psi},\mathcal{B}(\mathbf{\Psi}),\mu(.))$, un processus ponctuel peut être défini par sa loi par rapport à la mesure de Poisson de référence.

Définition 2.8

> Un ***processus ponctuel*** est une application mesurable d'un espace probabilisé et à valeurs dans $(\mathbf{\Psi},\mathcal{B}(\mathbf{\Psi}),\mu(.))$. On note $P(.)$ sa loi sur $(\mathbf{\Psi},\mathcal{B}(\mathbf{\Psi}),\mu(.))$.

Par la suite, nous travaillerons sur des processus X dont les lois $P(.)$ sont uniformément continues par rapport à $\mu(.)$, que l'on note $P << \mu$. Le théorème de Radon Nikodym nous donne une condition nécessaire et suffisante pour qu'une telle hypothèse sur nos mesures soit vérifiée, qui est que

$$\forall B \in \mathcal{B}(\mathbf{\Psi}) \,:\, \mu(B)=0 \Longrightarrow P(B)=0. \tag{2.5}$$

La dérivée de Radon Nikodym est alors la densité $f(.)$ de $P(.)$ par rapport à $\mu(.)$, et s'écrit

$$f=\frac{dP}{d\mu}$$

La définition d'un processus objet se réduit donc à la définition d'une application $h(.)$ mesurable positive sur $(\mathbf{\Psi},\mathcal{B}(\mathbf{\Psi}),\mu(.))$, et à la vérification de sa μ-intégrabilité, c'est-à-dire à l'existence d'une constante de normalisation. On a alors :

$$f(\mathbf{x}) = \frac{h(\mathbf{x})}{\int_{\mathbf{\Psi}} h(\mathbf{x})\mu(d\mathbf{x})} \tag{2.6}$$

La mesure $P(.)$ du processus s'écrit alors $\forall B \in \mathcal{B}(\mathbf{\Psi})$:

$$\begin{aligned} P(B) &= \int_B f(\mathbf{x})\mu(d\mathbf{x}) \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \int_{B\cap U^n} f(\mathbf{x})\mu_n(d\mathbf{x}). \end{aligned}$$

▸ Expression générale de la densité non normalisée

Généralement, les densités considérées sont de la forme :

$$h(\mathbf{x}) = \alpha(n(\mathbf{x}))g(\mathbf{x}) \tag{2.7}$$

où $n(\mathbf{x})$ représente le nombre d'objets de la configuration $\mathbf{x}$, et où $\alpha(.)$ et $g(.)$ sont deux applications mesurables positives définies respectivement sur $\mathbb{N}$ et sur $\mathbf{\Psi}$. La constante de normalisation n'est a priori pas connue lors de la définition du modèle.

Le premier terme correspond à une densité de processus considéré par rapport à la distribution de référence $\mu(.)$, et donc contrôle le nombre moyen d'objets, alors que le second terme permet de modéliser les intéractions entre les objets de la configuration. Nous donnerons par la suite des exemples de processus utilisant ce type de densités.

▸ Intensité conditionnelle de Papangelou

A l'exception du processus de Poisson, les processus objet peuvent être regroupés en deux catégories : les processus de type attractif et les processus de type répulsif. La plupart du temps, ce comportement peut être caractérisé de manière locale à l'aide de l'intensité conditionnelle de Papangelou.

Définition 2.9

> Soit $\mathbf{x}$ un processus ponctuel défini par une densité $f(\mathbf{x})$ par rapport à un processus ponctuel de Poisson de mesure d'intensité finie et non-atomique $\mu(.)$. Alors $\mathbf{x}$ a une ***intensité conditionnelle de Papangelou*** définie par :
>
> $$\lambda^*(\mathbf{x},\xi) = \frac{f(\mathbf{x}\cup\{\xi\})}{f(\mathbf{x})}, \forall \xi \in \{U - \bigcup_{i=1}^{n(\mathbf{x})} x_i\}.$$

Soient alors deux configurations $\mathbf{x}_1$ et $\mathbf{x}_2$. Nous interprétons

$$\mathbf{x}_2 \subset \mathbf{x}_1 \Rightarrow \lambda^*(\mathbf{x}_2,.) \geq \lambda^*(\mathbf{x}_1,.) \tag{2.8}$$

comme un cas répulsif, et

$$\mathbf{x}_2 \subset \mathbf{x}_1 \Rightarrow \lambda^*(\mathbf{x}_2,.) \leq \lambda^*(\mathbf{x}_1,.) \tag{2.9}$$

comme un cas attractif.

▸ Critères de stabilité

Pour que la densité non normalisée soit intégrable par rapport à $\mu(.)$, et comme le calcul de la constante de normalisation est très laborieux, des critères de stabilité ont été énoncés. Ces critères servent aussi à assurer la convergence des algorithmes de simulation des processus ponctuels.

Proposition 2.10 *:* **Stabilité au sens de Ruelle**

> Un processus défini par une densité non normalisée $h(.)$ par rapport à $\mu(.)$ est ***stable au sens de Ruelle*** s'il existe un nombre réel M_h tel que :
>
> $$h(\mathbf{x}) \leq M_h^{n(\mathbf{x})}, \forall \mathbf{x} \in \mathbf{\Psi} \tag{2.10}$$

♢ **Démonstration :**
Cette condition est suffisante pour que $h(.)$ soit intégrable, avec comme constante de normalisation :

$$\begin{aligned} \int_{\mathbf{\Psi}} h(\mathbf{x})\mu(d\mathbf{x}) &\leq \int_{\mathbf{\Psi}} M_h^{n(\mathbf{x})}\mu(d\mathbf{x}) \\ &= \sum_{k=0}^{\infty} \frac{M_h^k}{k!}\lambda(U)^k \\ &= e^{M_h \lambda(U)}. \end{aligned}$$

♢

Ce critère est peu contraignant pour la fonction $h(.)$ mais peut poser des problèmes de vérification. Aussi est-il utile d'introduire le second critère, plus contraignant mais plus facilement vérifiable. Cette seconde condition de stabilité est d'ailleurs utilisée pour établir certaines propriétés des schémas de simulation des processus ponctuels par des chaînes de Monte Carlo (voir [3]).

Proposition 2.11 *:* **Stabilité locale**

> Un processus défini par une densité non normalisée $h(.)$ par rapport à $\mu(.)$ est ***localement stable*** s'il existe un nombre réel M_h tel que :
>
> $$h(\mathbf{x} \cup \xi) \leq M_h\, h(\mathbf{x}), \forall \mathbf{x} \in \mathbf{\Psi}, \forall \xi \in U \tag{2.11}$$
>
> c'est à dire si l'intensité conditionnelle de Papangelou est bornée.

♢ **Démonstration :**
Cette condition implique la condition (2.10), puisque

$$h(\mathbf{x}) \leq h(\emptyset) M_h^{n(\mathbf{x})}, \forall \mathbf{x} \in \mathbf{\Psi}.$$

♢

2.1.3 Processus ponctuels de Markov

▸ Définition des processus ponctuels de Markov

Les processus ponctuels de Markov sont des processus ponctuels dont l'intensité conditionnelle de Papangelou ne dépend que d'un certain voisinage. Autrement dit, du point de vue énergétique, leurs densités s'écrivent au moyen de potentiels d'intéractions entre les objets du processus. On parle alors de processus de Gibbs.

Il convient dans un premier temps de se fixer une relation de voisinage $\sim$, symétrique et réflexive sur l'ensemble des singletons de $\mathbf{\Psi}$, autrement dit de $\mathbf{\Psi}_1$, qui est en fait U. On peut par exemple considérer la relation de proximité entre deux objets x_1 et x_2 de $U = \mathbb{R}^2$, où $d(.,.)$ est la fonction distance :

$$x_1 \sim x_2 \Leftrightarrow \|x_1 - x_2\| \leq d_\sim, d_\sim \in \mathbb{R}^+ \tag{2.12}$$

Il nous reste à définir la notion de densité héréditaire avant d'introduire la notion de processus ponctuel de Markov.

Définition 2.12

Une densité $f(.)$ est dite ***héréditaire*** si elle vérifie la relation :

$$f(\mathbf{x}) > 0 \Rightarrow f(\mathbf{y}) > 0, \forall \mathbf{y} \subseteq \mathbf{x}, \forall \mathbf{x} \in \mathbf{\Psi}.$$

Définition 2.13

On appelle ***processus ponctuel de Markov*** (au sens de Ripley-Kelly) sous la relation symétrique et réflexive $\sim$, tout processus dont la densité $f(.)$ est telle que :

1. $f(.)$ est héréditaire
2. Pour toute configuration $\mathbf{x}$ telle que $f(\mathbf{x}) > 0$, l'intensité conditionnelle de Papangelou $\lambda^*(\mathbf{x},\xi)$ dépend uniquement de ξ et des objets de $\mathbf{x}$ voisins de ξ au sens de la relation de voisinage $\sim$.

L'introduction de la notion de clique permet ensuite la caractérisation des processus ponctuels de Markov.

Définition 2.14

Soit $\sim$ une relation de voisinage symétrique et réflexive sur $\mathbf{\Psi}$. Une configuration $\mathbf{x}$ de $\mathbf{\Psi}$ est appelée une ***clique*** si tous les éléments de $\mathbf{x}$ sont voisins les uns des autres, c'est à dire si :

$$\forall x_1,x_2 \in \mathbf{x}, x_1 \sim x_2.$$

Le théorème suivant nous donne une condition nécessaire et suffisante sur la densité d'un processus pour que ce soit un processus ponctuel de Markov.

Théorème 2.15 *:* **Processus ponctuel de Markov**

> Un processus ponctuel de densité $f(.)$ est un ***processus ponctuel de Markov*** (au sens de Ripley-Kelly) si et seulement si $f(.)$ est de la forme :
>
> $$\begin{aligned} f(\mathbf{x}) &= \prod_{cliques\ \mathbf{y}\subseteq\mathbf{x}} \phi(\mathbf{y}) \\ &= \prod_{\mathbf{y}\subseteq\mathbf{x}} \psi(\mathbf{y}), \forall \mathbf{x} \in \mathbf{\Psi} \end{aligned}$$
>
> où $\psi(.)$ est une fonction d'intéraction, c'est à dire une fonction telle que $\psi(\mathbf{y}) = 1$ lorsque $\mathbf{y}$ n'est pas une clique.

Nous allons maintenant présenter deux exemples de processus ponctuels de Markov.

▸ Exemples de processus ponctuels de Markov

On peut exprimer les densités des processus des exemples suivants sous forme exponentielle, c'est à dire qu'il existe un vecteur $\theta \in \mathbb{R}^d$ et une application $\rho(.) : \mathbf{\Psi} \to \mathbb{R}^d$ telles que :

$$h(\mathbf{x}) = e^{<\theta,\rho(\mathbf{x})>}$$

où $< .,. >$ est le produit scalaire sur $\mathbb{R}^d$.

▷ Processus de Strauss

Soit $\mathbf{x}$ une configuration de l'espace $(\mathbf{\Psi},\mathcal{B}(\mathbf{\Psi}),\mu(.))$. Nous notons d'une part $n(\mathbf{x})$ le nombre d'objets contenus dans $\mathbf{x}$. Autrement dit, $\mathbf{x} \in \mathbf{\Psi}_{n(\mathbf{x})}$. D'autre part, $s(\mathbf{x})$ représente le nombre de paires d'objets voisins, au sens de la relation de voisinage $\sim$, dans la configuration $\mathbf{x}$. La densité d'un processus de Strauss s'écrit alors

$$h(\mathbf{x}) = e^{\beta.n(\mathbf{x})+\gamma.s(\mathbf{x})}, (\beta,\gamma) \in \mathbb{R}^2 \tag{2.13}$$

avec

$$s(\mathbf{x}) = card(\{\xi,\eta\} \subseteq \mathbf{x} : \{\xi\} \sim \{\eta\}, \{\xi\} \neq \{\eta\}). \tag{2.14}$$

En choisissant par exemple pour relation de voisinage $\sim$ la relation (2.12) l'intensité conditionnelle de Papangelou s'exprime sous la forme :

$$\lambda^*(\mathbf{x},\xi) = e^{\beta+\gamma.s(\mathbf{x},\xi)}$$

où

$$s(\mathbf{x},\xi) = \sum_{x_i\in\mathbf{x}} \mathbf{1}_{\|x_i-\xi\|\leq d_\sim}$$

Selon les valeurs de γ, un processus de Strauss peut avoir différents aspects :

1. Si $\gamma = 0$ on retrouve un processus ponctuel de Poisson homogène d'intensité $e^{\beta}.\lambda(.)$.

2. Si $\gamma < 0$ le processus montre une répulsion entre les points voisins, de plus en plus forte quand γ tend vers 0 (cf Fig. (2.4)). La limite $\gamma \to -\infty$ est un processus dit *hard-core*, la densité interdisant d'avoir deux objets voisins dans la configuration $\mathbf{x}$.

3. Si $\gamma > 0$, le processus de Strauss est attractif. Il y a explosion du processus (cf Fig. (2.5)). La densité n'est intégrable que si l'on change le terme $\beta^{n(\mathbf{x})}$ en un terme adéquat, comme $\mathbf{1}\{n(\mathbf{x}) \leq n_0\}$ par exemple. Ce dernier n'autorise pas de configurations $\mathbf{x}$ de taille supérieure à n_0.

▷ **Processus d'intéraction d'aire**

Le processus d'interaction d'aire possède des propriétés attractives intéressantes en traitement d'images, où il y a souvent des zones d'intérêt que l'on souhaite explorer sans risquer de faire exploser notre processus ponctuel (ce qui arrive avec le processus de Strauss).

Sa densité comprend un terme dépendant de la mesure $\nu(.)$ de la réunion des supports $Z(x_i)$ des objets de la configuration $\mathbf{x}$, $Z(\mathbf{x})$:

$$Z(\mathbf{x}) = \bigcup_{x_i \in \mathbf{x}} Z(x_i).$$

Elle s'écrit

$$h(\mathbf{x}) = e^{\beta.n(\mathbf{x}) - \gamma.\nu(Z(\mathbf{x}))}. \tag{2.15}$$

Plusieurs cas sont à différencier :

1. Lorsque $\gamma = 0$, ce processus est un processus de Poisson homogène d'intensité $e^{\beta}.\lambda(.)$.

2. Pour $\gamma > 0$ ce processus est attractif car il a tendance à favoriser les configurations $\mathbf{x}$ avec une faible valeur de $\nu\left(Z(\mathbf{x})\right)$.

3. Pour $\gamma < 0$, ce processus est régulier. Lorsque $\gamma \to -\infty$, ce processus converge vers un processus uniforme sur l'ensemble

   $$\mathbb{H} = \{\mathbf{x} \in \mathbf{\Psi} : \nu(Z(\mathbf{x})) = \nu^*\}, \nu^* = max_{\mathbf{x} \in \mathbf{\Psi}} \nu(Z(\mathbf{x})).$$

   $\mathbb{H}$ est donc constitué des configurations dont le support a une mesure maximale.

Couplé à une répulsion entre les objets se chevauchant, on peut alors obtenir des configurations qui recouvrent l'espace U entièrement (si on choisit $\gamma < 0$: Fig. (2.6) ou Fig. (2.7)), dont la densité s'exprime sous la forme :

$$h(\mathbf{x}) = e^{\beta.n(\mathbf{x}) + \gamma_1.s(\mathbf{x}) - \gamma_2.\nu(Z(\mathbf{x}))}.$$

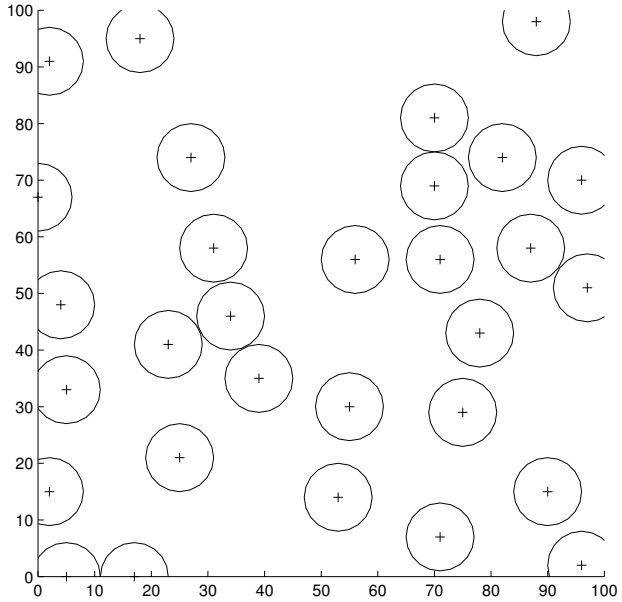

FIG. 2.4 – *Réalisation d'un processus de Strauss répulsif:* $\beta = ln(0.003), \gamma = ln(0.001)$ *et* $d_{\sim} = 6$

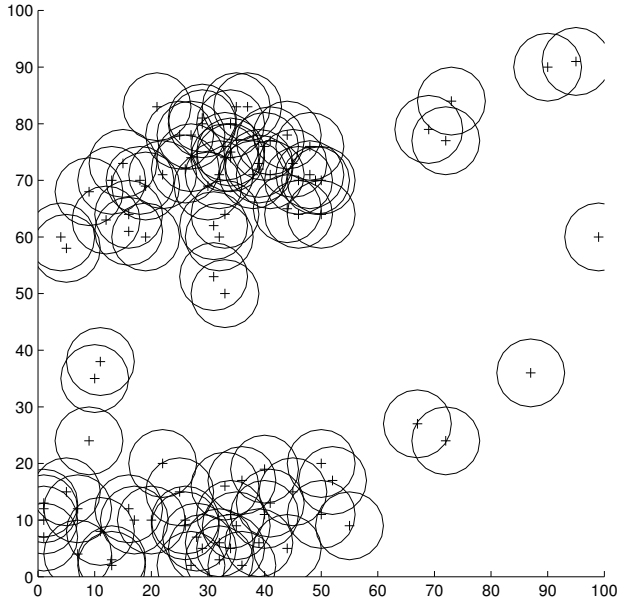

FIG. 2.5 – *Réalisation d'un processus de Strauss attractif:* $\beta = ln(0.003), \gamma = ln(2)$ *et* $d_{\sim} = 6$

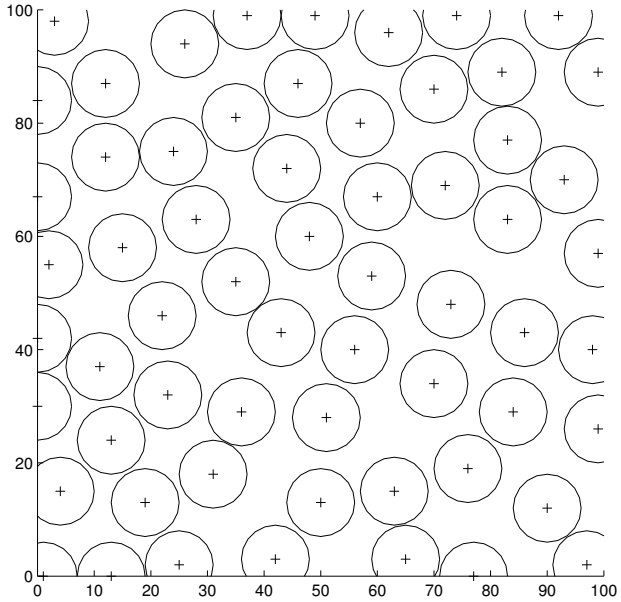

FIG. 2.6 – *Réalisation d'un processus d'intéraction d'aires, avec une répulsion :* $\beta = ln(0.003), \gamma_1 = ln(0.001), \gamma_2 = -5$ *et* $d_\sim = 6$

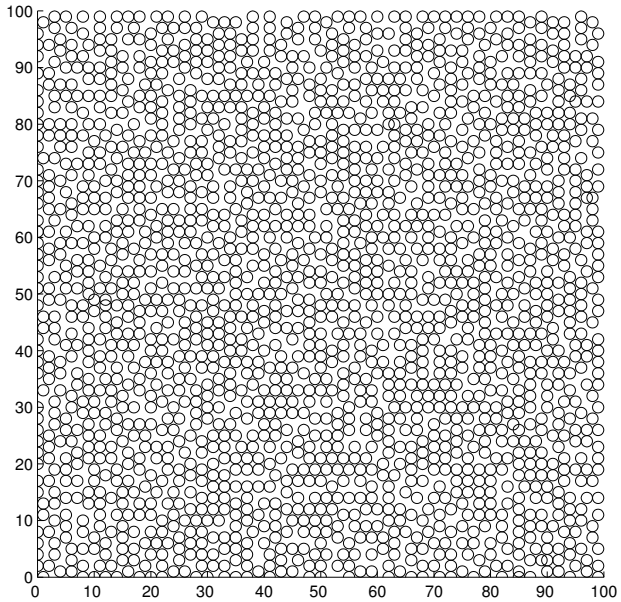

FIG. 2.7 – *Réalisation d'un processus d'intéraction d'aires, avec une répulsion :* $\beta = ln(0.003), \gamma_1 = ln(0.001), \gamma_2 = -5$ *et* $d_\sim = 1$

2.2. Chaînes de Markov et Théorie ergodique

Comme nous l'avons remarqué précédemment, les processus objet sont généralement définis par une densité a priori non normalisée sur l'espace probabilisé $(\mathbf{\Psi},\mathcal{B}(\mathbf{\Psi}),\mu(.))$. Autrement dit, ils sont définis par une densité par rapport à un processus de Poisson de référence, dont le calcul de la constante de normalisation $\int_{\mathbf{\Psi}} h(\mathbf{x})\mu(d\mathbf{x})$ n'est pas envisageable en pratique.

C'est pourquoi des méthodes pour obtenir des réalisations de tels processus sont utiles. Parmi elles, on distingue les algorithmes de type Monte Carlo, où l'on simule une chaîne de Markov convergeant vers la distribution recherchée.

Cette partie a donc pour but de nous rappeler quelques notions autour des chaînes de Markov et sur la théorie ergodique. Le lecteur intéressé par la théorie des chaînes de Markov et les algorithmes Markov Chain Monte Carlo (ou MCMC) pourra se reporter à [1], [23] et [27].

Soit $\pi(.)$ une loi sur l'espace probabilisé $(\mathbf{\Psi},\mathcal{B}(\mathbf{\Psi}),\mu(.))$, que nous cherchons à simuler. Dans l'étude ultérieure des algorithmes, on supposera que cette loi possède une densité $h(.)$ par rapport à la loi de référence $\mu(.)$. Sous certaines conditions une chaîne de Markov, dans l'espace probabilisé $(\mathbf{\Psi},\mathcal{B}(\mathbf{\Psi}),\mu(.))$, converge vers une suite d'échantillons suivant $\pi(.)$.

