



**HAL**  
open science

# Méthodes stochastiques en analyse d'image : des champs de Markov aux processus ponctuels marqués

Xavier Descombes

► **To cite this version:**

Xavier Descombes. Méthodes stochastiques en analyse d'image : des champs de Markov aux processus ponctuels marqués. Interface homme-machine [cs.HC]. Université Nice Sophia Antipolis, 2004. tel-00506084

**HAL Id: tel-00506084**

**<https://theses.hal.science/tel-00506084>**

Submitted on 27 Jul 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Habilitation à diriger des recherches

Méthodes stochastiques en analyse d'image :  
des champs de Markov aux processus  
ponctuels marqués

**Xavier Descombes**

	Albert Bijaoui	Président
	Patrick Bouthemy	Rapporteur
	Donald Geman	Rapporteur
Jury :	Christine Graffigne	Rapporteur
	Michel Schmitt	Rapporteur
	Marc Berthod	Examineur
	Henri Maître	Examineur
	Josiane Zerubia	Examineur

Soutenue le 2 février 2004



*A Natacha, Sonia et Nadia*



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Les champs de Markov</b>	<b>9</b>
2.1	Généralités : A priori ou modèle de texture . . . . .	9
2.1.1	Définitions . . . . .	9
2.1.2	Modèle de Potts et attache aux données gaussienne . . . . .	11
2.1.3	Les briques nécessaires . . . . .	12
2.2	Quelques contributions . . . . .	13
2.2.1	Estimation des classes : HVC et “K-means” entropique . . . . .	13
2.2.2	Modèle a priori : le chien modèle . . . . .	20
2.2.3	Estimation des paramètres : un algorithme de type MVMCMC . . . . .	34
2.2.4	Optimisation : Equations Différentielles Stochastiques . . . . .	39
2.3	Quelques applications traitées . . . . .	54
2.3.1	La texture . . . . .	54
2.3.2	La segmentation . . . . .	70
2.3.3	La restauration . . . . .	95
2.3.4	Champs de Markov sur graphes . . . . .	112
2.3.5	Shape from Shading . . . . .	128
2.4	Limites des champs de Markov . . . . .	135
<b>3</b>	<b>Les Processus Ponctuels Objets</b>	<b>137</b>
3.1	Généralités . . . . .	137
3.2	Extraction du ponctuel . . . . .	150
3.3	Extraction du linéique . . . . .	157
3.3.1	Le Candy-modèle . . . . .	157
3.3.2	Estimation des paramètres . . . . .	165
3.4	Extraction de surfacique . . . . .	167
3.4.1	Extraction des houppiers . . . . .	167
3.4.2	Extraction des bâtiments . . . . .	186
3.4.3	Segmentation . . . . .	205
3.5	Conclusion sur les Processus Ponctuels Marqués . . . . .	208
<b>4</b>	<b>Conclusion et perspectives</b>	<b>209</b>



# Chapitre 1

## Introduction

Ce mémoire résume mes travaux sur la décennie (entre 1994 et 2003) suivant ma thèse de doctorat qui était consacrée aux champs de Markov en analyse d'image et fut soutenue en décembre 1993. Ces travaux ont vu le jour dans différents laboratoires d'accueil :

- Laboratoire Image - Télécom Paris
- Université Catholique de Louvain (K.U.Leuven)
- Projet Pastis - INRIA Sophia Antipolis
- Institut Max Planck pour les Neurosciences - Leipzig
- Projet Ariana - INRIA Sophia Antipolis

La diversité des laboratoires d'accueil a permis de traiter de plusieurs applications sur différents types d'images parmi lesquelles :

- La binarisation d'empreintes digitales
- La segmentation d'IRM du cerveau
- La restauration d'angiographies cérébrales
- La détection des régions d'activation en IRM fonctionnelle
- La détection des espaces de Virchow Robin à partir d'IRM
- La segmentation d'images satellitaires
- L'extraction et la caractérisation des tissus urbains
- L'extraction des réseaux routiers
- La reconstruction du bâti