2.2.1 Notions sur les chaînes de Markov

Nous recherchons des conditions suffisantes pour que la chaîne de Markov converge vers une distribution $\pi(.)$ que l'on souhaite simuler, quel que soit l'état initial.

▸ Premières définitions

Définition 2.16

> Une suite aléatoire de configurations $(\mathbf{x}_i)$ forme une ***chaîne de Markov*** si la loi conditionnelle de $\mathbf{x}_j$ sachant $\mathbf{x}_{j-1},\mathbf{x}_{j-2},\ldots$ est la même que celle de $\mathbf{x}_j$ sachant $\mathbf{x}_{j-1}$.
> On dit de plus que la loi est ***homogène*** si la loi de $(\mathbf{x}_{j_1},\ldots,\mathbf{x}_{j_k})$ sachant $\mathbf{x}_{j_0}$ est la même que celle de $(\mathbf{x}_{j_1-j_0},\ldots,\mathbf{x}_{j_k-j_0})$ sachant $\mathbf{x}_0$ pour tout k et tout $(k+1)$-uplets $j_0 \leq j_1 \leq \ldots \leq j_k$.

Une chaîne de Markov est caractérisée par :

1. Sa loi initiale, c'est à dire la loi selon laquelle est tiré le premier échantillon $\mathbf{x}_0$.

2. Son noyau de transition, qui caractérise comment la chaîne passe d'un échantillon au suivant.

Définition 2.17

> On appelle ***noyau de transition*** toute fonction $K(.,.)$ définie sur $\mathbf{\Psi} \times \mathcal{B}(\mathbf{\Psi})$ telle que :
> 1. $\forall \mathbf{x} \in \mathbf{\Psi}, K(\mathbf{x},.)$ est une mesure de probabilité.
> 2. $\forall B \in \mathcal{B}(\mathbf{\Psi}), K(.,B)$ est mesurable.

Ce noyau nous permet de connaître le comportement de la chaîne puisque $\forall (k,B) \in (\mathbb{N} \times \mathcal{B}(\mathbf{\Psi}))$,

$$\begin{aligned}\mathbb{P}(\mathbf{x}_{k+1} \in B|\mathbf{x}_0,\ldots,\mathbf{x}_k) &= \mathbb{P}(\mathbf{x}_{k+1} \in B|\mathbf{x}_k) \\ &= \int_B K(\mathbf{x}_k,d\mathbf{x}).\end{aligned}$$

En posant $K^1(\mathbf{x},B) = K(\mathbf{x},B)$, nous noterons par la suite $K^n(\mathbf{x},B)$ le noyau de n transitions défini par récurrence par la formule suivante:

$$K^n(\mathbf{x},B) = \int_{\mathbf{\Psi}} K^{n-1}(\mathbf{y},B)K(\mathbf{x},d\mathbf{y}). \tag{2.16}$$

De même, nous emploierons la notation $P_{\mathbf{x}}(.)$ pour signifier $P(.|\mathbf{x}_0 = \mathbf{x})$.

▶ Stationnarité et Invariance

La stationnarité ou invariance est une propriété de base que l'on désire pour une chaîne de Markov.

Définition 2.18

Une mesure $\pi(.)$ est dite ***invariante*** pour le noyau de transition $K(.,.)$ d'une chaîne de Markov si $\pi K = \pi$, à savoir:

$$\begin{aligned}(\pi K)(B) &= \int_{\mathbf{\Psi}} K(\mathbf{x},B)\pi(d\mathbf{x}) \\ &= \pi(B)\,, \forall B \in \mathcal{B}(\mathbf{\Psi}).\end{aligned}$$

Si en outre, la mesure $\pi(.)$ est une mesure de probabilité, alors elle est appelée ***mesure stationnaire*** de la chaîne.
On dit que la chaîne est π***-invariante*** ou π***-stationnaire***.

Remarque:

Si une mesure $\pi(.)$ est stationnaire pour plusieurs noyaux de transitions $K_1(.,.),\ldots,K_n(.,.)$, alors $\pi(.)$ est aussi une mesure stationnaire pour le noyau de transition $K(.,.) = \sum_{i=1}^n p_i K_i(.,.)$, avec les $p_i > 0$ et $\sum_{i=1}^n p_i = 1$.

En effet,

$$\begin{aligned}(\pi K)(B) &= \int_{\mathbf{\Psi}} K(\mathbf{x},B)\pi(d\mathbf{x}) \\ &= \sum_{i=1}^n p_i \int_{\mathbf{\Psi}} K_i(\mathbf{x},B)\pi(d\mathbf{x}) \\ &= \sum_{i=1}^n p_i \pi(B) \\ &= \pi(B)\,, \forall B \in \mathcal{B}(\mathbf{\Psi}).\end{aligned}$$

Ceci nous sera très utile par la suite, lorsque notre noyau de transition sera défini par plusieurs noyaux de transitions choisis avec une probabilité p_i à chaque saut de la chaîne. Cette combinaison convexe de noyaux de proposition est en effet un des piliers de l'algorithme de simulation de

processus ponctuels que nous implanterons.

▸ Réversibilité

Définition 2.19

La chaîne est dite ***réversible*** pour la loi objectif $\pi(.)$ si son noyau de transition vérifie :

$$\int_B K(\mathbf{x},A)\pi(d\mathbf{x}) = \int_A K(\mathbf{x},B)\pi(d\mathbf{x}), \forall A, B \in \mathcal{B}(\mathbf{\Psi}). \qquad (2.17)$$

Proposition 2.20 *:*

La réversibilité implique la stationnarié de la loi $\pi(.)$

♢ **Démonstration :**
En remplaçant dans l'équation (2.17) le borélien B par $\mathbf{\Psi}$, on prouve que la π-réversibilité implique la stationnarié de la loi $\pi(.)$ puisque ce cas particulier mène à :

$$\begin{aligned} \int_{\mathbf{\Psi}} K(\mathbf{x},A)\pi(d\mathbf{x}) &= \int_A K(\mathbf{x},\mathbf{\Psi})\pi(d\mathbf{x}) \\ &= \pi(A)\,, \forall A \in \mathcal{B}(\mathbf{\Psi}). \end{aligned}$$

♢

Réversibilité $\Longrightarrow$ Stationnarité

▸ Irréductibilité

Une première étude de la robustesse d'une chaîne de Markov aux conditions initiales, est le caractère irréductible de la chaîne. Dans le cadre des algorithmes de Monte Carlo, elle permet d'assurer la convergence vers la distribution objectif sans pour autant nécessiter une étude préalable poussée du noyau de transition nécessaire à la détermination de conditions initiales correctes.

La ϕ-irréductibilité traduit la capacité de la chaîne à passer par tous les ensembles de ϕ-mesure non nulle.

Définition 2.21

Etant donnée une mesure $\phi(.)$ sur $\mathcal{B}(\mathbf{\Psi})$, la chaîne $(\mathbf{x}_i)$ est dite ***ϕ-irréductible*** si $\forall B \in \mathcal{B}(\mathbf{\Psi})$ tel que $\phi(B) > 0$:

$$\exists n \,:\, K^n(\mathbf{x},B) > 0, \forall \mathbf{x} \in \mathbf{\Psi}.$$

S'il existe une mesure $\phi(.)$ telle que la chaîne soit ϕ-irréductible, alors elle est aussi π-irréductible pour une mesure stationnaire $\pi(.)$, qui dans ce cas est la seule mesure stationnaire pour la chaîne.

Définition 2.22

Si la chaîne de Markov est ϕ-irréductible et qu'il existe une mesure stationnaire, on dit qu'elle est ***positive***.

Irréductibilité + Stationnarité $\Longrightarrow$ Positivité

▸ Ensembles petits et Apériodicité

Définition 2.23

On appelle ***ensemble petit*** tout ensemble $C \in \mathcal{B}(\mathbf{\Psi})$ tel qu'il existe un entier n et une mesure non nulle $\nu(.)$ vérifiant :

$$K^n(\mathbf{x},B) \geq \nu(B), \forall \mathbf{x} \in C, \forall B \in \mathcal{B}(\mathbf{\Psi}).$$

Nous verrons par la suite que l'on attend d'une chaîne de Markov que tous les ensembles bornés mesurables de l'espace des configurations soient petits.

Nous pouvons alors définir la notion de période d'une chaîne de Markov sur un espace d'états continu. Celle-ci diffère de la définition donnée dans le cadre d'une chaîne de Markov simulée sur un espace d'états discret, pour laquelle la période est le plus grand commun diviseur (pgcd) des t tels que pour tout $\mathbf{x}$, $K^t(\mathbf{x},\mathbf{x}) > 0$.

Définition 2.24

Une chaîne $(\mathbf{x}_i)$ ϕ-irréductible admet un ***cycle de longueur*** d s'il existe un ensemble petit C, d'entier et de loi de probabilité associés M et $\nu_M(.)$, tel que d soit le p.g.c.d. de

$$\{m \geq 1;\ \exists \delta_m > 0 \,|\, C \, petit \, pour \, m \, et \, \nu_m \geq \delta_m \nu_M\}.$$

Une chaîne est dite ***apériodique*** si elle est irréductible de période 1.

Une condition suffisante pour assurer l'apériodicité est que l'on autorise la chaîne de rester dans le même état :

$$K(\mathbf{x},\mathbf{x}) > 0, \forall \mathbf{x} \in \mathbf{\Psi} \tag{2.18}$$

2.2.2 Etude de la stabilité des chaînes de Markov

Il s'agit désormais de faire converger cette chaîne vers la distribution objectif $\pi(.)$. Il existe plusieurs critères de convergence. Le critère de convergence le plus faible est la convergence en loi, aussi appelé ergodicité.

▸ Ergodicité

Définition 2.25

Une chaîne de Markov de noyau de transition $K(.,.)$ ***converge en loi*** vers la distribution objectif $\pi(.)$ si pour presque tout $\mathbf{x} \in \mathbf{\Psi}$:

$$\lim_{n \to \infty} |K^n(\mathbf{x},.) - \pi(.)| = 0. \tag{2.19}$$

On dit dans ce cas qu'elle est ***ergodique***.

Proposition 2.26 *:* **Ergodicité**

Une chaîne de Markov irréductible, apériodique, et ayant une distribution invariante $\pi(.)$, est ergodique.

Positivité + Apériodicité $\Longrightarrow$ Ergodicité

▸ Récurrence au sens de Harris

Un autre type de convergence, plus fort, est la convergence au sens de la norme en variations totales défini pour deux mesures $\mu_1(.)$ et $\mu_2(.)$ par :

$$\|\mu_1(.) - \mu_2(.)\|_{VT} = max_{B \in \mathcal{B}(\mathbf{\Psi})} |\mu_1(B) - \mu_2(B)|. \tag{2.20}$$

Cette convergence permet de supprimer les problèmes de non-convergence de la chaîne pour certaines configurations initiales, mais requiert, en plus de la positivité et de l'apériodicité de la chaîne, la récurrence au sens de Harris. Il s'agit d'une notion forte, puisqu'elle impose un retour certain en un temps fini dans tout ensemble de mesure positive de l'espace.

Définition 2.27

Une chaîne $(\mathbf{x}_i)$ est ***récurrente au sens de Harris*** s'il existe une mesure $\phi(.)$ telle qu'elle soit ϕ-irréductible, et si tout ensemble A, tel que $\phi(A) > 0$, est ***récurrent au sens de Harris***, c'est-à-dire vérifie

$$P_{\mathbf{x}} \left(\sum_{i=1}^{\infty} \mathbf{1}_A(\mathbf{x}_i) = \infty \right) = 1, \forall \mathbf{x} \in A. \tag{2.21}$$

Ceci équivaut à $P_{\mathbf{x}}(\exists t : \mathbf{x}_t \in A) = 1$.
Une chaîne est ***positive au sens de Harris*** lorsqu'elle est récurrente au sens de Harris et qu'il existe une mesure de probabilité invariante pour cette chaîne.

On a finalement le théorème suivant qui assure la convergence en variations totales :

Théorème 2.28 *:* **Convergence en variations totales**

Si $(\mathbf{x}_i)$ est positive au sens de Harris et apériodique,

$$\lim_{n\to\infty} \left\| \int K^n(\mathbf{x},.)\nu(d\mathbf{x}) - \pi(.) \right\|_{VT} = 0 \tag{2.22}$$

pour toute distribution initiale $\nu(.)$.

Positivité au sens de Harris + Apériodicité $\Longrightarrow$ Convergence en variation totale

▸ Théorème limite

Théorème 2.29 *:* **Loi des grands nombres**

Soit $(\mathbf{x}_i)$ une chaîne positive au sens de Harris, de loi invariante intégrable $\pi(.)$. Alors pour tout couple de fonctions f et g strictement positives dans $\mathcal{L}^1(\mathbf{\Psi})$,

$$\lim_{n\to+\infty} \frac{\sum_{i=1}^n f(\mathbf{x}_i)}{\sum_{i=1}^n g(\mathbf{x}_i)} = \frac{\int f(x)d\pi(x)}{\int g(x)d\pi(x)}. \tag{2.23}$$

En particulier pour $g \equiv 1$:

$$\lim_{n\to+\infty} \frac{\sum_{i=1}^n f(\mathbf{x}_i)}{n} = \int f(x)d\pi(x). \tag{2.24}$$

Ce théorème pourra nous servir lorsque nous souhaiterons calculer des statistiques sur nos échantillons.

2.3. Algorithme de simulation : Metropolis-Hastings-Green

Soit $\pi(.)$ la loi objectif que l'on souhaite simuler. Green, dans [11], présente un algorithme de type Metropolis-Hastings à sauts réversibles, permettant de simuler une loi connue à un facteur multiplicatif près grâce à une chaîne de Markov. Cela nous permettra de ne pas calculer la constante de normalisation.

L'autre avantage de cet algorithme est qu'il autorise cette simulation sur des espaces dont la dimension varie, ce qui se prête tout à fait à notre étude.

Nous exposerons tout d'abord dans cette partie le principe de l'algorithme en l'appliquant au cas des processus ponctuels. Puis, nous remarquerons que sous certaines conditions, cet algorithme converge vers la loi objectif $\pi(.)$.

2.3.1 Présentation de l'algorithme

L'algorithme de Metropolis-Hastings-Green (MHG) permet la simulation au sein d'un même espace de dimension fixée, ainsi que les transitions (réversibles) d'un espace vers un autre espace de dimension différente. Nous pourrons donc explorer l'espace des configurations $\mathbf{\Psi}$ dans son ensemble, c'est à dire tous les $\mathbf{\Psi}_n$.

Il est présenté par Green comme une généralisation de l'algorithme de Metropolis-Hastings. La principale différence entre les deux est que Green remplace les densités non normalisées de l'algorithme de Metropolis-Hastings par des mesures non normalisée sur l'espace des configurations $\mathbf{\Psi}$. Quant à la densité de transition auxiliaire, elle est remplacée par le noyau $Q(\mathbf{x},B)$, avec $\mathbf{x} \in \mathbf{\Psi}$ et $B \in \mathcal{B}(\mathbf{\Psi})$.

▸ Le principe

Soient alors :

- $\pi(.)$ la loi objectif définie sur l'espace des configurations $\mathbf{\Psi}$.
- $Q(\mathbf{x},B)$ le noyau de propositions défini sur $\mathbf{\Psi} \times \mathcal{B}(\mathbf{\Psi})$.
- $(\mathbf{x}_i)$ une chaîne de Markov sur $\mathbf{\Psi}$.

Le principe de l'algorithme de Metropolis-Hastings-Green est le suivant : à chaque itération, on propose une modification de l'état courant $\mathbf{x}_t$ suivant le noyau de proposition $Q(\mathbf{x}_t,.)$, que l'on accepte ou que l'on rejette avec une probabilité α. Cette modification de l'état courant $\mathbf{x}_t$ vers un état $\mathbf{y}$ est communément appelée ***mouvement***.

Comme démontré dans [11] par Green, le calcul de la probabilité d'acceptation d'un mouvement implique la π-réversibité, et donc l'invariance de la mesure $\pi(.)$ pour notre chaîne de Markov.

Pour obtenir un tel résultat, Green suppose l'existence d'une mesure symétrique $\xi(.,.)$, de telle sorte que $\pi(.)Q(.,.)$ soit absolument continu par rapport à $\xi(.,.)$: $\pi(.)Q(.,.) \ll \xi(.,.)$.

Une condition suffisante pour qu'une mesure soit absolument continue par rapport à une autre est donnée par la proposition suivante.

Proposition 2.30 *:* **Radon-Nikodym**

Soient $\mu_1(.)$ et $\mu_2(.)$ deux mesures définies sur un même ensemble $\mathbf{\Psi}$. Si pour tout borélien $B \in \mathbf{\Psi}$,

$$\mu_2(B) = 0 \Rightarrow \mu_1(B) = 0, \tag{2.25}$$

alor $\mu_1(.)$ est absolument continue par rapport à $\mu_2(.)$.

Dans ce cas, $\pi(d\mathbf{x})Q(\mathbf{x},d\mathbf{y})$ admet une densité finie $f(\mathbf{x},\mathbf{y})$ par rapport à $\xi(d\mathbf{x},d\mathbf{y})$ sur $\mathbf{\Psi} \times \mathbf{\Psi}$, et la dérivée de Radon-Nikodym s'écrit :

$$f(\mathbf{x},\mathbf{y}) = \frac{\pi(d\mathbf{x})Q(\mathbf{x},d\mathbf{y})}{\xi(d\mathbf{x},d\mathbf{y})}. \tag{2.26}$$

Ce point sera détaillé après la présentation du noyau de proposition.

▸ Rôle et composition du noyau de proposition

Nous allons présenter cet algorithme dans le cadre d'un noyau de proposition multiple : $Q(.,.)$ sera alors formé de plusieurs noyaux de proposition $Q_i(.,.)$ décrivant chacun un mouvement différent que l'on autorise :

$$Q(\mathbf{x},.) = \sum_{i \in I} Q_i(\mathbf{x},.) < 1.$$

Ainsi, à chaque itération t, nous sélectionnons dans un premier temps parmi les noyaux $Q_i(.,.)$ celui que l'on va utiliser pour proposer un mouvement à partir de la configuration courante $\mathbf{x}_t$. Chacun des noyaux Q_i est choisi avec la probabilité $Q_i(\mathbf{x}_t,\mathbf{\Psi})$, tandis qu'avec la probabilité restante, à savoir $1 - \sum_{i \in I} Q_i(\mathbf{x}_t,\mathbf{\Psi})$, nous imposons de rester dans l'état courant.

Comme nous le verrons plus tard, pour tout $i \in I$, il est nécessaire que $\pi(.)Q_i(.,.)$ soit absolument continue par rapport à une mesure symétrique $\xi_i(.,.)$ sur $\mathbf{\Psi} \times \mathbf{\Psi}$. De cette façon, on peut définir pour chacun des noyaux une dérivée de Radon Nikodym particulière $f_i(.,.)$:

$$f_i(\mathbf{x},\mathbf{y}) = \frac{\pi(d\mathbf{x})Q_i(\mathbf{x},d\mathbf{y})}{\xi_i(d\mathbf{x},d\mathbf{y})}. \tag{2.27}$$

Le rôle de ces noyaux de proposition est fondamental. Ils jouent essentiellement sur la vitesse de convergence de notre algorithme, et sur l'exactitude du résultat obtenu. En effet, ils doivent nous permettre à chaque itération d'explorer une grande partie de l'espace des configurations afin de ne pas rester bloqué dans un minimum local de l'énergie à minimiser ($\Leftrightarrow$ densité à maximiser).

▸ Détails de l'algorithme

Metropolis-Hastings-Green

1. A partir de la configuration courante $\mathbf{x}_t$:

   $\rightarrow$ on choisit avec la probabilité $p_i(\mathbf{x}_t) = Q_i(\mathbf{x}_t,\mathbf{\Psi})$, un des noyaux de proposition $Q_i(\mathbf{x}_t,.)$ et on passe à l'étape 2.

   $\rightarrow$ avec la probabilité $1 - \sum_{i\in I} p_i(\mathbf{x})$, on reste dans l'état courant et on passe à la prochaine itération.

2. On génère la configuration $\mathbf{y}$, connaissant la configuration $\mathbf{x}_t$, selon le noyau sélectionné $\frac{Q_i(\mathbf{x}_t,.)}{p_i(\mathbf{x})}$.

3. On calcule le ***rapport de Green*** relatif au noyau i, $R_i(\mathbf{x}_t,\mathbf{y})$:

$$\boxed{R_i(\mathbf{x}_t,\mathbf{y}) = \frac{f_i(\mathbf{y},\mathbf{x}_t)}{f_i(\mathbf{x}_t,\mathbf{y})} = \frac{\pi(d\mathbf{y})Q_i(\mathbf{y},d\mathbf{x}_t)}{\pi(d\mathbf{x}_t)Q_i(\mathbf{x}_t,d\mathbf{y})}} \tag{2.28}$$

4. Avec la probabilité $\alpha_i(\mathbf{x}_t,\mathbf{y}) = min(1,R_i(\mathbf{x}_t,\mathbf{y}))$ on accepte le nouvel état $\mathbf{y}$, sinon on conserve $\mathbf{x}_t$.

▸ Preuve de π-réversibilité

La richesse de l'algorithme de Green repose sur un choix judicieux de la probabilité d'acceptation du mouvement, qui implique la π-réversibité, puisque l'équation (2.17) est alors satisfaite.

La proposition suivante ainsi que sa démonstration nous indiquent comment Green parvient à l'expression de cette probabilité.

Proposition 2.31 *:* **Rapport de Green**

Soient $Q_i(\mathbf{x},A)$, $i \in I$ et $A \in \mathbb{B}(\mathbf{\Psi})$, les différents noyaux de proposition vérifiant $\sum_{i\in I} Q_i(\mathbf{x},\mathbf{\Psi}) \leq 1$ pour tout $\mathbf{x} \in \mathbf{\Psi}$. Alors l'algorithme proposé par Green simule une chaîne de Markov réversible.

$\diamondsuit$ **Démonstration :**
Montrons en effet que (2.17) est vérifiée. Tout d'abord, notons que puisque $\sum_{i\in I} Q_i(\mathbf{x},\mathbf{\Psi}) \leq 1$, aucun mouvement n'est proposé avec la probabilité $r(\mathbf{x}) = 1 - \sum_{i\in I} Q_i(\mathbf{x},\mathbf{\Psi})$.

Nous définissons alors $\alpha_i(\mathbf{x},\mathbf{x}')$ comme la probabilité d'accepter un mouvement de $\mathbf{x}$ à $\mathbf{x}'$ avec le noyau de proposition Q_i, et

$$\rho_i(\mathbf{x}) = \int_{\mathbf{\Psi}} Q_i(\mathbf{x},d\mathbf{x}')[1 - \alpha_i(\mathbf{x},\mathbf{x}')]$$

comme la probabilité de rester en $\mathbf{x}$ malgré les propositions du noyau Q_i.

Le noyau de transition de la chaîne $K(.,.)$ peut alors s'écrire sous la forme suivante, $\forall \mathbf{x} \in \mathbf{\Psi}$ et $\forall B \in \mathbb{B}(\mathbf{\Psi})$:

$$K(\mathbf{x},B) = \rho(\mathbf{x})\mathbf{1}_{\mathbf{x}\in B} + \sum_{i\in I} \int_B Q_i(\mathbf{x},d\mathbf{x}')\alpha_i(\mathbf{x},\mathbf{x}') \tag{2.29}$$

avec $\rho(\mathbf{x})$ la probabilité de rester en $\mathbf{x}$, soit suite à un mouvement proposé par un des noyaux Q_i qui n'a pas été accepté, soit puisque aucun mouvement n'a été proposé :

$$\rho(\mathbf{x}) = r(\mathbf{x}) + \sum_{i \in I} \rho_i(\mathbf{x}).$$

En remplaçant (2.29) dans (2.17), on obtient l'équation suivante qui doit être vérifiée pour que notre noyau de proposition convienne :

$$\sum_{i \in I} \int_B \pi(d\mathbf{x}) \int_A Q_i(\mathbf{x},d\mathbf{x}')\alpha_i(\mathbf{x},\mathbf{x}') = \sum_{i \in I} \int_A \pi(d\mathbf{x}') \int_B Q_i(\mathbf{x}',d\mathbf{x})\alpha_i(\mathbf{x},\mathbf{x}'). \tag{2.30}$$

Pour que (2.30) tienne, il suffit que pour tout $i \in I$, $A,B \in \mathbb{B}(\mathbf{\Psi})$

$$\int_B \pi(d\mathbf{x}) \int_A Q_i(\mathbf{x},d\mathbf{x}')\alpha_i(\mathbf{x},\mathbf{x}') = \int_A \pi(d\mathbf{x}') \int_B Q_i(\mathbf{x}',d\mathbf{x})\alpha_i(\mathbf{x},\mathbf{x}').$$

En supposant que $\pi(d\mathbf{x})Q_i(\mathbf{x},d\mathbf{x}')$ soit dominé par une mesure symétrique ξ_i sur $\mathbf{\Psi} \times \mathbf{\Psi}$, et qu'il existe une dérivée au sens de Radon-Nikodym

$$f_i(\mathbf{x},\mathbf{x}') = \frac{\pi(d\mathbf{x})Q_i(\mathbf{x},d\mathbf{x}')}{\xi_i(d\mathbf{x},d\mathbf{x}')}$$

on a :

$$\int_B \pi(d\mathbf{x}) \int_A Q_i(\mathbf{x},d\mathbf{x}')\alpha_i(\mathbf{x},\mathbf{x}') = \int_B \int_A \xi_i(d\mathbf{x},d\mathbf{x}')f_i(\mathbf{x},\mathbf{x}')\alpha_i(\mathbf{x},\mathbf{x}') \tag{2.31}$$

et

$$\int_A \pi(d\mathbf{x}') \int_B Q_i(\mathbf{x}',d\mathbf{x})\alpha_i(\mathbf{x}',\mathbf{x}) = \int_A \int_B \xi_i(d\mathbf{x}',d\mathbf{x})f_i(\mathbf{x}',\mathbf{x})\alpha_i(\mathbf{x}',\mathbf{x}). \tag{2.32}$$

Finalement, (2.31) et (2.32) sont égales d'après la symétrie de la mesure ξ_i, et si en outre

$$f_i(\mathbf{x},\mathbf{x}')\alpha_i(\mathbf{x},\mathbf{x}') = f_i(\mathbf{x}',\mathbf{x})\alpha_i(\mathbf{x}',\mathbf{x}).$$

Ceci est vérifié si on pose

$$\alpha_i(\mathbf{x},\mathbf{x}') = min(1, \frac{f_i(\mathbf{x}',\mathbf{x})}{f_i(\mathbf{x},\mathbf{x}')}).$$

Avec ce choix de coefficient d'acceptation du mouvement, la chaîne est réversible et la loi $\pi(.)$ est donc stationnaire.

$\Diamond$

2.3.2 Convergence de la chaîne de Markov considérée

Nous allons démontrer dans cette sous-partie que l'algorithme de Green converge bien vers la mesure objectif $\pi(.)$.

▸ Convergence en variations totales : rappels

Comme exposé précédemment, nous rappelons que les conditions suffisantes pour que la chaîne de Markov simulée par l'algorithme Metropolis-Hastings-Green converge en variations totales vers une mesure objectif $\pi(.)$ sont :

1. la stationnarité de la mesure $\pi(.)$ (déjà démontrée)

2. l'irréductibilité de la chaîne

3. l'apériodicité

4. la récurrence au sens de Harris

Le cas de l'apériodicité peut se régler rapidement. Elle est en effet assurée dès que le noyau $Q(.,.)$ permet de rester dans l'état courant $\mathbf{x}$. Autrement dit, il suffit d'imposer aux différents noyaux $Q_i(.,.)$, quel que soit $\mathbf{x}$, l'inégalité $\sum_{i\in I} p_i(\mathbf{x}) = \sum_{i\in I} Q_i(\mathbf{x}) < 1$.

Les autres propriétés découlent de l'hypothèse de stabilité (2.11) faite sur la densité du processus à modéliser.

Nous arrivons alors à un point où il nous faut définir des mouvements pour parcourir l'espace des configurations. En effet, l'irréductibilité de la chaîne par exemple impose que tout ensemble de mesure non nulle soit accessible à partir de n'importe quel point de l'espace des configurations $\mathbf{\Psi}$. Nous allons donc commencer par définir deux noyaux de proposition qui nous permettront d'explorer tout cet espace et de changer de dimension : le noyau de naissance, et le noyau de mort.

▸ Description des mouvements de naissance et de mort

On souhaite simuler un processus ponctuel de densité $h(.)$ non normalisée par rapport à une mesure d'un processus de Poisson de référence $\mu(.)$, de mesure d'intensité la mesure de Lebesgue $\lambda(.)$ sur U. Par la suite, on notera $\mathbf{x} = \{x_1, \ldots, x_{n(\mathbf{x})}\}$ un élément de l'espace des configurations $\mathbf{\Psi}$.

A partir de la configuration courante $\mathbf{x}_t$, une naissance consiste à proposer d'ajouter un objet u de l'espace objet U à cette configuration, c'est à dire à remplacer $\mathbf{x}_t$ par $\mathbf{y} = \mathbf{x}_t \cup \{u\}$. A l'inverse, la mort consiste en la suppression d'un des objets de la configuration courante $\mathbf{x}_t$. C'est à dire que l'on choisit un entier $i \in [1,n(\mathbf{x})]$, et que l'on propose de remplacer $\mathbf{x}_t$ par $\mathbf{y} = \mathbf{x}_t \setminus \{x_i\} = \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{n(\mathbf{x})}\}$.

Il est essentiel de définir simultanément ces deux noyaux, puisque ce sont deux mouvements symétriques : si on peut passer grâce à une naissance d'une configuration $\mathbf{x}$ à une configuration $\mathbf{y}$, on doit être capable, pour la réversibilité de la chaîne, de passer de $\mathbf{y}$ à $\mathbf{x}$.

▸ Noyaux de proposition de naissance et de mort

On considère un noyau de proposition global $Q(.,.)$, composé lui-même de deux noyaux, l'un proposant une naissance et l'autre une mort :

$$Q(\mathbf{x},.) = Q_n(\mathbf{x},.) + Q_m(\mathbf{x},.)$$

Nous pourrions faire varier les probabilités de proposer une naissance ou une mort en fonction de la configuration courante $\mathbf{x}_t$, comme par exemple en

- proposant plus souvent des naissances quand le nombre d'objets de la configuration est faible.
- proposant plus de morts inversement lorsque la configuration comporte un grand nombre d'objets

Mais afin d'alléger les calculs, nous supposons fixées ces grandeurs :

$$Q_n(\mathbf{x},\mathbf{\Psi}) = p_n \,, Q_m(\mathbf{x},\mathbf{\Psi}) = p_m. \tag{2.33}$$

Pour s'assurer de l'apériodicité de la chaîne de Markov simulée par l'algorithme, il suffirait de choisir p_n et p_m tels que $p_n + p_m = p_{nm} < 1$. Une mise en facteur dans (2.33), ainsi qu'une redéfinition (à ce facteur près) des noyaux de proposition nous mène à :

$$Q(\mathbf{x},.) = p_n Q_n(\mathbf{x},.) + p_m Q_m(\mathbf{x},.).$$

Dans ce cas, on se retrouve avec $Q_n(\mathbf{x},\mathbf{\Psi}) = Q_m(\mathbf{x},\mathbf{\Psi}) = 1$.

▷ **Naissance**

Pour la naissance, on choisit λ-uniformément un objet de U et on propose de l'ajouter à la configuration. On rappelle que $\lambda(.)$ est la mesure de Lebesgue sur l'espace objet U, et qu'elle est la mesure d'intensité du processus de Poisson de référence par rapport auquel notre processus ponctuel est défini.

Soient alors B un borélien de $\mathcal{B}(\mathbf{\Psi})$, et $\mathbf{x}_t$ la configuration courante :

$$Q_n(\mathbf{x}_t,B) = \int_{u\in U} \mathbf{1}_B(\mathbf{x}_t \cup u) \frac{\lambda(du)}{\lambda(U)} \tag{2.34}$$

On remarque bien que si $\mathbf{x}_t \in \mathbf{\Psi}_n$, alors ce noyau concentre tout son poids dans $\mathbf{\Psi}_{n+1}$. Autrement dit, le couple de configurations $(\mathbf{x}_t,\mathbf{y})$ est dans l'ensemble

$$\mathcal{N} = \{(\mathbf{x}_t,\mathbf{y}) \in \mathbf{\Psi}_n \times \mathbf{\Psi}_{n+1} : \mathbf{x}_i = \mathbf{y}_i \forall i \leq n\}.$$

▷ **Mort**

Pour la mort, on choisit uniformément un objet de la configuration courante et on propose de le supprimer. Soient alors B un borélien de $\mathcal{B}(\mathbf{\Psi})$, et $\mathbf{x}_t$ la configuration courante :

$$Q_m(\mathbf{x}_t,B) = \sum_{u\in\mathbf{x}_t} \mathbf{1}_B(\mathbf{x}_t \setminus u) \frac{1}{n(\mathbf{x}_t)} \tag{2.35}$$

On remarque bien que si $\mathbf{x}_t \in \mathbf{\Psi}_{n+1}$, alors ce noyau concentre tout son poids dans $\mathbf{\Psi}_n$. Les configurations étant non ordonnées, on peut s'arranger pour ce que les n premiers objets des deux configurations soient les mêmes. Ainsi le couple de configurations $(\mathbf{x}_t,\mathbf{y})$ est dans l'ensemble

$$\mathcal{M} = \{(\mathbf{x}_t,\mathbf{y}) \in \mathbf{\Psi}_{n+1} \times \mathbf{\Psi}_n : \mathbf{x}_i = \mathbf{y}_i \forall i \leq n\}.$$

▸ Mesure symétrique ξ

Il nous reste à trouver la mesure symétrique $\xi(.,.)$ par rapport à laquelle $\pi(.)Q(.,.)$ est absolument continu. Cette mesure $\xi(.,.)$ est à définir sur $\mathbf{\Psi} \times \mathbf{\Psi}$. Soient alors $A, B \in \mathcal{B}(\mathbf{\Psi})$. Si ni une naissance ni une mort n'est possible pour passer de A à B, alors $\xi(A,B)$ peut être nul puisque dans ce cas $\pi(.)Q(.,.) = 0$ (et donc (2.25) reste vérifiée).

Choisissons alors comme mesure la mesure suivante :

$$\xi(A,B) = \int_{\mathbf{\Psi}} \int_{u\in U} \mathbf{1}_A(\mathbf{x})\mathbf{1}_B(\mathbf{x}\cup u)\lambda(du)\mu(d\mathbf{x}) + \int_{\mathbf{\Psi}} \mathbf{1}_A(\mathbf{x}) \sum_{u\in\mathbf{x}} \mathbf{1}_B(\mathbf{x}\setminus u)\mu(d\mathbf{x}).$$

En remarquant que par définition, $\mathbf{\Psi} = \bigcup_n \mathbf{\Psi}_n$, on a alors :

$$\xi(A,B) = \sum_n \int_{\mathbf{\Psi}_n} \int_{u\in U} \mathbf{1}_{A\cap\mathbf{\Psi}_n}(\mathbf{x})\mathbf{1}_{B\cap\mathbf{\Psi}_{n+1}}(\mathbf{x}\cup u)\lambda(du)\mu(d\mathbf{x}) + \sum_n \int_{\mathbf{\Psi}_n} \mathbf{1}_{A\cap\mathbf{\Psi}_{n+1}}(\mathbf{x}) \sum_{u\in\mathbf{x}} \mathbf{1}_{B\cap\mathbf{\Psi}_n}(\mathbf{x}\setminus u)\mu(d\mathbf{x}).$$

On pose alors $A_n = A\cap\mathbf{\Psi}_n$, $B_n = B\cap\mathbf{\Psi}_n$. On obtient donc pour $\xi(A,B)$ et $\xi(B,A)$:

$$\begin{aligned}
\xi(A,B) &= \sum_n \int_{\mathbf{\Psi}_n} \int_{u\in U} \mathbf{1}_{A_n}(\mathbf{x})\mathbf{1}_{B_{n+1}}(\mathbf{x}\cup u)\lambda(du)\mu(d\mathbf{x}) + \sum_n \int_{\mathbf{\Psi}_n} \mathbf{1}_{A_n}(\mathbf{x}) \sum_{u\in\mathbf{x}} \mathbf{1}_{B_{n-1}}(\mathbf{x}\setminus u)\mu(d\mathbf{x}) \\
&= \sum_n \xi(A_n,B_{n+1}) + \xi(A_n,B_{n-1})
\end{aligned}$$

et

$$\begin{aligned}
\xi(B,A) &= \sum_n \int_{\mathbf{\Psi}_n} \int_{u\in U} \mathbf{1}_{B_n}(\mathbf{x})\mathbf{1}_{A_{n+1}}(\mathbf{x}\cup u)\lambda(du)\mu(d\mathbf{x}) + \sum_n \int_{\mathbf{\Psi}_n} \mathbf{1}_{B_n}(\mathbf{x}) \sum_{u\in\mathbf{x}} \mathbf{1}_{A_{n-1}}(\mathbf{x}\setminus u)\mu(d\mathbf{x}) \\
&= \sum_n \xi(B_n,A_{n+1}) + \xi(B_n,A_{n-1}).
\end{aligned}$$

Il suffit alors, pour obtenir la symétrie de $\xi(.,.)$, c'est à dire l'égalité de $\xi(A,B)$ et $\xi(B,A)$, de prouver que

$$\xi(A_{n+1},B_n) = \xi(B_n,A_{n+1}).$$

Or, d'après la définition de la mesure $\mu(.)$ de référence (2.3)

$$\begin{aligned}
\xi(A_{n+1},B_n) &= \frac{e^{-\lambda(U)}}{(n+1)!} \int_{U^{n+1}} \mathbf{1}_{A_{n+1}}(\mathbf{x}) \sum_{u\in\mathbf{x}} \mathbf{1}_{B_n}(\mathbf{x}\setminus u)\lambda^{n+1}(d\mathbf{x}) \\
&= \frac{e^{-\lambda(U)}}{(n+1)!} \int_{U^{n+1}} \mathbf{1}_{A_{n+1}}(x_1,\ldots,x_n,x_{n+1}) \sum_{i=1}^{n+1} \mathbf{1}_{B_n}(\mathbf{x}\setminus\{x_i\})\lambda(dx_1)\ldots\lambda(dx_{n+1}) \\
&= \frac{e^{-\lambda(U)}}{(n+1)!} \int_{U^{n+1}} (n+1)\mathbf{1}_{A_{n+1}}((x_1,\ldots,x_n)\cup\{x_{n+1}\})\mathbf{1}_{B_n}(x_1,\ldots,x_n)\lambda(dx_1)\ldots\lambda(dx_{n+1}) \\
&= \frac{e^{-\lambda(U)}}{n!} \int_{U^n}\int_U \mathbf{1}_{B_n}(x_1,\ldots,x_n)\mathbf{1}_{A_{n+1}}((x_1,\ldots,x_n)\cup\{x_{n+1}\})\lambda(dx_{n+1})\lambda(dx_1)\ldots\lambda(dx_n) \\
&= \xi(B_n,A_{n+1}).
\end{aligned}$$

Il nous faut désormais calculer la dérivée de Radon Nikodym pour nos deux noyaux.

▸ Dérivées de Radon Nikodym

Une autre démonstration des dérivées de Radon Nikodym pour les noyaux naissance/mort est disponible dans [3]. Elle décompose le noyau $Q(.,.)$ en une somme de noyaux $Q_n(.,.)$ n'agissant que

sur les configurations $\mathbf{x} \in \mathbf{\Psi}_n$.

▷ **Cas d'une naissance**

A partir de la configuration courante $\mathbf{x}_t$, on propose la naissance λ-uniforme d'un objet $u \in U$, à travers la configuration $\mathbf{y} = \mathbf{x}_t \cup u$. La densité objectif $\pi(.)$ est supposée avoir une densité non normalisée $h(.)$ par rapport au processus de Poisson de référence, comme décrit dans les parties antérieures. Le calcul de la dérivée de Radon Nikodym devient alors:

$$\begin{aligned} f(\mathbf{x}_t,\mathbf{y}) &= \frac{\pi(d\mathbf{x}_t)Q(\mathbf{x}_t,d\mathbf{y})}{\xi(d\mathbf{x}_t,d\mathbf{y})} \\ &= \frac{h(\mathbf{x}_t)\mu(d\mathbf{x}_t)p_n Q_n(\mathbf{x}_t,d\mathbf{y})}{\mu(d\mathbf{x}_t)\lambda(du)} \\ &= \frac{h(\mathbf{x}_t)p_n\lambda(du)}{\lambda(U)\lambda(du)} \\ &= \frac{h(\mathbf{x}_t)p_n}{\lambda(U)} \end{aligned}$$

▷ **Cas d'une mort**

A partir de la configuration courante $\mathbf{x}_t$, on propose la mort d'un objet $u \in \mathbf{x}_t$, à travers la configuration $\mathbf{y} = \mathbf{x}_t \setminus u$. La dérivée de Radon Nikodym s'écrit:

$$\begin{aligned} f(\mathbf{x}_t,\mathbf{y}) &= \frac{\pi(d\mathbf{x}_t)Q(\mathbf{x}_t,d\mathbf{y})}{\xi(d\mathbf{x}_t,d\mathbf{y})} \\ &= \frac{h(\mathbf{x}_t)\mu(d\mathbf{x}_t)p_m Q_m(\mathbf{x}_t,d\mathbf{y})}{\mu(d\mathbf{x}_t)} \\ &= \frac{h(\mathbf{x}_t)p_m}{n(\mathbf{x}_t)} \end{aligned}$$

▸ Calcul du rapport de Green

En gardant les mêmes notations que précédemment, le calcul du rapport de Green est maintenant immédiat:

▷ **Cas d'une naissance**

$$\boxed{R(\mathbf{x}_t,\mathbf{x}_t \cup \{u\}) = \frac{f(\mathbf{x}_t \cup \{u\},\mathbf{x}_t)}{f(\mathbf{x}_t,\mathbf{x}_t \cup \{u\})} = \frac{p_m}{p_n}\frac{h(\mathbf{x}_t \cup \{u\})}{h(\mathbf{x})}\frac{\lambda(U)}{n(\mathbf{x})+1}} \tag{2.36}$$

▷ **Cas d'une mort**

$$\boxed{R(\mathbf{x}_t,\mathbf{x}_t \setminus \{u\}) = \frac{f(\mathbf{x}_t \setminus \{u\},\mathbf{x}_t)}{f(\mathbf{x}_t,\mathbf{x}_t \setminus \{u\})} = \frac{p_n}{p_m}\frac{h(\mathbf{x}_t \setminus \{u\})}{h(\mathbf{x})}\frac{n(\mathbf{x})}{\lambda(U)}} \tag{2.37}$$

Ainsi donc, l'algorithme de Green, une fois les deux mouvements de naissance et de mort définis, simule une chaîne de Markov ayant pour distribution stationnaire $\pi(.)$. Il nous reste à nous intéresser

à l'irréductibilité et à la récurrence au sens de Harris.

▸ Irréductibilité de la chaîne

On suppose toujours que la densité objectif $\pi(.)$ possède une densité que l'on note $h(.)$ par rapport au processus de Poisson de référence.

Définition 2.32

> On appelle ***ensemble borné*** de l'espace des configurations $\mathbf{\Psi}$, tout ensemble A tel que
> $$\exists m : \forall \mathbf{x} \in A, n(\mathbf{x}) \leq m.$$

Le résultat suivant montre que l'algorithme de Green assure l'irréductibilité.

Proposition 2.33 *:* **Irréductibilité, Ensembles petits**

> Si la densité non normalisée $h(.)$ vérifie la condition de stabilité (2.11), alors l'algorithme de Green simule une chaîne de Markov irréductible et tout ensemble borné de $\mathbf{\Psi}$ est petit.

♢ **Démonstration :**
Montrons tout d'abord que la chaîne de Markov est irréductible. Notons $\mu_0(.)$ la mesure sur $\mathbf{\Psi}_0 = \{\emptyset\}$ qui donne une masse un au point $\{\emptyset\}$.

Nous pouvons minorer, grâce à (2.11), la probabilité d'accepter une mort :

$$min\left(1, \frac{p_n}{p_m}\frac{h(\mathbf{x} \setminus u)}{h(\mathbf{x})}\frac{n(\mathbf{x})}{\lambda(U)}\right) \geq \frac{p_n}{p_m}\frac{1}{M_h\lambda(U)}$$

où M_h est la constante introduite dans (2.11).

La probabilité de proposer une mort est p_m, et donc en partant de la configuration nulle, la probabilité de rester dans $\mathbf{\Psi}_0$ est $K(\{\emptyset\}, \mathbf{\Psi}_0) \geq p_m$.

Pour toute configuration $\mathbf{x} \in \mathbf{\Psi}$, $n(\mathbf{x})$ représente toujours son nombre d'objets. En choisissant un entier $m \geq n(\mathbf{x})$ on a :

$$\begin{aligned} K^m(\mathbf{x}, \mathbf{\Psi}_0) &\geq K^{n(\mathbf{x})}(\mathbf{x}, \mathbf{\Psi}_0) K^{m-n(\mathbf{x})}(\{\emptyset\}, \mathbf{\Psi}_0) \\ &\geq \left(\frac{p_n}{p_m}\frac{1}{M_h\lambda(U)}\right)^{n(\mathbf{x})} (p_m)^{m-n(\mathbf{x})} \\ &\geq \left(\frac{p_n}{p_m}\frac{1}{M_h\lambda(U)}\right)^{m} (p_m)^{m} \end{aligned}$$

si l'on choisit M_h assez grand pour que $\frac{p_n}{p_m}\frac{1}{M_h\lambda(U)} \leq 1$. Donc on obtient

$$K^m(\mathbf{x}, \mathbf{\Psi}_0) \geq \left(\frac{p_n}{M_h\lambda(U)}\right)^{m}. \tag{2.38}$$

(2.38) démontre bien que la chaîne de Markov est μ_0-irréductible. En effet, le seul ensemble μ_0-positif est $\mathbf{\Psi}_0 = \{\emptyset\}$, et dès que $m \geq n(\mathbf{x})$, $K^m(\mathbf{x},\mathbf{\Psi}_0) > 0$.

Ensuite, notons $\phi(.)$ une mesure que nous choisissons proportionnelle à $\mu_0(.)$: $\phi = a\mu_0$, avec $a = \left(\frac{p_n}{M_h \lambda(U)}\right)^m$. Soit $\mathcal{C}_m = \{\mathbf{x} \in \mathbf{\Psi} : n(\mathbf{x}) \leq m\}$. $\mathcal{C}_m$ est petit puisque

$$K^m(\mathbf{x},\mathbf{\Psi}_0) \geq a\mu_0(\mathbf{\Psi}_0) = a, \forall \mathbf{x} \in \mathcal{C}_m.$$

Ce qui démontre la deuxième partie de la proposition.

♢

▸ Récurrence de Harris et Ergodicité géométrique

Définition 2.34

Une chaîne de Markov est dite ***ergodique géométriquement***, s'il existe une constante $r > 1$ telle que $\forall \mathbf{x} \in \mathbf{\Psi}$:

$$\sum_{n=1}^{\infty} r^n \|K^n(\mathbf{x},.) - \pi(.)\| < \infty.$$

Un résultat de [23] nous donne une condition suffisante pour qu'une chaîne de Markov soit ergodique géométriquement.

Proposition 2.35 : Condition de drift géométrique

Soit une chaîne de Markov de noyau de transition $K(.,.)$. S'il existe une fonction $V : \mathbf{\Psi} \to [1,\infty)$, des constantes $a < 1$ et $b < \infty$, ainsi qu'un ensemble petit $\mathcal{C} \in \mathcal{B}(\mathbf{\Psi})$, tels que $\forall \mathbf{x} \in \mathbf{\Psi}$

$$KV(\mathbf{x}) = \int_{\mathbf{\Psi}} K(\mathbf{x},d\mathbf{y})V(\mathbf{y}) \leq aV(\mathbf{x}) + b\mathbf{1}_{\mathcal{C}}(\mathbf{x}), \tag{2.39}$$

alors la chaîne de Markov est ergodique géométriquement.

Le résultat suivant nous permettra de conclure sur la convergence en variations totales de la chaîne.

Proposition 2.36 : Récurrence de Harris, Ergodicité géométrique

Si la densité non normalisée $h(.)$ vérifie la condition de stabilité (2.11), alors l'algorithme de Green simule une chaîne de Markov récurrente au sens de Harris et ergodique géométriquement.

♢ **Démonstration :**
Nous pouvons tout d'abord, grâce à (2.11), minorer la probabilité d'accepter une mort et majorer celle d'accepter une naissance :

$$\begin{aligned} \alpha(\mathbf{x},\mathbf{x} \setminus u) &= min\left(1, \frac{p_n}{p_m} \frac{h(\mathbf{x} \setminus u)}{h(\mathbf{x})} \frac{n(\mathbf{x})}{\lambda(U)}\right) \\ &\geq min\left(1, \frac{p_n}{p_m} \frac{n(\mathbf{x})}{M_h \lambda(U)}\right) \end{aligned}$$

et

$$\begin{aligned}\alpha(\mathbf{x},\mathbf{x}\cup u) &= min\left(1,\frac{p_m}{p_n}\frac{h(\mathbf{x}\cup u)}{h(\mathbf{x})}\frac{\lambda(U)}{n(\mathbf{x})+1}\right)\\ &\leq \frac{p_m}{p_n}\frac{M_h\lambda(U)}{n(\mathbf{x})+1}.\end{aligned}$$

Pour tout $\epsilon \in (0,1)$, dès que $n(\mathbf{x}) \geq N_\epsilon = \frac{p_m}{p_n}\frac{M_h\lambda(U)}{\epsilon}$, on a :

$$\alpha(\mathbf{x},\mathbf{x}\setminus u) = 1$$

et

$$\alpha(\mathbf{x},\mathbf{x}\cup u) \leq \epsilon.$$

Dans ce cas, en posant $V(\mathbf{x}) = A^{n(\mathbf{x})}$, avec $A > 1$:

$$\begin{aligned}KV(\mathbf{x}) &= \left(\frac{p_m\alpha(\mathbf{x},\mathbf{x}\setminus u)}{A} + p_n\alpha(\mathbf{x},\mathbf{x}\cup u)A + p_m\left(1-\alpha(\mathbf{x},\mathbf{x}\setminus u)\right) + p_n\left(1-\alpha(\mathbf{x},\mathbf{x}\cup u)\right)\right)V(\mathbf{x})\\ &\leq \left(\frac{p_m}{A} + p_n\epsilon A + p_n(1-\epsilon)\right)V(\mathbf{x})\end{aligned}$$

On choisit alors ϵ assez petit pour que le terme entre parenthèses soit inférieur à 1, ce qui est possible puisque quand $\epsilon \to 0$, ce terme tend vers $\frac{p_m}{A} + p_n < 1$. Avec un tel ϵ, on trouve facilement une constante $a < 1$ telle que

$$KV(\mathbf{x}) \leq aV(\mathbf{x}),$$

dès que $n(\mathbf{x}) \geq N_\epsilon$.

Il suffit alors de choisir comme ensemble petit l'ensemble borné $\mathcal{C} = \{\mathbf{x} \in \mathbf{\Psi} : n(\mathbf{x}) < N_\epsilon\}$ (qui est petit d'après la proposition précédente), et comme constante b un majorant de $KV(\mathbf{x})$ sur cet ensemble $\mathcal{C}$, à savoir $b = A^{N_\epsilon+1}$.

$\Diamond$

▸ Conclusion

L'algorithme Metropolis-Hastings-Green réunit bien toutes les propriétés suffisantes à démontrer sa convergence en variations totales vers la mesure objectif $\pi(.)$, si la condition de stabilité (2.11) tient.

Une première simulation par cet algorithme MHG que nous avons menée, fut celle du processus de Poisson homogène décrit plus haut, et dont la Fig. (2.2) est l'une des réalisations. Nous pouvons remarquer sur la Fig. (2.8), que le nombre moyen d'objets du processus tend bien vers l'espérance (2.2) après un grand nombre d'itérations (algorithme arrêté après 5 millions d'itérations), ce qui est conforme à nos attentes.

2.3.3 Quelques pistes à explorer

L'algorithme de Green n'est bien entendu pas le seul permettant de simuler une chaîne de Markov, ne connaissant que la densité non normalisée de la loi objectif. De même, certaines améliorations de l'algorithme MHG sont envisageables. Cette partie expose différentes pistes, en cours d'exploration, qui pourront se révéler utiles à notre étude.

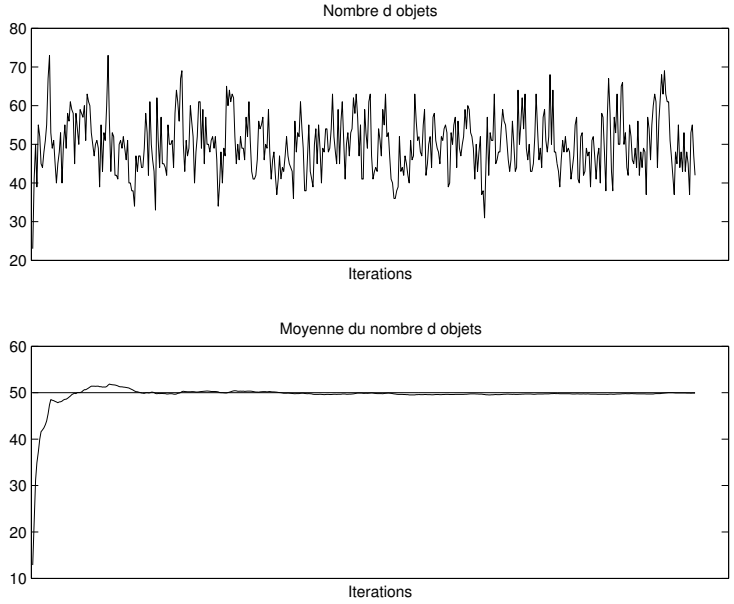

FIG. 2.8 – *Statistiques d'un processus de Poisson homogène, a=0.005.*

▸ Simulation parfaite

Ces méthodes de simulation nous donnent un critère d'évaluation de la convergence de la chaîne de Markov vers la loi objectif. Dans ses travaux, [21] et [22] en particulier, Lund applique ce modèle à la détection de sommets d'arbres.

De même, la question du contrôle de la convergence vers la loi objectif de notre algorithme de simulation n'a pas encore été abordée. Elle est développée dans plusieurs ouvrages, notamment [27], et mérite d'être étudiée profondément dans notre travail.

▸ Rejet reporté

Dans [12], Green et Mira introduisent une variante de l'algorithme MHG. Supposons que nous sommes dans l'état $\mathbf{x}_t$, et qu'une proposition de mouvement vers un état $\mathbf{y}$ vient d'être rejetée. L'algorithme MHG, dans sa forme classique, consisterait à oublier ce rejet, et à proposer un nouveau mouvement à partir de $\mathbf{x}_t$. Ceci a pour conséquence d'augmenter l'autocorrélation de la chaîne de Markov, ainsi que le temps de convergence.

Dans leur nouvelle approche, Green et Mira, après le rejet du premier mouvement, nous suggèrent d'effectuer un nouvel essai, selon une distribution qui peut dépendre du premier état rejeté. Cette seconde proposition augmente la probabilité de quitter l'état courant. L'article de Green et Mira nous explique comment s'assurer de la réversibilité de la chaîne malgré tout, propriété indispensable pour la convergence.

Typiquement, cela peut servir dans le cas où l'on se retrouve bloqué dans un état, et que le noyau de proposition ne nous permet d'en sortir qu'avec une trop faible probabilité. L'information contenue dans les premiers rejets peut se révéler utile au calcul de nouvelles distributions de noyaux de proposition, et augmenter la dynamique de la chaîne.

CHAPITRE

3 Notre modèle pour la détection des houppiers

Comme annoncé dans la partie précédente, nous allons nous placer dans le cadre de processus objet simulé par l'algorithme de Green. Plusieurs étapes sont nécessaires à la définition de notre modèle géométrique.

Tout d'abord, il convient de spécifier les objets d'intérêt sur lesquels nous allons travailler, et qui composeront les configurations que nous rechercherons dans notre image. Puis, nous étudierons quelle loi objectif nous devons donner à la chaîne de Markov, afin d'obtenir les résultats escomptés. Enfin, les mouvements du noyau de proposition seront décrits, et nous aborderons le problème de la convergence de l'algorithme.

Nous veillerons à conserver les notations introduites au cours des parties précédentes, sauf avis contraire.

3.1. Analyse des données : définition des objets d'intérêt

Pour observer des objets tels que les arbres, une résolution submétrique est indispensable. En effet, l'ordre de grandeur de la dimension transversale (horizontale) variant de quelques mètres pour certain, à une dizaine de mètres pour d'autres, il est essentiel de travailler sur de tels supports afin d'obtenir une image de qualité visuelle suffisante.

Les données sur lesquelles tout notre travail repose seront donc des images aériennes dites ***Très Haute Résolution*** (THR).

3.1.1 Des sources d'images Très Haute Résolution

Les nouvelles données numériques de télédétection THR ouvrent des perspectives nouvelles dans la connaissance des arbres et des forêts, que n'ont pas été capables d'explorer les satellites d'observation à résolution décamétrique qui opéraient jusqu'alors.

L'évolution du contexte technologique nous permet aujourd'hui d'envisager l'acquisition de données THR, par le biais du scannage de photographies aériennes analogiques, de l'utilisation de caméras numériques 4.000 * 4.000 pixels (opérationnelles à l'IGN), ou de satellites THR. Ces derniers ont

récemment fait leur apparition sur le marché par le biais d'Ikonos (résolution de 1m), et sont en plein développement (futurs satellites Pléiades, de résolution 0,7m) .

▸ Images satellitaires

Auparavant, les satellites nous offraient des images de résolution 30m (Landsat ETM), 20m (SPOT XS), 15m (Landsat Pan), 10m (SPOT Pan), ou 5,8m (IRS-ID, codage sur 6 bits seulement, au lieu de 8 bits pour les autres). Aussi les objets identifiés (surfaces d'eau, champs, forêts) avaient-ils des dimensions trop élevées pour pouvoir étudier des arbres isolés, ainsi que la structure interne (densité des arbres, espacement) des forêts.

A l'heure actuelle, très rares sont les satellites commerciaux pouvant nous offrir une résolution submétrique. Sur le marché, on trouve aujourd'hui le satellite américain Ikonos, à un mètre de résolution en mode panchromatique. Il s'agit du plus utilisé en haute résolution. On peut également citer le satellite Eros A, qui atteint 1.8m de résolution, toujours en mode panchromatique. Mais dans un futur proche, de grands espoirs sont placés sur les satellites Pléiades. Pour avoir un ordre d'idées de ce qui se fait, voici la gamme proposée par Spot Image en 2002 :

Produits Spot Image	Satellite	Nombre de bandes	Résolution
2.5m couleurs	Spot 5	3 bandes	2.5m
2.5m noir et blanc	Spot 5	1 bande	2.5m
5m couleurs	Spot 5	3 bandes	5m
5m noir et blanc	Spot 5	1 bande	5m
10m couleurs	Spot 4 et 5	3 bandes	10m
10m noir et blanc	Spot 1 à 4	1 bande	10m
20m couleurs	Spot 1 à 4	3 bandes (4 pour Spot 4)	20m

TAB. 3.1 – *Produits SPOT en juillet 2002.*

D'après cette table, la plus fine image que puisse aujourd'hui fournir Spot 5 est une image de résolution 2.5m, couleurs ou noir et blanc. Ces produits nécessitent la création d'une image 2.5m, elle-même obtenue après fusion de deux images panchromatiques de résolution 5m acquises simultanément. Pour la couleur, il faut en outre une image en mode multispectral (3 bandes : typiquement vert, rouge, et proche infra-rouge) de résolution 10m, ce qui fait 3 images à acquérir.

Le mode panchromatique se caractérise par une observation réalisée dans une seule bande spectrale située dans la partie visible du spectre électromagnétique. La bande panchromatique de Spot 5 est comprise entre 0.48μm et 0.71μm, ce qui représente une largeur de bande deux fois plus élevée que celle des 3 bandes des images multispectrales.

Nous nous retrouvons face à un problème physique de tout premier plan, les capteurs ne pouvant pas indéfiniment gagner en présision sur le domaine spatial sans perdre en présision sur le domaine fréquentiel.

▸ Images aériennes

Les photographies aériennes qui nous serviront de base dans notre étude, et sur lesquelles travaillent les experts de l'***Inventaire Forestier National*** (IFN), sont des clichés analogiques de 23cm × 23cm, que l'on peut scanner jusqu'à une précision de 50cm au sol.

C'est d'ailleurs la résolution qui est atteinte pour nos images, scannées avec une définition de 600 dpi (dots per inch), et desquelles on extrait des zones de 576 × 480 pixels, qui représentent sur le

terrain un espace rectangulaire de 311m sur 259m (et donc 0.54m par pixel).

Les 3 bandes de fréquences sont situées entre 520 et 900 νm, le bleu est donc filtré. Une des bandes spectrales des capteurs aériens est la bande proche infra-rouge (700 à 900 νm), qui fait ressortir sur l'image les chlorophylliens, c'est à dire les feuillus. Ceux-ci auront en effet une teinte plus rouge, alors que les résineux auront une teinte plus foncée. Les deux autres sont de 520 à 620 νm (vert-jaune) et de 620 à 700 νm (rouge).

La Fig. (3.1) est la première donnée réelle sur laquelle nous avons testé nos algorithmes. C'est une zone de forêts se trouvant en Saône et Loire. La Fig. (3.2) est extraite de cette image. A droite de celle-ci se trouve une zone de peupliers sur laquelle nous allons travailler. A gauche, on a une zone dire de taillis sous futaie, c'est à dire un étage supérieur, la futaie, composée ici de chênes principalement, et un étage inférieur, sans doute composé de charmes et de chênes. Cette structure étagée est classique dans nos forêts.

3.1.2 Objectifs fixés

▸ Aide apportée aux experts

En collaboration avec l'IFN, cette étude sur la détection des houppiers devra mettre en évidence des processus automatiques d'interpétation d'images aériennes de forêts.

Bien que l'objectif soit d'automatiser ces traitements, la présence d'un expert restera de toute évidence indispensable, tant pour donner des renseignements utiles à l'algorithme, comme par exemple lui apporter des connaissances géologiques du terrain (types d'arbres que l'on peut y trouver, pente, ...), que pour valider ses résultats.

Aussi, notre objectif à long terme sera avant tout d'aider les experts forestiers, et de leur faire gagner un temps précieux dans leur travail de reconnaissance sur ordinateur.

▸ Différents paramètres

Plusieurs paramètres, naturels pour la plupart, sont à prendre en considération au cours de notre étude. Ceux-ci jouent en effet un rôle capital dans l'interprétation des images aériennes, et nous devront nous adapter spécifiquement à chacun d'eux dans nos algorithmes.

Il s'agit principalement :

1. des espèces d'arbres distinctes présentes sur le cliché. Nous serons en effet amenés à étudier des images pouvant contenir différentes espèces d'arbres. Mais la variabilité intraspécifique (variation de tailles, de forme, liées à l'âge, au sein d'une espèce) n'est pas pour autant à négliger, elle est même toute aussi importante que celle interspécifique (entre espèces) selon les experts.

2. du nombre d'arbres, ainsi que de la densité de la zone de forêt étudiée.

3. de l'illumination.

4. de la pente du terrain, qui avec l'illumination peut créer des jeux d'ombres difficiles à faire interpréter par un algorithme automatique.

FIG. 3.1 – *Image aérienne d'un peuplement de Saône et Loire ©IFN.*

FIG. 3.2 – *Image de Saône et Loire : à gauche des taillis sous futaies principalement composés de chênes, à droite des peupliers ©IFN.*

A long terme, nous espérons être en mesure de traiter des images faisant intervenir des variations de ces paramètres, c'est à dire de créer un algorithme sachant s'adapter à différentes illuminations, différentes prises de vue, et différentes espèces d'arbres.

▸ Premier champ d'investigation : les peupleraies

Dans un premier temps, il a été convenu avec l'IFN de se concentrer sur une première espèce d'arbres : les peupliers.

Les peupliers recouvrent quelques 200.000 hectares du territoire français, regroupés en zones que l'on nomme des ***peupleraies***. Ils poussent typiquement dans les fonds des vallées, les pieds dans l'eau, sur des terrains par conséquent plats.

De plus, la connaissance a priori des peupleraies est riche. La plupart du temps, ces arbres plantés par l'homme suivent en effet un certain alignement, et sont espacés régulièrement. Leur espacement est tel que la plupart du temps, on observe un espace vide entre deux arbres sur les images. Cette information se révèlera très utile dans la mise en place de notre modèle.

Il s'agit donc d'un cas assez simple, du moins en comparaison avec d'autres, sur lequel nous engageons nos séries de tests. Une fois l'étude des peupliers bien avancée, au cours de la thèse qui suivra ce stage de DEA, nous pourrons aborder d'autres situations plus complexes.

3.1.3 Objets d'intérêt

▸ Approche objet

L'approche objet prend tout son sens dans les méthodes de détection de houppiers que nous utilisons : les processus objet. En effet, les objets du processus que nous simulerons seront justement l'objet arbre.

Plusieurs niveaux d'étude sont alors envisageables : de l'arbre à la forêt, en passant par les bosquets. Toute l'information que nous pourrons donner sur les bosquets d'arbres contenus dans la forêt, ainsi que sur les arbres les composant, sera intéressante.

▸ Modèle géométrique

Sur la partie droite de l'image de la Fig. (3.2), on s'aperçoit que la forme géométrique des peupliers est très difficile à déterminer avec exactitude, d'autant plus qu'elle varie d'un arbre à l'autre. Le choix du modèle géométrique de nos objets est donc une première décision d'importance à prendre.

Plusieurs possibilités de modélisation de nos objets s'offrent à nous :

1. forme polygonale : on pourrait tout à fait modéliser chaque arbre par un polygone, d'un nombre indéterminé de côtés. Cependant, étant donné qu'un peuplier ne représente sur l'image qu'une dizaine de pixels, ce serait inutilement se compliquer la tâche, pour une amélioration peu sensible. En outre, l'espace objet serait alors difficilement identifiable.

2. forme elliptique : dans ce cas, l'objet est caractérisé par sa position, la dimension du grand axe, la dimension du petit axe, ainsi que son orientation.

3. forme circulaire : cas particulier du précédent, celui-ci a pour avantage de ne faire entrer en jeu que trois variables pour caractériser chacun des objets, puisque l'orientation n'existe plus, et que les deux dimensions sont les mêmes.

Nous avons opté dans un premier temps, pour une modélisation circulaire des peupliers. Ainsi, chaque objet est parfaitement caractérisé par la position de son centre dans l'image (X_c,Y_c) et par son rayon R_c. L'espace objet sur lequel nous travaillerons sera par conséquent un sous-ensemble de $\mathbb{R}^3$ qu'il faudra déterminer.

▸ L'espace objet U

▷ Position

Les images de données étant finies, les coordonnées des objets seront inscrites dans un pavé de $\mathbb{R}^2$. On note $\mathcal{P}$ cet espace des positions : $\mathcal{P} = [0,X_M] \times [0,Y_M]$.

▷ Marques

Il en sera de même pour les marques des objets, c'est à dire les rayons des cercles. On peut raisonnablement supposer qu'aucun arbre n'aura une dimension supérieure à R_M, ni inférieure à R_m (pouvant être pris égal à 0 si nous n'avons aucune information supplémentaire).

L'espace des marques, noté $\mathcal{K}$, est alors simplement $\mathcal{K} = [R_m,R_M]$.

L'espace objet a donc la forme suivante :

$$U = \mathcal{P} \times \mathcal{K} = [0,X_M] \times [0,Y_M] \times [R_m,R_M].$$

Pour définir un élément u de U, on écrira $u = (X_u,Y_u,R_u)$.

L'espace des configurations $\mathbf{\Psi}$ est défini de la même façon que dans la partie précédente, à partir de notre espace objet U.

3.2. Modèle de la loi objectif

3.2.1 Cadre Bayésien

▸ Processus ponctuel de référence

Le processus ponctuel de référence, conformément à notre exposé de la partie précédente, sera de Poisson de mesure d'intensité $\lambda(.)$. Cette mesure $\lambda(.)$ sur l'espace objet sera la mesure de Lebesgue définie sur $\mathbb{R}^3$.

La loi de ce processus sera notée $\mu(.)$. Nous rappelons que, par définition d'un processus objet, il doit être processus ponctuel sur l'espace U ainsi que sur l'espace des positions. De plus, la loi $\mu(.)$ s'écrit :

$$\mu(B) = e^{-\lambda(U)} \left(\mathbf{1}_{[\emptyset \in B]} + \sum_{n=1}^{\infty} \frac{\mu_n(B)}{n!} \right)$$

avec :

$$\mu_n(B) = \int \dots \int \mathbf{1}_{[\{u_1,\dots,u_n\} \in B]} \lambda(du_1) \dots \lambda(du_n).$$

▸ Densité à simuler

Soit $h(.)$ la densité du processus ponctuel que l'on souhaite simuler. La mesure objectif $\pi(.)$, mesure invariante de la chaîne de Markov, s'écrira alors :

$$\pi(d\mathbf{x}) = h(\mathbf{x})\mu(d\mathbf{x}).$$

L'expression de la densité $h(.)$ est assurément un des points-clés de notre démarche. Non seulement elle doit contenir les informations suffisantes nous permettant de détecter les houppiers sur les images de forêts, mais aussi elle doit vérifier la condition de stabilité (2.11) que l'on rappelle ici :

$$h(\mathbf{x} \cup \xi) \leq M_h \, h(\mathbf{x}), \forall \mathbf{x} \in \mathbf{\Psi}, \forall \xi \in U.$$

Sous cette hypothèse, ainsi que celle de réversibilité, nous sommes assurés de la convergence de la chaîne de Markov vers la mesure objectif $\pi(.)$.

▸ Densité a posteriori

La densité $h(\mathbf{x})$ modélise en réalité la densité de probabilité de la configuration $\mathbf{x}$, étant donnée l'image $\mathcal{I}$ de départ. Nous l'écrivons $h(\mathbf{x})$:

$$h(\mathbf{x}) = h(\mathbf{x}|\mathcal{I}).$$

Cette écriture est celle de la densité a posteriori des configurations. Dans un cadre Bayésien, on peut alors appliquer la règle de Bayes :

$$h(\mathbf{x}) = \frac{h_p(\mathbf{x})\mathcal{L}(\mathcal{I}|\mathbf{x})}{f(\mathcal{I})} \propto h_p(\mathbf{x})\mathcal{L}(\mathcal{I}|\mathbf{x}). \tag{3.1}$$

L'écriture Bayésienne est très robuste mathématiquement. Cependant elle nécessite d'être capable d'exprimer la vraisemblance de l'image étant donnée la configuration $\mathbf{x}$, $\mathcal{L}(\mathcal{I}|\mathbf{x})$, c'est à dire qu'à partir d'une configuration d'objets nous devons être en mesure de reconstituer une image de données. Le terme $h_p(\mathbf{x})$ est quant à lui une connaissance a priori sur les configurations.

Nous rechercherons en fait la "meilleure" configuration d'objets possible sur notre image $\mathcal{I}$. Par la suite, nous expliquerons comment nous avons choisi d'interpréter ce terme "meilleur". Nous verrons que dans notre approche, cela consiste à chercher le maximum a posteriori

$$\mathbf{x}^*(\mathcal{I}) = arg\ \max_{\mathbf{x}}(h(\mathbf{x}))$$

c'est à dire la configuration maximisant la densité a posteriori $h(.)$.

3.2.2 Terme a priori

Ce terme renseigne sur toutes les connaissances a priori sur les configurations que l'on recherche. Par exemple, on peut y inclure un terme répulsif de Strauss, afin de s'affranchir des configurations comportant des objets qui se chevauchent.

Une bonne définition des contraintes que l'on souhaite imposer aux configurations nous permettrait d'accélérer la convergence de l'algorithme: au cours des itérations, nous n'accepterions en effet que les configurations intéressantes au sens de cet a priori et pourrions nous retrouver plus rapidement dans un mode de la densité a posteriori. On peut répertorier plusieurs contraintes dans notre exemple.

▷ **Superposition des objets**

Partant du principe que l'un de nos objectifs est de dénombrer les arbres, il faut éviter de compter deux fois le même.

Ainsi, une configuration comportant deux cercles se superposant, et détectant le même houppier, est donc à bannir. Par contre, une configuration avec deux cercles se superposant en partie, et détectant deux houppiers distincts, doit être possible si nous étudions une image oà les arbres sont trs proches les uns des autres. Ces deux cas de figure sont représentées sur la Fig. (3.3), réalisée à partir d'une image synthétique.

Nous devons donc veiller à ce que deux cercles de notre configuration ne se superposent pas trop. Deux pistes sont alors envisageables. D'un côté nous pouvons inclure dans le terme a priori une répulsion de Strauss sur deux objets dont les supports s'intersectent:

$$h_{strauss}(\mathbf{x}) = e^{\gamma_{strauss}.s(\mathbf{x})}$$

avec $s(\mathbf{x})$ le nombre de paires d'objets dans un tel cas. La constante γ sera évidemment choisie négative puisque nous souhaitons instaurer une répulsion entre les objets.

Cette première hypothèse se révèle par moments trop forte. Sur nos images d'arbres nous avons parfois deux houppiers qui s'intersectent légèrement, et nous voudrions les détecter sans être pénalisé par le terme de Strauss pour autant. Une autre alternative est un terme d'intéraction d'aires, mais qui ne s'intéresserait qu'à l'aire en commun entre deux cercles d'une configuration:

$$h_{aire}(\mathbf{x}) = e^{\gamma_{aire}.\lambda_2(\mathcal{S}(\mathbf{x}))}$$

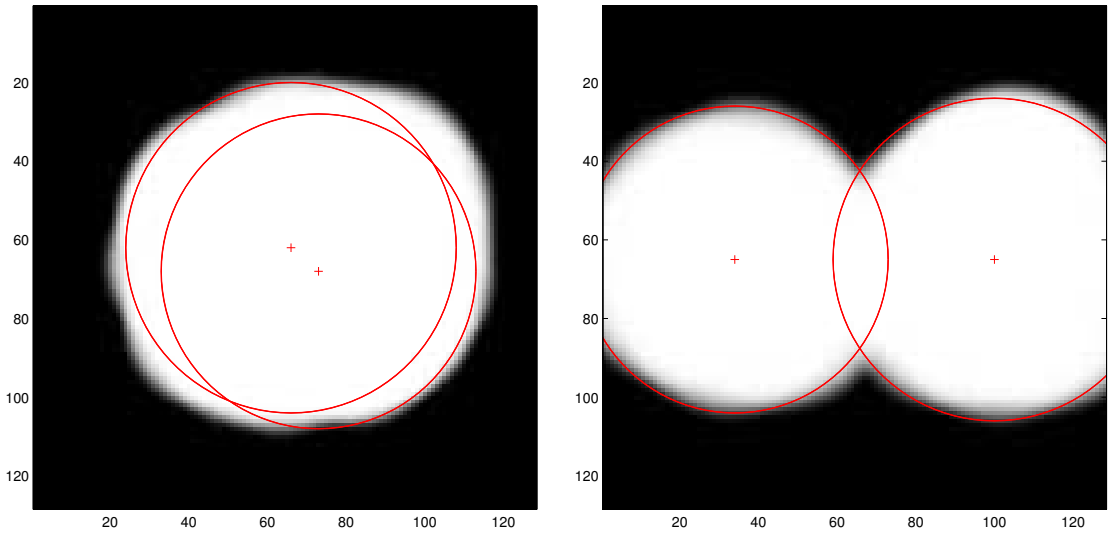

FIG. 3.3 – *A gauche : Cas de superposition à éviter, les deux cercles détectant le même objet. A droite : Superposition acceptable, deux objets distincts étant délimités.*

avec $\lambda_2(.)$ la mesure de Lebesgue sur $\mathbb{R}^2$, et $\mathcal{S}(\mathbf{x})$ l'union des intersections des supports des objets de $\mathbf{x}$. Une fois de plus, le coefficient γ est choisi négatif pour les mêmes raisons qu'auparavant.

▷ **Alignement**

Plantés par l'homme, les peupleraies observent la plupart du temps un alignement et un espacement régulier dans deux directions orthogonales. Nous pouvons le remarquer par exemple dans la Fig. (3.2), même si certains trous dans la zone sont présents. Aussi, une transformée de Fourier appliquée à l'image pourrait nous renseigner sur les fréquences fondamentales des positions des arbres.

Nous ne pouvons pas nous limiter au calcul de cette transformée de Fourier pour détecter les arbres, puisque d'une part par endroit il y a des zones vides, et d'autre part tous les arbres ne respectent pas cet alignement régulier. Cependant, nous pouvons favoriser dans le terme a priori les configurations présentant une telle propriété.

Dans notre premier modèle, nous n'avons pas inclus ce terme dans la densité a priori.

▸ Bilan

Au final, $h_p(.)$ s'écrit :

$$h_p(\mathbf{x}) = (\epsilon.h_{strauss}(\mathbf{x}) + (1-\epsilon).h_{aire}(\mathbf{x})) \tag{3.2}$$

avec ϵ valant 0 ou 1, suivant le type de répulsion que l'on souhaite instaurer entre les objets.

3.2.3 Terme d'attaches aux données

Le terme de vraisemblance $\mathcal{L}(\mathcal{I}|\mathbf{x})$ correspond à la probabilité, étant donnée une configuration d'objets $\mathbf{x}$, d'observer une image $\mathcal{I}$.

Travaillant sur des images discrètes dans [31] et [32], H. Rue fait de l'inférence bayésienne à l'aide de processus ponctuels simulés par des chaînes de Markov. Son terme de vraisemblance est intéressant,

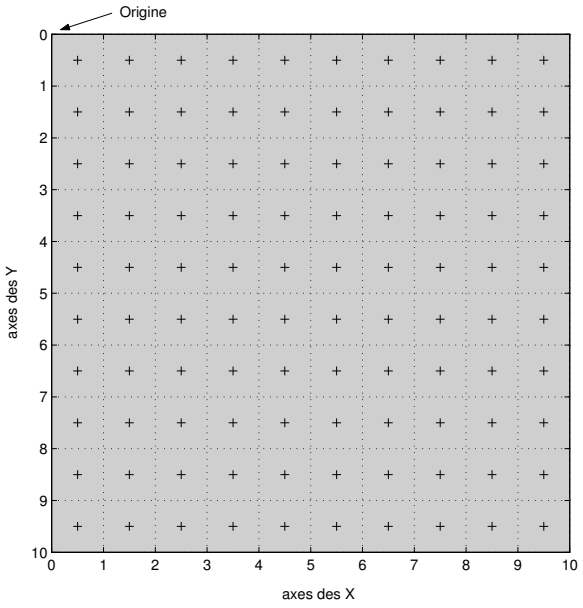

FIG. 3.4 – *Repère de l'espace des positions sur la matrice d'une image $\mathcal{I}$ de taille* 10×10. *Le quadrillage sépare les pixels de l'image $\mathcal{I}$, et les croix représentent les centres de chacun des pixels.*

puisqu'il distingue deux classes de pixels dans l'image, selon qu'ils sont ou non contenus dans un des objets de son processus, chacune des classes ayant des propriétés distinctes.

Nous utiliserons la même démarche pour notre attache aux données, qui sera une attache au niveau pixélique : nous regarderons si tous les pixels ont correctement été classifiés. Se posent alors le problème de la discrétisation, ainsi que celui de la formulation de cette vraisemblance.

▸ Problème de la discrétisation

L'image de données étant discrete, et les objets de la configuration définis sur un espace U continu (plus exactement, c'est la continuité de l'espace des positions qui nous intéresse ici), il nous faut définir une relation d'attache entre un objet défini sur U, et son représentant sur l'image.

Dans la pratique, l'espace des positions est $\mathcal{P} = [0,X_M] \times [0,Y_M]$, avec X_M et Y_M entiers, correspondant à la largeur et la hauteur de l'image.

On peut visualiser l'ensemble de ces positions sur l'image en se munissant d'un repère orthonormé avec pour origine le coin en haut à gauche. Ainsi donc, le pixel $p = \{i,j\}$, $0 \leq i < X_M, 0 \leq j < Y_M$, de la i^{eme} colonne et de la j^{eme} ligne, recouvre le sous-ensemble $[i,i+1] \times [j,j+1]$ de $\mathcal{P}$.

Définition 3.1

> Soit p un pixel $\{i,j\}$. Le ***centre du pixel*** p est le point central de p dans l'espace des positions $\mathcal{P}$, c'est à dire $(i + \frac{1}{2},j + \frac{1}{2})$.

Toutes ces notations sont reprises dans le schéma récapitulatif de la Fig. (3.4).

Soit alors $\mathbf{x} = (x_1, \ldots, x_n)$ une configuration d'objets. Pour chacun de ses composants x_i, on peut alors définir la notion de représentant sur l'image.

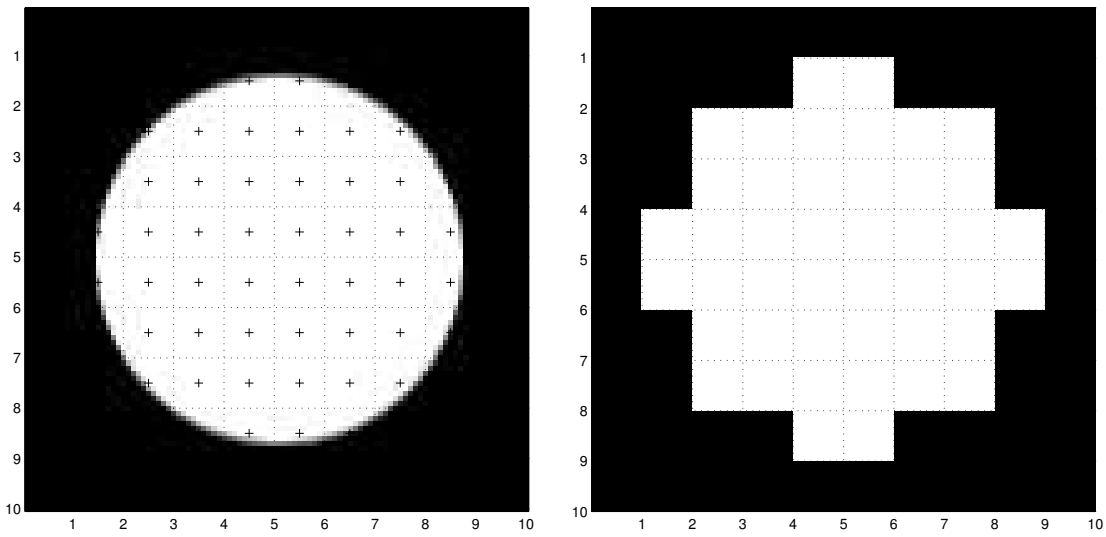

FIG. 3.5 – *A gauche : objet dans l'espace continu des positions, placé sur la grille de l'image. A droite : représentant de l'objet.*

Définition 3.2

On appelle ***représentant sur l'image*** $\mathcal{I}$ de $x_i = (X_{x_i}, Y_{x_i}, R_{x_i})$, l'ensemble $\mathcal{T}_{\mathcal{I}}(x_i)$ des pixels de $\mathcal{I}$ ayant leur centre dans le support de x_i.
Si les x_i sont des cercles, ces pixels vérifient l'inégalité suivante :

$$\sqrt{(i+\frac{1}{2}-X_{x_i})^2+(j+\frac{1}{2}-Y_{x_i})^2} \leq R_{x_i} \Longrightarrow (i,j) \in \mathcal{T}_{\mathcal{I}}(x_i).$$

La Fig. (3.5) donne un exemple d'objet circulaire et de son représentant sur l'image. La discrétisation des données, notamment sur des objets de petite taille comme des arbres (sur une image de résolution 0.5m), pose donc de nombreuses difficultés quant à leur représentation dans le domaine continu. Nous nous sommes limités à des cercles dans un premier temps, mais sans doute qu'une forme elliptique sera plus adaptée dans nos applications futures.

▸ Définition des classes de pixels

On peut alors attribuer à chacun des pixels de l'image $\mathcal{I}$ une appartenance à une des deux classes suivantes :

1. La classe des arbres $\mathcal{C}_A$: un pixel appartient à $\mathcal{C}_A$ si il appartient à un des représentants $\mathcal{T}_{\mathcal{I}}(x_i)$ des objets de la configuration $\mathbf{x}$:

$$\mathcal{C}_A = \bigcup_{x_i \in \mathbf{x}} \mathcal{T}_{\mathcal{I}}(x_i).$$

2. La classe du fond $\mathcal{C}_F$: un pixel appartient à cette classe si il n'appartient pas à la classe des arbres.

Autrement dit, nous avons réalisé une partition de l'image discrète en deux ensembles de pixels.

▸ Modélisation des classes

Dans le terme de vraisemblance de l'expression de la densité a posteriori, nous devons quantifier la vraisemblance d'une image étant donnée une configuration d'objets. Se reportant aux travaux de Rue, nous avons émis l'idée de calculer une vraisemblance sur chacun des pixels de l'image.

A partir d'une configuration $\mathbf{x}$, nous avons introduit précédemment les notions suffisantes pour obtenir une classification de tous les pixels, dans l'une des classes $\mathcal{C}_A$ ou $\mathcal{C}_F$. Se pose désormais la question essentielle : comment juger si cette classification est bonne ?

En observant nos images, on remarque que la bande infra-rouge sépare nettement le niveau de gris des arbres et du fond. Dans une première approche, on décide alors de modéliser chacune de ces classes par une distribution normale de ses niveaux de gris :

$$\mathcal{C}_A : \mathcal{N}(\mu_A,\sigma_A), \mathcal{C}_F : \mathcal{N}(\mu_F,\sigma_F),$$

Ainsi, pour un pixel $p \in \mathcal{C}_A$, de niveau de gris y_p sur $\mathcal{I}$, la vraisemblance de cette classification est

$$P(p \in \mathcal{C}_A) = \frac{1}{\sqrt{2\pi}\sigma_A} exp\left(\frac{-(y_p - \mu_A)^2}{2\sigma_A^2}\right).$$

Si ce même pixel était classifié dans $\mathcal{C}_F$, sa vraisemblance serait

$$P(p \in \mathcal{C}_F) = \frac{1}{\sqrt{2\pi}\sigma_F} exp\left(\frac{-(y_p - \mu_F)^2}{2\sigma_F^2}\right).$$

Ce modèle pourrait tout à fait être développé si on avait plusieurs espèces d'arbres différentes. Cependant, les distributions gaussiennes de ces niveaux de gris risqueraient de se confondre, et l'évaluation de la vraisemblance serait faussée puisqu'un pixel risquerait d'être vraisemblable pour plusieurs classes.

L'information donnée par les niveaux de gris est donc très faible. Il nous faudra par la suite, lorsque nous étudierons des images comportant différentes espèces d'arbres, établir une signature robuste de chacune de ces espèces, en calculant par exemple certains ***attributs de texture***.

▸ Formulation de la vraisemblance

La vraisemblance de l'image $\mathcal{L}(\mathcal{I}|\mathbf{x})$, puisque l'on suppose les pixels indépendants conditionnellement à leur appartenance à une classe donnée, est donc :

$$\mathcal{L}(\mathcal{I}|\mathbf{x}) = \prod_{p\in\mathcal{C}_A} \frac{1}{\sqrt{2\pi}\sigma_A} exp\left(\frac{-(y_p - \mu_A)^2}{2\sigma_A^2}\right) \prod_{p\in\mathcal{C}_F} \frac{1}{\sqrt{2\pi}\sigma_F} exp\left(\frac{-(y_p - \mu_F)^2}{2\sigma_F^2}\right). \qquad (3.3)$$

3.2.4 Stabilité du processus

Un terme de contrôle sur l'intensité du processus, noté $\beta^{n(\mathbf{x})}$, vient se rajouter à la mesure du processus de Poisson de référence. Il aura une influence sur l'espérance du nombre de points de la configuration finale. La loi objectif $\pi(.)$ s'exprime alors :

$$\pi(d\mathbf{x}) = h(\mathbf{x})\beta^{n(\mathbf{x})}\mu(d\mathbf{x}). \tag{3.4}$$

Suite aux choix de la densité a priori et de la vraisemblance, nous devons vérifier si la condition de stabilité (2.11) tient.

Pour cela, remarquons d'une part que la vraisemblance $\mathcal{L}(\mathcal{I}|\mathbf{x})$ est majorée et minorée par $\mathcal{L}_{min}$ et $\mathcal{L}_{max}$ définis de la façon suivante :

$$\mathcal{L}_{min} = \prod_{p\in\mathcal{C}_A\cup\mathcal{C}_F} \frac{1}{\sqrt{2\pi}\max(\sigma_A,\sigma_F)} exp\left(\frac{-\max\left(y_p-\mu_A, y_p-\mu_F\right)^2}{2\min(\sigma_A,\sigma_F)^2}\right) > 0 \tag{3.5}$$

et

$$\mathcal{L}_{max} = \prod_{p\in\mathcal{C}_A\cup\mathcal{C}_F} \frac{1}{\sqrt{2\pi}\min(\sigma_A,\sigma_F)} exp\left(\frac{-\min\left(y_p-\mu_A, y_p-\mu_F\right)^2}{2\max(\sigma_A,\sigma_F)^2}\right) > 0. \tag{3.6}$$

D'autre part, l'intensité conditionnelle de Papangelou du terme a priori $h_p(.)$ est majorée :

$$\frac{h_p(\mathbf{x}\cup\{\xi\})}{h_p(\mathbf{x})} \leq 1, \forall\xi \in \{U - \bigcup_{i=1}^{n(\mathbf{x})} x_i\}. \tag{3.7}$$

En conclusion de (3.5), (3.6) et (3.7), on obtient

$$\frac{h(\mathbf{x}\cup\{\xi\})}{h(\mathbf{x})} \leq \frac{\mathcal{L}_{max}}{\mathcal{L}_{min}} = M_h, \forall\xi \in \{U - \bigcup_{i=1}^{n(\mathbf{x})} x_i\}$$

ce qui nous assure la stabilité locale du processus, et donc les bonnes propriétés de la chaîne de Monte Carlo qui le simule.

3.2.5 Estimateurs

Trouver une estimée de la configuration d'objets qui correspond le mieux à notre image, c'est déjà se donner une fonction de coût sur l'espace des configurations. L'estimée est alors la configuration qui minimise l'espérance du coût suivant la loi a posteriori.

▸ Estimateur de Bayes

Soit $\mathcal{I}$ une image, toute la connaissance que l'on détient sur la configuration d'objets optimale sur cette image est contenue dans la densité a posteriori $h(\mathbf{x}) = h(\mathbf{x}|\mathcal{I})$.

Soit $\mathcal{L}(\mathbf{x},\hat{\mathbf{x}})$ une fonction de coût, qui mesure la perte dûe à l'estimation de la configuration optimale par $\hat{\mathbf{x}}$. On pose classiquement $\mathcal{L}(\mathbf{y},\mathbf{y}) = 0$, et $\mathcal{L}(\mathbf{y},\mathbf{z}) > 0$ si $\mathbf{y}$ est différent de $\mathbf{z}$.

Une fois cette fonction définie, on peut obtenir l'***estimateur de Bayes*** $\mathbf{x}^*$ en minimisant l'espérance a posteriori de ce coût appelé ***risque de Bayes*** :

$$\mathbf{x}^* = arg \min_{\hat{\mathbf{x}}} \left(\mathbb{E}_{\mathbf{x}|\mathcal{I}} [\mathcal{L}(\mathbf{x},\hat{\mathbf{x}})] \right) \tag{3.8}$$

Un premier estimateur est le ***maximum a posteriori***. Cependant, nous verrons que ce n'est pas le seul, et que d'autres sont plus adaptés pour les problèmes de traitement d'images.

▸ Maximum a posteriori

▷ Calcul de l'estimateur

La fonction de coût qui lui est associée est :

$$\mathcal{L}_{MAP}(\mathbf{x},\hat{\mathbf{x}}) = \begin{cases} 0 & \text{si } \mathbf{x} = \hat{\mathbf{x}}; \\ 1 & \text{sinon.} \end{cases} \tag{3.9}$$

L'estimateur de Bayes associé, $\mathbf{x}_{MAP}$, solution du problème de minimisation (3.8), est donc la configuration qui maximise la densité a posteriori :

$$\begin{aligned} \mathbf{x}_{MAP} &= arg \max_{\hat{\mathbf{x}}} (h(\mathbf{x}|\mathcal{I})) \\ &= arg \max_{\hat{\mathbf{x}}} (h(\mathbf{x})) \,. \end{aligned}$$

▷ Recuit simulé

Dans la pratique, pour obtenir la configuration qui maximise la densité a posteriori $h(.)$, on utilise un ***recuit simulé***. Son principe est de simuler une chaîne de Markov de densité objectif non plus $h(.)$, mais $h(.)^{\frac{1}{T_k}}$, où T_k décroit vers 0 quand le nombre d'itérations augmente.

Ainsi, cela revient à faire évoluer $h(.)$ en dirac au niveau de ses modes principaux. La décroissance vers 0 de la température nous assure donc de trouver le maximum de la densité a posteriori.

Cependant, de fortes conditions sur la vitesse de décroissance des températures T_k sont nécessaires pour s'assurer de la convergence de l'algorithme de recuit simulé. Les travaux de D. et S. Geman, ainsi que de B. Hajek, montrent qu'une décroissance logarithmique

$$T_k = \frac{a}{\ln(k+1)}$$

convient pour assurer la convergence d'un algorithme de type Metropolis (ce qui est notre cas) par recuit simulé.

Cependant, une telle décroissance est trop lente pour être implanté : aussi, lors des simulations sur ordinateur, nous lui préférons une décroissance géométrique, ce qui nous fait sortir du cadre théorique de la convergence, mais donne en pratique donne des résultats corrects.

$$T_{k+1} = \alpha_{temp}.T_k, 0 < a < 1, 1 - a \ll 1.$$

▸ Autres estimateurs

La difficulté de fournir des estimateurs relève du fait que l'on doit être capable de trouver la solution du problème de minimisation (3.8).

H. Rue, dans [28], [29], [30], et [32], nous fournit des méthodes de modélisation de fonctions de coûts, pour lesquelles on peut résoudre (3.8). En particulier, la ***distance de Baddeley***, introduite par A.J. Baddeley, paraît tout à fait appropriée à notre étude, puisqu'elle mesure une distance entre deux images.

H. Rue s'est donc chargé de modéliser une fonction de coût s'inspirant de la distance de Baddeley. Ce nouvel estimateur pourrait se révéler plus adapté que le premier, lorsque nous effectuerons une série de tests en le substituant au maximum a posteriori utilisé jusqu'alors.

3.3. Noyau de proposition adopté

En théorie, l'algorithme de Metropolis-Hastings-Green, pourvu des mouvements de naissance et de mort seulement, converge vers la loi objectif. Les propriétés mathématiques sont en effet toutes vérifiées. Dans la pratique, le noyau de proposition est plus complexe pour nous permettre d'accélérer cette convergence, déjà ralentie par l'utilisation d'un recuit simulé et de la lente décroissance des températures.

Notre objectif est en effet de modéliser des noyaux de propositions adaptés à la loi objectif, afin d'éviter au maximum des mouvements inutiles et de minimiser le taux de rejet. Ces mouvements seront groupés en deux catégories dans cette partie : les mouvements dits ***simples***, c'est à dire ceux modifiant un des objets de la configuration courante, et les mouvement ***complexes***, qui changent le nombre d'objets de la configuration (comme la naissance et la mort).

Soient $\mathbf{x}$ une configuration, et $A \in \mathcal{B}(\mathbf{\Psi})$. Le noyau de propositions $Q(\mathbf{x},A)$, défini sur $\mathbf{\Psi} \times \mathcal{B}(\mathbf{\Psi})$, peut se décomposer sous la forme d'une somme de noyaux de propositions, une fois mises en facteurs les probabilités $Q_i(\mathbf{x},\mathbf{\Psi}) = p_i$:

$$Q(\mathbf{x},A) = \sum_{i \in I} p_i Q_i(\mathbf{x},A)$$

où chacun des noyaux $Q_i(\mathbf{x},.)$ décrit un mouvement.

3.3.1 Mouvements simples

Principalement deux mouvements simples nécessitent d'être inclus dans notre noyau de propositions : la translation des objets (qui affecte leurs composants dans l'espace des positions $\mathcal{P}$), ainsi que la dilatation (qui affecte leur composant dans l'espace des marques $\mathcal{K}$).

Ces mouvements simples sont la combinaison d'une mort et d'une naissance, il font donc gagner du temps à l'algorithme, qui n'a plus besoin d'accepter ces deux mouvements, mais juste un seul.

▶ Translation

▷ Principe

Soit $x_i = (X_{x_i},Y_{x_i},R_{x_i})$ un des objets d'une configuration $\mathbf{x}$. Le mouvement de translation propose de modifier ses coordonnées (X_{x_i},Y_{x_i}) dans l'espace des positions $\mathcal{P}$.

Pour cela, nous introduisons une variable aléatoire θ sur un compact symétrique de $\mathbb{R}^2$, Θ : ainsi, si $\theta \in \Theta$, alors $-\theta \in \Theta$. Cette variable aléatoire suit une loi λ_2-uniforme sur Θ, $\lambda_2(.)$ étant la mesure de Lebesgue sur $\mathbb{R}^2$. Dans notre algorithme, nous avons choisi pour Θ le disque de rayon 1.

En posant $\Theta = [-a_X,a_X] \times [-b_Y,b_Y]$, les étapes de la translation sont donc les suivantes :

1. Choix uniforme d'un objet x_i de la configuration, avec la probabilité $\frac{1}{n(\mathbf{x})}$, que l'on va translater.

2. Génération de la variable aléatoire $\theta = (\theta_X,\theta_Y) \in \Theta$.

3. Calcul de l'objet x_i translaté, noté $\theta(x_i) = (X_{x_i} + \theta_X, Y_{x_i} + \theta_Y, R_{x_i})$. Si celui-ci n'appartient pas à U (car sa position est hors de l'espace des positions $\mathcal{P}$), le mouvement est aussitôt rejeté.

4. Calcul du rapport de Green R_t.

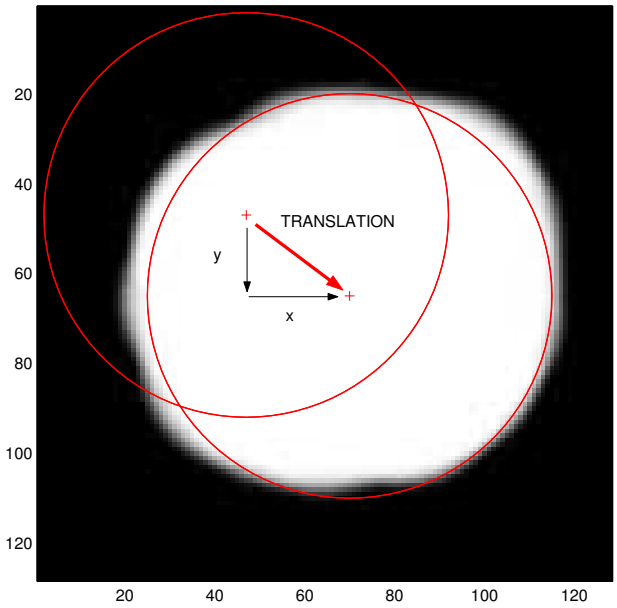

FIG. 3.6 – *Mouvement de translation*

5. Acceptation ou non de la nouvelle configuration $\mathbf{y} = \mathbf{x} \cup \{\theta(x_i)\} \setminus \{x_i\}$.

Ce mouvement est illustré dans la Fig. (3.6).

▷ **Noyau de translation et mesure symétrique**

Avec les mêmes notations que précédemment, le noyau de translation d'une configuration $\mathbf{x}$ vers un borélien A sera noté $Q_t(\mathbf{x},A)$, et choisi avec une probabilité p_t. On a :

$$Q_t(\mathbf{x},A) = \sum_{x_i \in \mathbf{x}} \frac{1}{n(\mathbf{x})} \int_{\theta \in \Theta} \mathbf{1}_A(\mathbf{x} \cup \{\theta(x_i)\} \setminus \{x_i\}) \frac{\lambda_2(d\theta)}{\lambda_2(\Theta)}. \tag{3.10}$$

La mesure symétrique $\xi_t(.,.)$ définie sur $\mathbf{\Psi} \times \mathbf{\Psi}$, par rapport à laquelle $\pi(.)Q_t(.,.)$ sera absolument continue, est :

$$\xi_t(A,B) = \int_{\mathbf{\Psi}} \mathbf{1}_A(\mathbf{x}) \sum_{x_i \in \mathbf{x}} \int_{\theta \in \Theta} \mathbf{1}_B(\mathbf{x} \cup \{\theta(x_i)\} \setminus \{x_i\}) \lambda_2(d\theta) \mu(d\mathbf{x}). \tag{3.11}$$

En posant $\mathbf{y} = \mathbf{x} \cup \{\theta(x_i)\} \setminus \{x_i\}$, la dérivée de Radon Nikodym $f_t(\mathbf{x},\mathbf{y})$, associée à ce mouvement de translation, s'exprime facilement :

$$
\begin{aligned}
f_t(\mathbf{x},\mathbf{y}) &= \frac{\pi(d\mathbf{x})Q_t(\mathbf{x},d\mathbf{y})}{\xi_t(d\mathbf{x},d\mathbf{y})} \\
&= \frac{h(\mathbf{x})\mu(d\mathbf{x})Q_t(\mathbf{x},d\mathbf{y})}{\mu(d\mathbf{x})\lambda_2(d\theta)} \\
&= \frac{h(\mathbf{x})}{n(\mathbf{x})\lambda_2(\Theta)}.
\end{aligned}
$$

▷ **Rapport de Green**

Le rapport de Green associé à cette transformation s'obtient donc :

$$
\boxed{R_t(\mathbf{x},\mathbf{x} \cup \{\theta(x_i)\} \setminus \{x_i\}) = \frac{h(\mathbf{x} \cup \{\theta(x_i)\} \setminus \{x_i\})}{h(\mathbf{x})}}. \tag{3.12}
$$

▸ Dilatation

▷ **Principe**

Soit $x_i = (X_{x_i},Y_{x_i},R_{x_i})$ un des objets d'une configuration $\mathbf{x}$. Le mouvement de dilatation propose quant à lui de modifier sa coordonnée R_{x_i} dans l'espace des marques $\mathcal{K}$.

Pour cela, nous introduisons une variable aléatoire δ sur un compact symétrique de $\mathbb{R}$, Δ : ainsi, si $\delta \in \Delta$, alors $-\delta \in \Delta$. Cette variable aléatoire suit une loi λ-uniforme sur Δ, $\lambda(.)$ étant la mesure de Lebesgue sur $\mathbb{R}$. Dans notre algorithme, Δ est le segment $[-1,1]$.

En posant $\Delta = [-c_R,c_R]$, les étapes de la translation sont donc les suivantes :

1. Choix uniforme d'un objet x_i de la configuration, avec la probabilité $\frac{1}{n(\mathbf{x})}$, que l'on va dilater.
2. Génération de la variable aléatoire $\delta \in \Delta$.
3. Calcul de l'objet x_i dilaté, noté $\delta(x_i) = (X_{x_i},Y_{x_i},R_{x_i} + \delta)$. Si celui-ci n'appartient pas à U (car son rayon est hors de l'espace des marques $\mathcal{K} = [R_{min},R_{max}]$), le mouvement est aussitôt rejeté
4. Calcul du rapport de Green.
5. Acceptation ou non de la nouvelle configuration $\mathbf{y} = \mathbf{x} \cup \{\delta(x_i)\} \setminus \{x_i\}$.

Ce mouvement est illustré dans la Fig. (3.7).

▷ **Noyau de dilatation et mesure symétrique**

Le noyau de dilatation d'une configuration $\mathbf{x}$ vers un borélien A sera noté $Q_d(\mathbf{x},A)$, et choisi avec une probabilité p_d. On a :

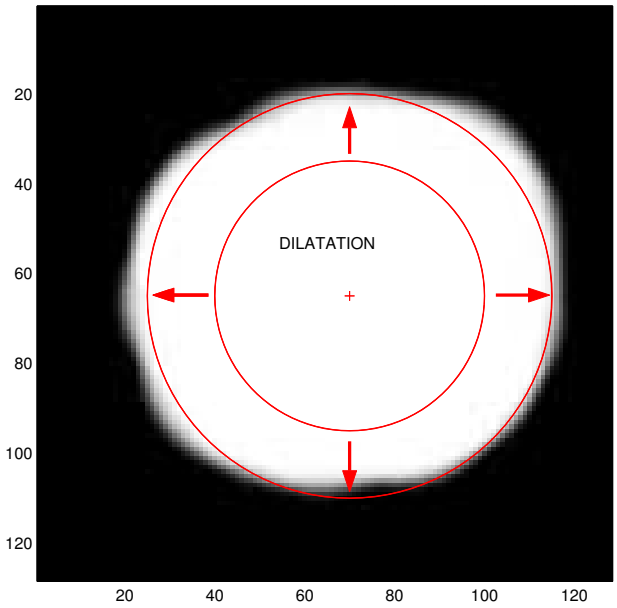

FIG. 3.7 – *Mouvement de dilatation*

$$Q_d(\mathbf{x},A) = \sum_{x_i \in \mathbf{x}} \frac{1}{n(\mathbf{x})} \int_{\delta \in \Delta} \mathbf{1}_A(\mathbf{x} \cup \{\delta(x_i)\} \setminus \{x_i\}) \frac{\lambda(d\delta)}{\lambda(\Delta)}. \tag{3.13}$$

La mesure symétrique $\xi_d(.,.)$ définie sur $\mathbf{\Psi} \times \mathbf{\Psi}$, par rapport à laquelle $\pi(.)Q_d(.,.)$ sera absolument continue, est :

$$\xi_d(A,B) = \int_{\mathbf{\Psi}} \mathbf{1}_A(\mathbf{x}) \sum_{x_i \in \mathbf{x}} \int_{\delta \in \Delta} \mathbf{1}_B(\mathbf{x} \cup \{\delta(x_i)\} \setminus \{x_i\}) \lambda(d\delta) \mu(d\mathbf{x}). \tag{3.14}$$

En posant $\mathbf{y} = \mathbf{x} \cup \{\delta(x_i)\} \setminus \{x_i\}$, la dérivée de Radon Nikodym $f_d(\mathbf{x},\mathbf{y})$, associée à ce mouvement de dilatation, s'exprime facilement :

$$\begin{aligned} f_d(\mathbf{x},\mathbf{y}) &= \frac{\pi(d\mathbf{x})Q_d(\mathbf{x},d\mathbf{y})}{\xi_d(d\mathbf{x},d\mathbf{y})} \\ &= \frac{h(\mathbf{x})\mu(d\mathbf{x})Q_d(\mathbf{x},d\mathbf{y})}{\mu(d\mathbf{x})\lambda(d\delta)} \\ &= \frac{h(\mathbf{x})}{n(\mathbf{x})\lambda(\Delta)}. \end{aligned}$$

▷ **Rapport de Green**

Le rapport de Green associé à cette transformation s'obtient donc :

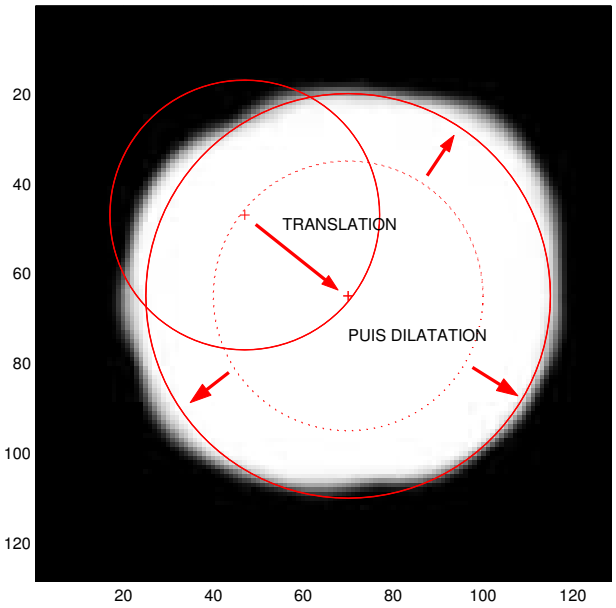

FIG. 3.8 – *Mouvement de translation et de dilatation*

$$\boxed{R_d(\mathbf{x},\mathbf{x}\cup\{\delta(x_i)\}\setminus\{x_i\}) = \frac{h(\mathbf{x}\cup\{\delta(x_i)\}\setminus\{x_i\})}{h(\mathbf{x})}}. \tag{3.15}$$

▸ Combinaison de mouvements simples

Enfin, le dernier mouvement simple que nous proposons est la combinaison d'une translation et d'une dilatation. Parfois en effet, une simple translation ou une simple dilatation n'est pas acceptée, alors que la combinaison des deux peut se révéler efficace.

Avec les mêmes notations que précédemment, le mouvement de translation-dilatation consiste en la succession d'une dilatation puis translation. La Fig. (3.8) illustre cette transformation.

Son noyau de proposition, d'une configuration $\mathbf{x}$ vers un borélien A, sera noté $Q_{td}(\mathbf{x},A)$ et choisi avec une probabilité p_{td}. On a :

$$Q_{td}(\mathbf{x},A) = \sum_{x_i\in\mathbf{x}} \frac{1}{n(\mathbf{x})} \int_{\theta\in\Theta}\int_{\delta\in\Delta} \mathbf{1}_A(\mathbf{x}\cup\{\theta(\delta(x_i))\}\setminus\{x_i\}) \frac{\lambda(d\delta)}{\lambda(\Delta)} \frac{\lambda_2(d\theta)}{\lambda_2(\Theta)}. \tag{3.16}$$

Le rapport de Green associé à cette transformation s'obtient de la même façon que les précédents :

$$\boxed{R_{td}(\mathbf{x},\mathbf{x}\cup\{\theta(\delta(x_i))\}\{\setminus x_i\}) = \frac{h(\mathbf{x}\cup\{\theta(\delta(x_i))\}\setminus\{x_i\})}{h(\mathbf{x})}}. \tag{3.17}$$

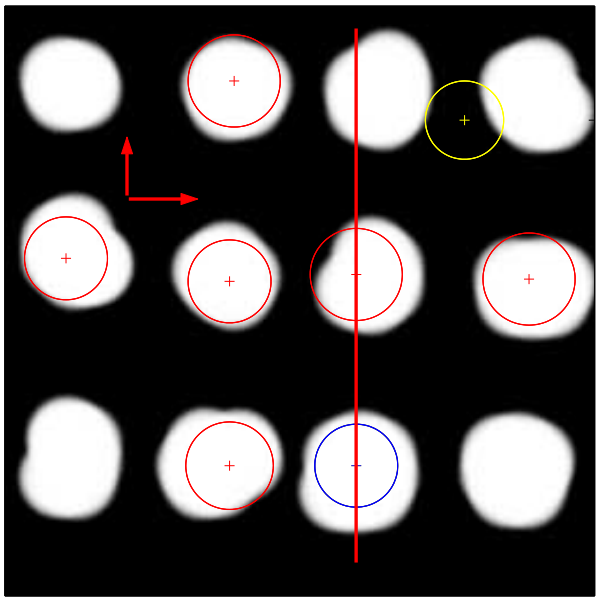

FIG. 3.9 – *Deux types de naissance. En rouge : les objets de la configuration à l'instant t. En jaune : proposition de naissance uniforme d'un objet. En bleu : proposition de naissance non uniforme d'un objet, utilisant les objets déjà existants ainsi que les directions privilégiées d'alignement.*

3.3.2 Mouvements complexes

On appelle mouvement complexe tout mouvement aboutissant à un changement de dimension de la configuration $\mathbf{x}$.

Les mouvements de naissance et de mort uniformes ont déjà été présentés dans une partie antérieure. Ce ne sont pas les seuls à mériter d'être proposés. Ainsi nous verrons dans notre prochain chapitre qu'un mouvement proposant la fusion de deux objets, ou bien au contraire la scission en deux d'un cercle, améliorerait nos résultats. De même, une naissance non uniforme, mais proposée dans un voisinage d'un objet déjà existant, et dans une direction prenant en compte les directions d'alignements si elles existent, serait bénéfique.

Les bases mathématiques de ces différents mouvements sont acquises [25], mais l'implantation n'a pas encore été effectuée.

Le schéma de la Fig. (3.9) donne un exemple de naissance non uniforme, en se servant d'une direction observée par un alignement dans l'image, qui peut se révéler plus efficace qu'une naissance uniforme.

3.4. Bilan : notre algorithme

Voici en résumé, l'algorithme de processus ponctuels simulés par une chaîne de Markov que nous avons implanté. Sont décrites ici les principales étapes de celui-ci, programmé en C++.

⊛ ***Initialisation***

- Lecture de l'image de données.
- Rentrée des paramètres.
- Initialisation de la configuration $\mathbf{x}_{t=0}$ à la configuration nulle.

⊛ ***Boucle d'itération***

↻ Itération numéro k.

- Calcul de la température T_k.
- Choix d'un des 5 mouvements, avec les probabilités $\{p_n,p_m,p_t,p_d,p_{td}\}$.
- Proposition d'une nouvelle configuration $\mathbf{y}$ à partir de la configuration courante $\mathbf{x}_k$, faite par le noyau sélectionné.
- Acceptation ou Rejet, suite au calcul du rapport de Green.
- Fin de l'itération k. $k = k + 1$.

⊛ ***Fin de l'algorithme***

Quand la température T_k est inférieure à une limite donnée, on stoppe les itérations. Affichage des objets de la configuration sur l'image de données. Analyse du résultat obtenu.

CHAPITRE

4 Premiers résultats sur des images réelles et Interprétations

Nous avons effectué une série de tests de notre modèle sur des images. Tout d'abord, ils furent menés sur des images synthétiques, dans le but d'améliorer la syntaxe C++ et la vitesse de l'algorithme (au sens informatique, et non mathématique), puis sur des images fournies par l'IFN.

Dans ce chapitre, nous présenterons ces résultats et les comparerons avec ceux obtenus, à partir d'une même image, avec la méthode de suivi de vallées introduite par F.A. Gougeon.

Les critères d'évaluation furent difficiles à déterminer, d'autant plus que nous ne disposions ni d'une vérité terrain censée être la "solution" à obtenir, ni de l'aide d'un expert. Aussi, nous axerons principalement nos interprétations des images sur les erreurs importantes de détection que nous pouvons repérer.

4.1. Images étudiées

Nous avons extrait manuellement dans les images fournies par l'IFN des zones de peupliers d'environ 60m sur 60m (128×128 pixels) afin d'accélérer le traitement informatique, la vitesse de décroissance des températures du recuit simulé se devant d'être d'autant plus faible que l'espace objet à explorer est grand.

La Fig. (4.1) montre les 2 imagettes qui ont été extraites de l'image des forêts de Saône et Loire, sur lesquelles nous avons testé nos algorithmes.

Toutes les images de notre base de données comportent des zones de peupliers à extraire parmi de la végétation, des vignes ou d'autres espèces d'arbres, ou des fleuves. Nous avons dans un premier temps contourné ce problème, en choisissant d'extraire manuellement les parties qui nous intéressaient.

L'idéal aurait été d'établir un procéssus automatique de segmentation de l'image par espèce d'arbres, en utilisant par exemple des informations de textures afin d'obtenir une signature particulière pour chacune des espèces. Suivant les objectifs que nous nous fixerons par la suite, nous aurons ou non à réaliser une telle classification.

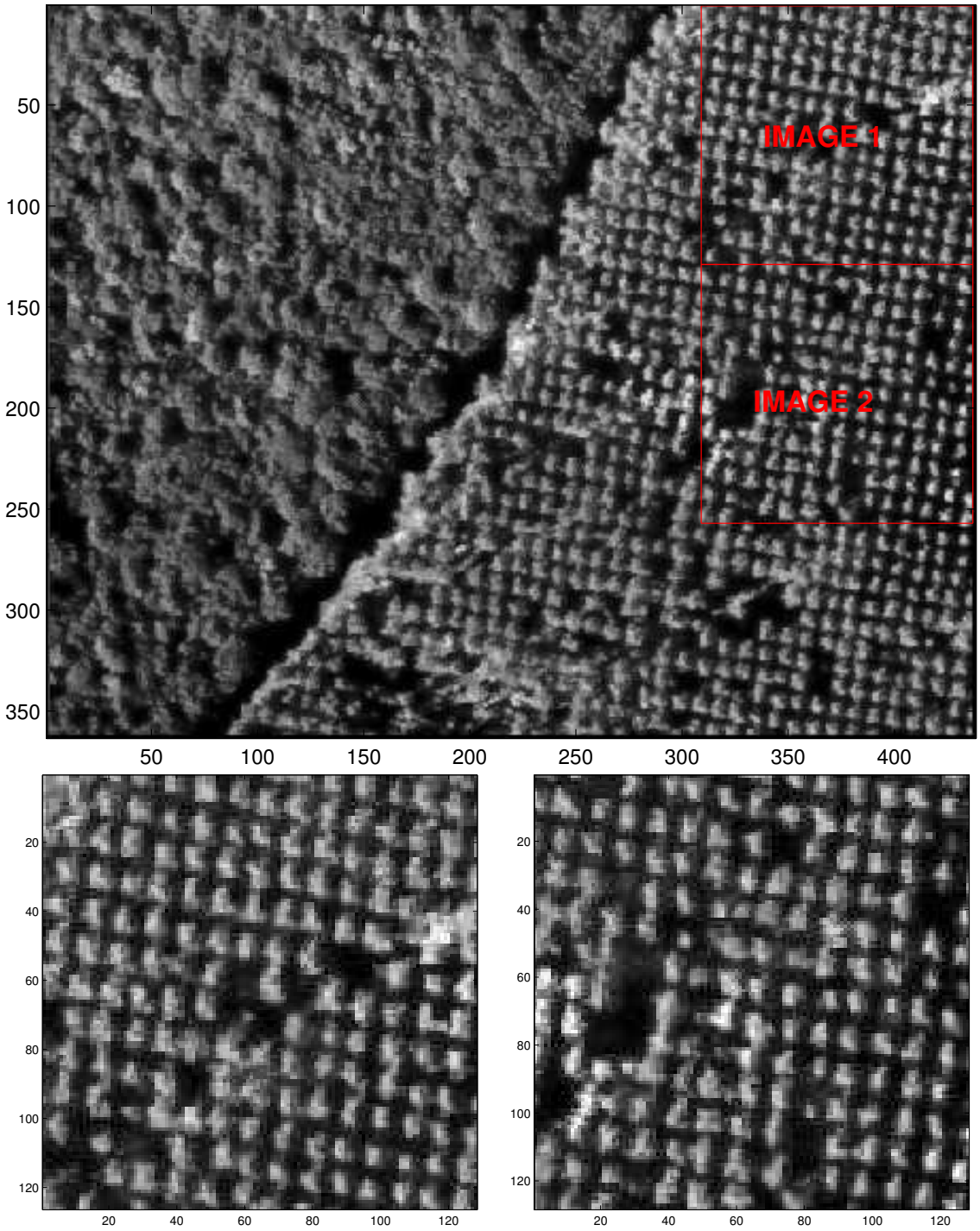

FIG. 4.1 – *Partition de l'image initiale. En haut : image originale, les carrés rouges représentent les deux zones extraites. En bas à gauche : Image 1. En bas à droite : Image 2.* ©*IFN*

4.2. Résultats et Interprétations

Les résultats obtenus étant très semblables sur les deux imagettes 1 et 2, nous ne présenterons que ceux relatifs à l'imagette 1. On s'aperçoit en effet que leurs strucutres sont similaires. Les interprétations que nous formulerons seront donc valables pour l'ensemble des tests effectués.

4.2.1 Choix des paramètres

Si l'on se réfère à la partie précédente présentant notre modèle, les différents paramètres de ce dernier sont :

1. pour la densité a priori $h_p(.)$: β,ϵ,$\gamma_{strauss}$,γ_{aire}.

   Au cours de nos tests effectués sur l'imagette 1, le choix entre une répulsion de Strauss ou une intéraction d'aires de la densité a priori, se révéla décisif.

   Ayant tout d'abord privilégié la répulsion de Strauss ($\epsilon = 1$), nous avons remarqué que de nombreux arbres avaient tendance à être détectés par plusieurs cercles de la configuration, qui se chevauchaient et formaient donc des paires d'objets pénalisées par notre terme répulsif. Cette sur-détection peut s'expliquer par le fait que la "perte" au niveau de la densité a priori était largement compensée par le "gain" au niveau du terme de vraisemblance. En effet, pour chaque nouvelle paire, l'ordre de grandeur de $e^{\gamma_{strauss}}$ était négligeable par rapport au terme de vraisemblance, qui favorisait une telle configuration. Abaisser le coefficient $\gamma_{strauss}$ n'était pas non plus une solution plausible, puisqu'alors aucun chevauchement n'aurait été possible.

   Nous avons donc préféré utiliser la répulsion par intéraction d'aires, avec un coefficient d'intéraction assez fort : $\gamma_{aire} = ln(0.01)$. Celle-ci répond correctement à nos attentes, car elle pénalise les objets d'autant plus qu'ils se superposent à d'autres.

   Enfin, nous avons remarqué que le coefficient pondérateur β, sorti de l'expression de la densité $h(.)$ d'après (3.4), influait peu sur les résultats obtenus. Par contre, il agit principalement sur la dynamique de l'algorithme lorsque la température est élevée, en faisant naître plus ou moins d'objets, objets qui seront tués par la suite s'ils sont mal positionnés. A faible température, son influence est négligeable comparée à celle de $h(\mathbf{x})^{\frac{1}{T}}$.

   Nous devions donc à la fois faire naître assez d'objets dans la phase transitoire de l'algorithme (forte températures), mais pas trop, afin d'éviter de perdre du temps par la suite en tuant ces objets qui s'avèrent mal placés. Le choix de $\beta = 10$ s'est avéré judicieux dans la pratique.

2. pour le terme de vraisemblance $\mathcal{L}(\mathcal{I}|\mathbf{x})$: les moyennes et variances des deux classes, à savoir μ_A,σ_A,μ_F,σ_F.

   Au moment d'initialiser l'algorithme, nous choisissons les paramètres initiaux des deux classes. Des procédés automatiques (k-means) peuvent nous aider dans cette décision. Puis, nous lançons une première série d'itérations. A partir de celle-ci, nous estimons à nouveau ces paramètres du modèle en sélectionnant des échantillons parmi les configurations d'objets, et en utilisant les théorèmes limites (2.24).

   Dans l'imagette 1, l'estimation donne alors : $\mu_A = 115$, $\sigma_A = 41$, $\mu_F = 48$, et $\sigma_F = 27$. La Fig. (4.2) montre que les distributions de ces classes sont bien disjointes, ce qui nous laisse présager une classification de qualité.

3. pour le recuit simulé : $T_{t=0}$,α_{temp},T_{min}.

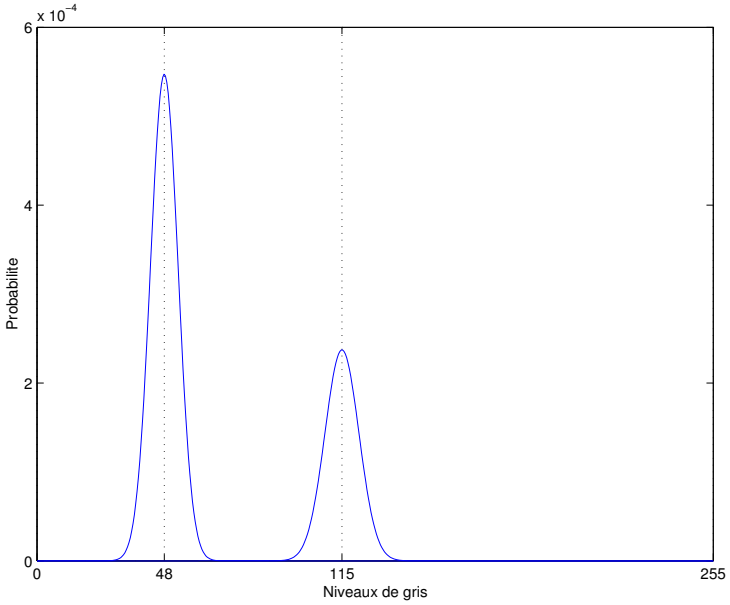

FIG. 4.2 – *Répartition des niveaux de gris des deux classes*

Nous avons opté pour une décroissance géométrique par palliers, de largeur s. Ainsi, les températures $(T_{ks},\ldots,T_{(k+1)s-1})$ restent égales, et $T_{(k+1)s} = \alpha_{temp}.T_{ks}$. Aux paramètres du recuit simulé s'ajoute donc s la largeur du pallier.

La décroissance géométrique fut assurée avec le paramètre $\alpha_{temp} = 0.99995$. Même si ceci peut paraître très lent, il n'en est rien en réalité : nous rappelons en effet que théoriquement, il aurait fallu une décroissance logarithmique de la température pour nous assurer de la convergence de l'algorithme.

4. pour les noyaux de propositions : p_n,p_m,p_t,p_d,p_{td}.

   A chaque itération, les différents noyaux de propositions furent choisis de façon équiprobable. Nous avons veillé à ce que la somme des probabilités soit inférieures à 1 pour assurer l'apériodicité de la chaîne. Finalement, les paramètres en question furent les suivants : $p_n = p_m = p_t = p_d = p_{td} = 0.199$.

4.2.2 Exemples de résultats obtenus

Nous présentons deux de nos résultats, aux Fig. (4.3) et Fig. (4.4), représentatifs de la série de tests que nous avons menée. Dans le deuxième exemple, un refroidissement du recuit simulé plus rapide a été opéré.

La machine sur laquelle nous avons travaillé est une ***SUN - 440 MHz***.

4.2.3 Interprétations

Comme expliqué plus haut, l'absence de vérité terrain pour cette parcelle rend impossible l'interprétation de nos résultats au niveau du nombre d'arbres détectés. Par contre, nous pouvons observer plusieurs erreurs de détection de houppiers.

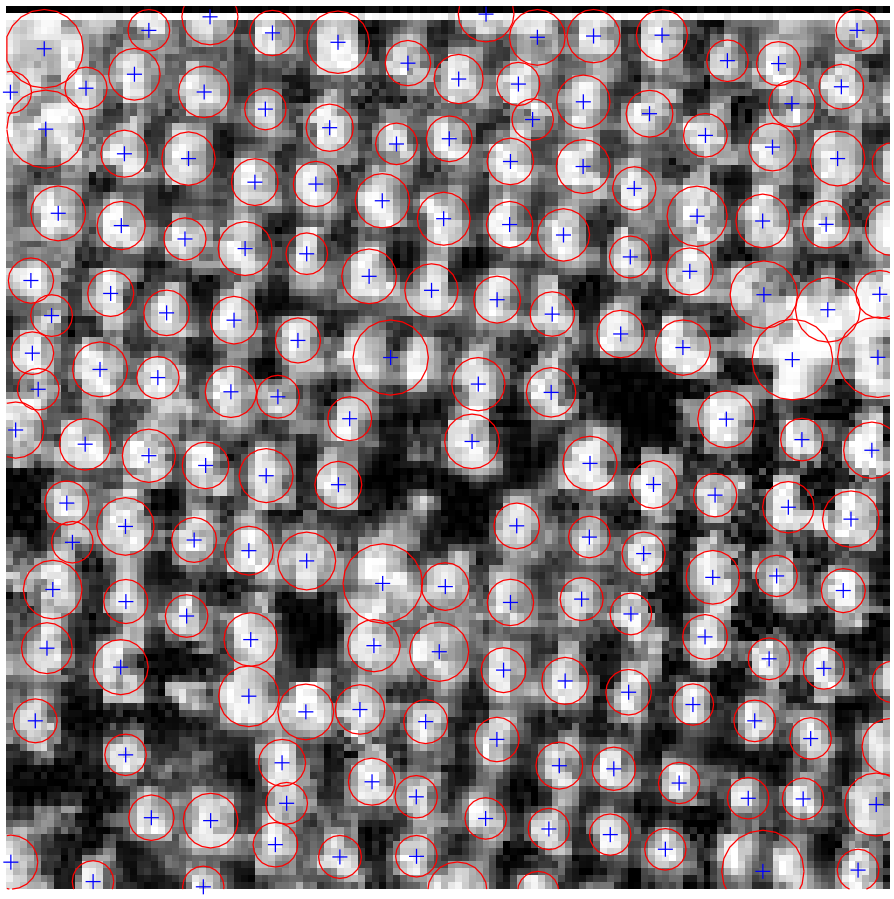

FIG. 4.3 – $T_{t=0} = 0.1$, $s = 100$, *10 millions d'itérations*, $T_{min} = 6.73\,10^{-4}$, *temps CPU : 812 secondes*

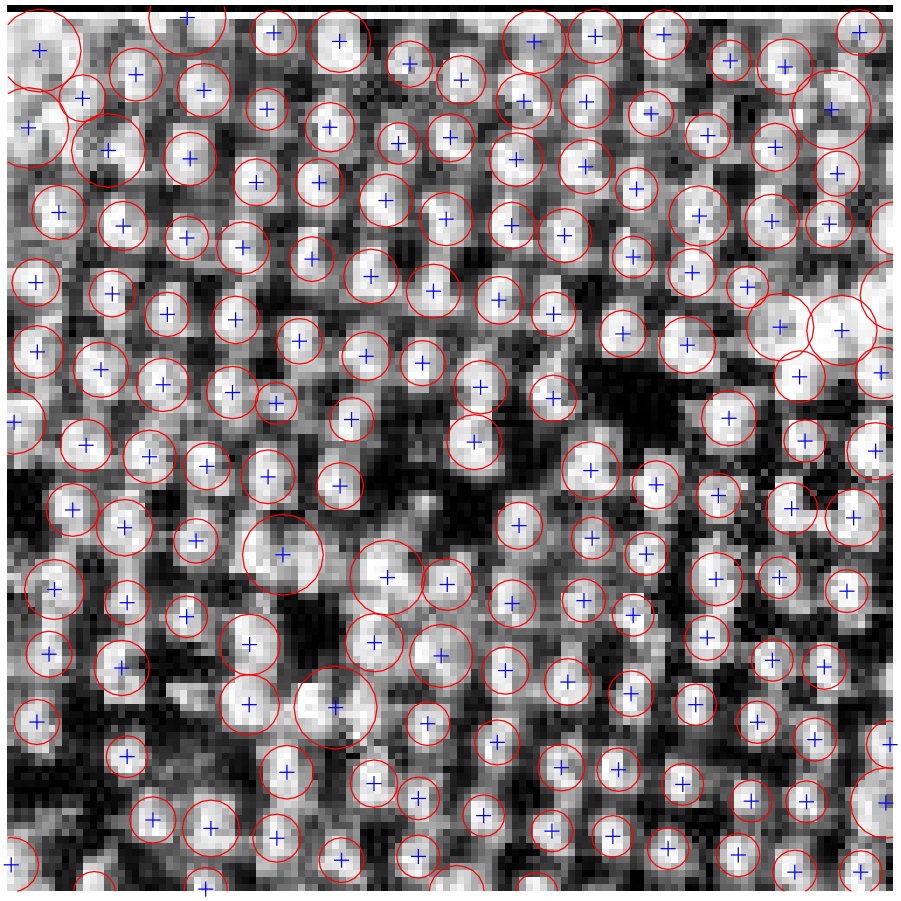

FIG. 4.4 – $T_{t=0} = 0.1$, $s = 10$, *1 million d'itérations*, $T_{min} = 6.73\,10^{-4}$, *temps CPU : 89 secondes*

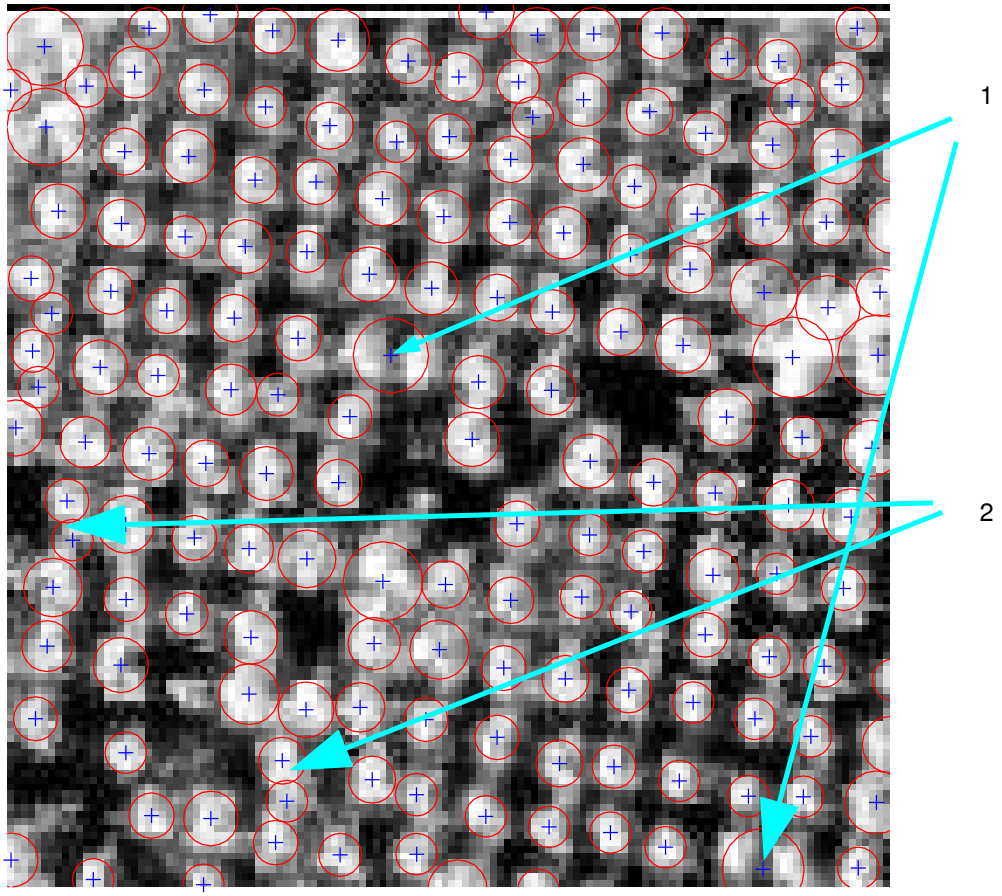

FIG. 4.5 – *Quelques erreurs de détection. Cas 1 : sous-détection. Cas 2 : sur-détection.*

▸ Manque de mouvements et problèmes avec le recuit simulé

La Fig. (4.5) met en évidence les deux cas principaux d'erreurs de détection. Le cas 1 est rencontré lorsqu'un gros cercle s'étale sur deux arbres, et le cas 2 lorsque deux petits cercles sont nécessaires pour détecter un arbre. Avec les mouvements que nous lui autorisions, la configuration aurait dû effectuer deux étapes pour résoudre d'elle-même ce problème. Dans le premier cas, il s'agisait de translater et de réduire la taille du cercle, puis d'en faire naître un deuxième sur l'autre arbre (translation-dilatation + naissance). Dans le deuxième cas, il aurait fallu proposer de tuer un des deux cercles, puis de translater et d'agrandir l'autre (mort + translation-dilatation).

Raisonnons au niveau énergétique afin de mieux comprendre ce phénomène, et considérons qu'il nous faut trouver le minimum global d'une énergie $E(\mathbf{x})$, reliée à la densité a posteriori par : $h(\mathbf{x}) = e^{-\frac{E(\mathbf{x})}{T_k}}$. Reportons nous pour cela au schéma de la Fig. (4.6).

Nous nous retrouvons dans les deux cas "coincés" dans un minimum local, duquel nous ne pourrions sortir que s'il existait un seul mouvement capable de passer de la configuration actuelle, c'est à dire le point A sur le schéma, à une configuration d'énergie plus faible, le point B.

Avec les mouvements disponibles dans notre noyau de propositions, nous avons vu que pour sauter de A à B, il nous fallait passer par un point intermédiaire C, une configuration d'énergie plus grande que A. Or, le passage de A à C est rejeté d'autant plus facilement que la température est faible, puisque le rapport de Green est proportionnel à $e^{\frac{E(A)-E(C)}{T_k}}$. En effet, pour les températures

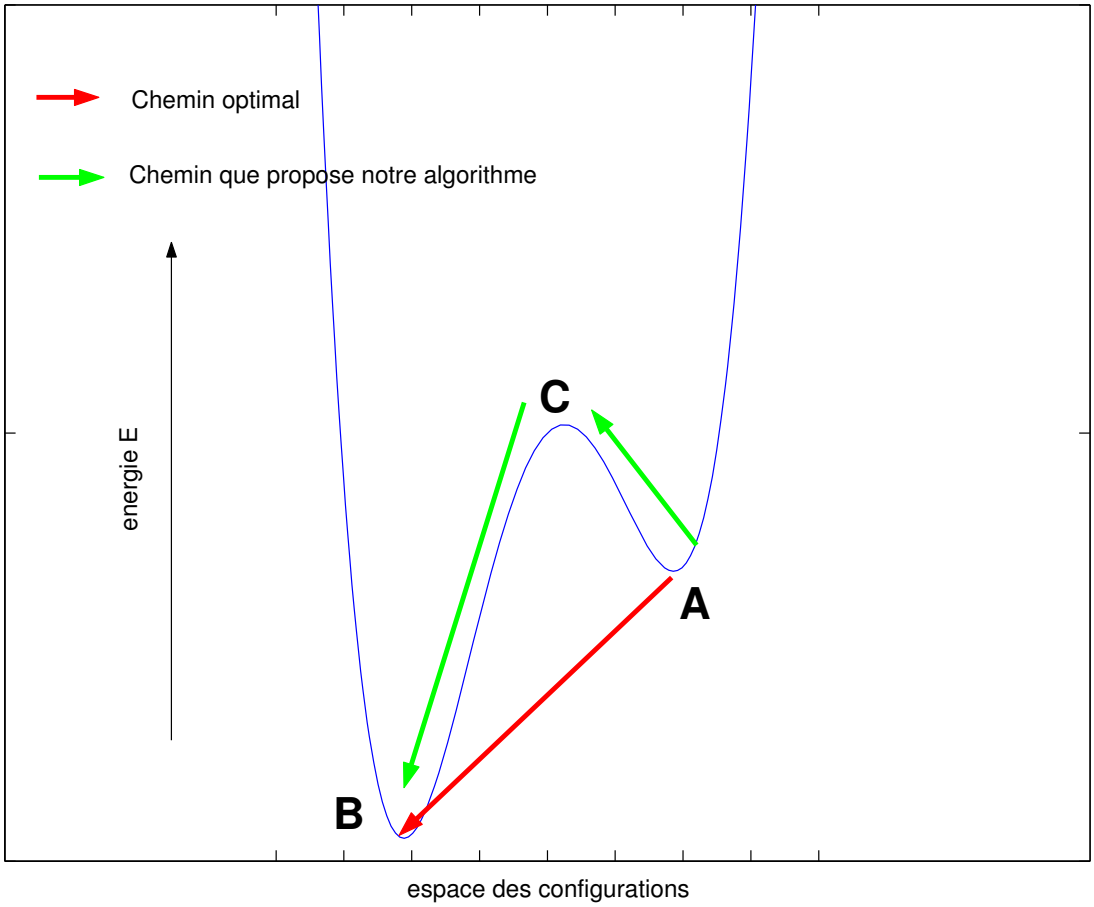

FIG. 4.6 – *L'analogie énergétique*

proches de 0, le comportement de la chaîne est quasi déterministe, et n'accepte de nouvelles configurations que si elles ont une énergie plus faible que la configuration courante.

Autrement dit, c'est en baissant extrêmement lentement la température que nous pourrions éviter de nous retrouver bloqués dans un minimum local, afin d'avoir le temps d'effectuer ces deux mouvements.

Une autre solution à ce problème serait d'inclure dans le noyau de proposition des mouvements qui permettent de passer de A à B en une étape, à savoir la fusion de deux cercles, ou inversement la scission en deux d'un cercle.

▸ Progrès envisageables sur la densité

De nombreuses améliorations peuvent être apportées à notre modèle de densité, tant au niveau de l'a priori que de la vraisemblance. En effet, nous nous sommes rendus compte que celle-ci, finalement, n'intervenait que peu dans les solutions données par l'algorithme, qui dépendaient beaucoup plus des paramètres du recuit simulé.

Pour gagner du temps de calcul, des idées ont été proposées aux chapitres précédents (naissance non uniforme par exemple), mais c'est notre modèle entier qu'il faudra revoir : le plus urgent sera de remodéliser l'attache aux données, en incluant des attributs de textures. Cela ne sera sans doute pas utile pour le cas des peupliers, qui aurait surtout besoin des mouvements de fusion et de scission, mais le sera pour des images plus complexes contenant différentes espèces d'arbres.

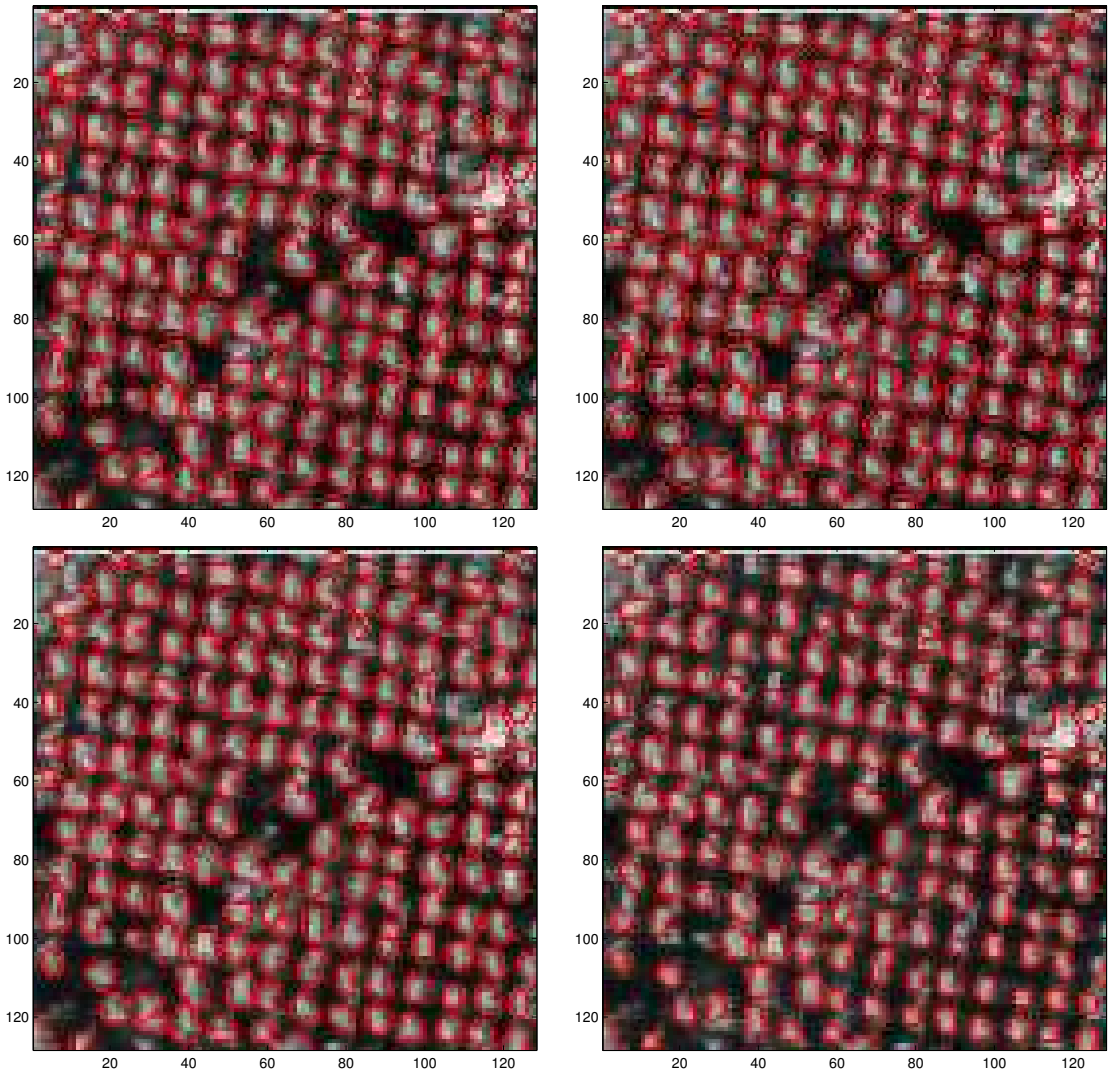

FIG. 4.7 – *Résultats à partir du suivi de vallées. De haut en bas et de gauche à droite : (s=4,$\theta = 60$), (s=2,$\theta = 50$), (s=2,$\theta = 70$) et (s=2,$\theta = 90$). 4 secondes de traitement pour chacun.*

4.3. Comparaison avec le suivi de vallées

4.3.1 Résultats obtenus avec le suivi de vallées

Afin d'évaluer les performances de notre détection de houppiers à partir de processus objets, nous avons testé sur les mêmes images l'approche pixélique de F.A. Gougeon, et son algorithme de suivi de vallées. Les 4 images de la Fig. (4.7) montrent les résultats obtenus pour différentes valeurs des paramètres (s,θ), qui sont respectivement les paramètres de filtrage et de seuillage.

Le meilleur résultat semble être obtenu pour les paramètres (s=2,$\theta = 70$). La détection du contour de l'arbre n'est cependant pas toujours exacte, notamment dans les zones de forte densité. On peut imaginer qu'une telle méthode ne marche pas correctement avec d'autres espèces plus denses, comme le faisait remarquer F.A. Gougeon dans ses articles.

4.3.2 Tableau comparatif

Nous souhaitons préciser une nouvelle fois que la série de tests réalisée sur des peupliers n'est que le début de notre travail de détection des houppiers à partir de processus objet, destiné à traiter une classe d'images plus vaste.

Nos résultats avec les peupliers seront sans doute bien améliorés par le rajout des mouvements de fusion et de scission. De même, le temps de calcul informatique pourra être réduit en retravaillant le code ou en utilisant une machine plus performante.

Cependant, il est clair que notre approche objet pourrait s'adapter plus facilement que l'approche pixélique à d'autres espèces d'arbres. Elle est plus évolutive que son homologue, cette dernière n'utilisant aucune autre information que les niveaux de gris. En effet une connaissance a priori des espèces peut se révéler précieuse, de même qu'une modélisation des intéractions entre les arbres pour des milieux forestiers denses.

Enfin, il existe des méthodes stochastiques d'estimation de nos paramètres, ce qui nous donne un avantage par rapport au suivi de vallées, pour lequel les paramètres doivent être évalués au moyen des expériences.

Critères d'évaluation	Suivi de vallées	Processus objet
Qualité de la détection	+	+
Rapidité de l'algorithme	++	−
Evolutions envisageables	−	++

TAB. 4.1 – *Tableau comparatif des deux méthodes testées. Suivant les critères d'évaluation, des notes de −− à ++ sont attribuées aux deux approches.*

Conclusions et Perspectives

Des mathématiques à l'informatique

Ce stage fut l'occasion de travailler sur un sujet à la frontière entre les mathématiques et l'informatique. La théorie sur les processus objet est très riche, mais les domaines d'applications sont pour l'instant rares.

Face à la théorie, les dialogues avec les experts des milieux forestiers ont su nous confronter à la réalité. Ils ont fait état de l'importance pratique et économique que revêt le développement d'un processus semi-automatique de détection des houppiers.

L'apport de l'approche objet

Les processus objet se révèlent être très prometteurs pour la détection de houppiers, et l'obtention de statistiques sur les peuplements forestiers destinées aux organismes d'inventaire et de gestion nationaux. L'approche objet permet en effet d'exploiter les connaissances a priori des experts sur les différentes espèces, et pourrait s'en servir pour résoudre de nombreux cas complexes de détection, auxquels les approches classiques ne peuvent s'attaquer.

Au cours de ce stage, nous avons été en mesure de développer un modèle géométrique simple, nous donnant déjà des résultats intéressants.

Développements à venir

Les voies d'exploration pour améliorer notre modèle sont très nombreuses, et seront étudiées au cours de la thèse qui succèdera au stage. Un des avantages principaux de notre méthode est sans aucun doute sa grande capacité de développement.

Le modèle géométrique en lui-même sera à retravailler, de même que l'attache aux données. Nous envisageons en effet une étude de la texture des arbres, afin d'affiner la détection et de permettre une classification suivant les différentes espèces. De plus, nous testerons de nouveaux estimateurs de la configuration d'objets solution, puisque certains semblent plus adaptés au traitement des images.

Bibliographie

[1] P. Brémaud. *Markov chain : Gibbs fields, Monte Carlo simulation and queues*. Springer-Verlag, 2001.

[2] L. Garcin, X. Descombes, J. Zerubia, and H. Le Men. Building detection by markov object point processes and a mcmc algorithm. Rapport de Recherche n^o 4206, INRIA, juin 2001.

[3] C.J. Geyer and J. Moller. *Stochastic geometry, likelihood and computation*, chapter Likelihood inference for spatial point processes. Chapmann and Hall, Londres, 1998.

[4] F.A. Gougeon. Individual tree identification from high resolution meis images. In M.D. Gillis and D.G. Leckie, editors, *Proc. of the International Forum on Airborne Multispectral Scanning for Forestry and Mapping (with Emphasis on MEIS)*, pages 117–128, Forêts Canada, Val-Morin, Québec, 13-16 avril 1992.

[5] F.A. Gougeon. A crown-following approach to the automatic delineation of individual tree crowns in high spatial resolution aerial images. *Canadian Journal of Remote Sensing, 21(3)*, pages 274–284, 1995.

[6] F.A. Gougeon. A system for individual tree crown classification of conifer stands at high spatial resolution. In *Proc. of the* 17^{th} *Cananadian Symposium on Remote Sensing*, pages 635–642, Saskatchewan, Canada, 13-15 juin 1995.

[7] F.A. Gougeon. Automatic individual tree crown delineation using a valley-following algorithm and rule-based system. In D.A. Hill and D.G. Leckie, editors, *Proc. of the International Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, pages 11–23, Victoria, British Columbia, Canada, 10-12 février 1998.

[8] F.A. Gougeon and D.G. Leckie. Forest regeneration : individual tree crown detection techniques for density and stocking assessment. In D.A. Hill and D.G. Leckie, editors, *Proc. of the International Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, pages 169–177, Victoria, British Columbia, Canada, 10-12 février 1998.

[9] F.A. Gougeon and D.G. Leckie. Individual tree crown image analysis: a step towards precision forestry. In *Proc. of the* 1^{st} *International Precision Forestry Symposium*, 17-20 juin 2001.

[10] F.A. Gougeon, D.G. Leckie, I. Scott, and D. Paradine. Individual tree crown species recognition: the nahmint study. In D.A. Hill and D.G. Leckie, editors, *Proc. of the International Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, pages 209–223, Victoria, British Columbia, Canada, 10-12 février 1998.

[11] P.J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika 82*, pages 711–7320, 1995.

[12] P.J. Green and A. Mira. Delayed rejection in reversible jump metropolis hastings. Technical Report S-01-99, University of Bristol, 1999.

[13] M. Imberty and X. Descombes. Simulation de processus objets : étude de faisabilité pour une application à la segmentation d'image. Rapport de Recherche n^o 3881, INRIA, février 2000.

[14] C. Lacoste, X. Descombes, and J. Zerubia. A comparative study of point processes for line network extraction in remote sensing. Rapport de Recherche n^o 4516, INRIA, juillet 2002.

[15] M. Larsen. Finding an optimal match window for spruce top detection based on an optical tree model. In D.A. Hill and D.G. Leckie, editors, *Proc. of the International Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, pages 55–66, Victoria, British Columbia, Canada, 10-12 février 1998.

[16] M. Larsen. Individual tree top position estimation by template voting. In *Proc. of the Fourth International Airborne Remote Sensing Conference and Exhibition / 21st Canadian Symposium on Remote Sensing*, volume 2, pages 83–90, Ottawa, Ontario, 21-24 juin 1999.

[17] M. Larsen and M. Rudemo. Using ray-traced templates to find individual trees in aerial photographs. In Michael Frydrych, Jussi Parkkinen, and Ari Visa, editors, *Proc. of the 10th Scandinavian Conference on Image Analysis*, volume 2, pages 1007–1014, Lappeenranta, Finland, 1997.

[18] D.G. Leckie, C. Burnett, T. Nelson, C. Jay, N. Walsworth, F.A. Gougeon, and E. Cloney. Forest parameter extraction through computer-based analysis of high resolution imagery. In *Proc. of the Fourth International Airborne Remote Sensing Conference and Exhibition / 21st Canadian Symposium on Remote Sensing*, Ottawa, Ontario, 21-24 juin 1999.

[19] D.G. Leckie, M. Gillis, F.A. Gougeon, M. Lodin, J. Wakelin, and Y. Xiaoping. Computer-assisted photointerpretation aid to forest inventory mapping: some possible approaches. In D.A. Hill and D.G. Leckie, editors, *Proc. of the International Forum of Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, pages 335–343, Victoria, British Columbia, Canada, 10-12 février 1998.

[20] D.G. Leckie and F.A. Gougeon. An assessment of both visual and automated tree counting and species identification with high spatial resolution multispectral imagery. In D.A. Hill and D.G. Leckie, editors, *Proc. of the International Forum of Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, pages 141–152, Victoria, British Columbia, Canada, 10-12 février 1998.

[21] J. Lund. *Statistical inference and perfect simulation for point processes observed with noise.* PhD thesis, The Royal Veterinary and Agricultural University, Department of Mathematics and Physics, Copenhague, d'ecembre 1999.

[22] J. Lund and M. Rudemo. Models for point processes observed with noise. *Biometrika*, 87(2):235–249, 2000.

[23] S.P. Meyn and R.L. Tweedie. *Markov Chains and Stochastic Stability.* Springer-Verlag, 1993.

[24] M. Ortner, X. Descombes, and J. Zerubia. Building extraction from digital elevation model. Rapport de Recherche n^o 4517, INRIA, juillet 2002.

[25] M. Ortner, X. Descombes, and J. Zerubia. Improved rjmcmc point process sampler for object detection on images by simulated annealing. Rapport de Recherche n^o 4900, INRIA, août 2003.

[26] P.S. Petersen. *Finding single trees in aerial imagery.* PhD thesis, The Royal Veterinary and Agricultural University, Department of Economics and Natural Resources, Copenhague, décembre 2001.

[27] C. Robert. *Méthodes de Monte Carlo par chaînes de Markov.* Economica, Paris, 1996.

[28] H. Rue. New loss function in bayesian imaging. *American Statistical Association*, 90(431):900–908, septembre 1995.

[29] H. Rue. A loss function model for the restoration of grey level images. *Scandinavian Journal of Statistics*, 24:103–114, 1997.

[30] H. Rue. Baddeley's delta metric. *Encyclopedia of Statistical Sciences*, 3:158–162, 1999.

[31] H. Rue and M. Hurn. Bayesian object identification. *Biometrika*, 86:649–660, mai 1999.

[32] H. Rue and A.R. Syversveen. Bayesian object recognition with baddeley's delta loss. *Advances Applied Probability*, 30:64–84, 1998.

[33] R. Stoica, X. Descombes, and J. Zerubia. A markov point process for road extraction in remote sensed images. Rapport de Recherche n^o 3923, INRIA, avril 2000.

[34] D. Stoyan, W.S. Kendall, and J. Mecke. *Stochastic Geometry and its applications.* Wiley, 2nde edition, 1995.

[35] M.N.M. van Lieshout. *Markov point processes and their applications.* Imperial College Press, Londres, 2000.