Du point de vue méthodologique, ces travaux s'inscrivent dans le cadre des méthodes stochastiques pour l'analyse d'image. Une première partie traite des champs de Markov et s'inscrit dans la lignée de mes travaux de thèse. Nous apportons ici quelques contributions à la modélisation, à l'estimation des paramètres, et à l'optimisation qui représentent les trois piliers d'une approche par champs de Markov en traitement des images. Les champs de Markov sont alors déclinés sur un grand nombre d'applications. Nous montrons ainsi la généralité de l'approche markovienne tant sur le plan des applications abordées (texture, segmentation, restauration,...) que sur celui des images traitées (IRM, IRMf, satellitaires optique et radar,...). Cette première partie se conclut sur les limites des champs de Markov. Nous constatons la difficulté d'intégrer des contraintes géométriques fortes à partir d'interactions locales sur des variables atomiques pixeliques. Cette limite prend toute son importance pour traiter des images à haute ou très haute résolution. La notion d'objet est alors un apport essentiel pour la modélisation géométrique des structures recherchées en analyse d'image. Cette notion d'objet n'est pas totalement incompatible avec



les approches markoviennes si nous considérons les champs de Markov sur graphe. Dans ce cas, les objets peuvent être représentés par les noeuds du graphe. A ces noeuds sont associées des attributs aléatoires modélisant la géométrie de l'objet. Les arêtes du graphe permettent de modéliser des interactions entre objet. Le problème est alors de définir le graphe sur lequel est défini le champ de Markov. En outre, cela requiert la connaissance du nombre d'objets et de la localisation des interactions (donc implicitement la localisation relative des objets). Ce type d'approche a donc essentiellement été proposé pour obtenir des algorithmes multi-échelle ou encore pour la sélection d'objets parmi un ensemble de candidats. Malheureusement, pour un grand nombre de problèmes, le nombre d'objets et leur inter-relations ne sont pas connus. Cela nécessite donc un modèle permettant de gérer un nombre aléatoire d'objets, tout en contrôlant leur géométrie et leur localisation. La deuxième partie de ce manuscrit est consacrée à de tels modèles. Ce sont les processus ponctuels marqués qui consistent à définir une densité relativement à la mesure de Poisson. Chaque point représente un objet sous-jacent dont la géométrie est dérivée de la valeur des marques du points. La densité permet de définir des interactions entre objets voisins. Nous montrons différentes applications des processus ponctuels marqués pour des problèmes d'analyse d'image. Ces applications sont classées suivant le type d'objets recherchés (ponctuels, linéaires ou surfaciques). Nous montrons le potentiel de cette approche et sa généralité. Nous montrons également qu'une implantation pratique des algorithmes d'optimisation associés à de tels modèles est faisable avec un temps de calcul qui, bien que plus lourd que pour les méthodes markoviennes, reste raisonnable.

## Chapitre 2

# Les champs de Markov

### 2.1 Généralités : A priori ou modèle de texture

#### 2.1.1 Définitions

**Introduction aux champs de Markov-** Soit  $S \subset \mathcal{Z}^v$  la trame de l'image. Chaque site  $s \in S$  prend ses valeurs  $x_s$  dans l'espace d'états  $\Lambda$ . Nous considérons un champ aléatoire  $(\Omega = \Lambda^{\text{card } S}, \mathcal{F}, P)$ . Ce champ est dit markovien si et seulement si :

$$\forall X = (x_s)_{s \in S} \in \Omega, \quad P(X) > 0 \quad (2.1)$$

$$\forall X = (x_s)_{s \in S}, \forall s \in S, \quad P(x_s | x_t, t \in S - \{s\}) = P(x_s | x_t, t \in V_s) \quad (2.2)$$

où  $V_s$  est un ensemble fini de sites appelé voisinage de  $s$  tel que :

$$\forall s \in S, \quad s \notin V_s \quad (2.3)$$

$$\forall (s, t) \in S^2, \quad s \in V_t \Rightarrow t \in V_s \quad (2.4)$$

Le théorème d'Hammersley-Clifford permet d'écrire un champ de Markov sous la forme d'un champ de Gibbs :

$$P(X) = \frac{1}{Z} \exp - \sum_{c \in \mathcal{C}} V_c(x_s, s \in c), \quad (2.5)$$

où  $Z$  est la constante de normalisation appelée fonction de partition,  $\mathcal{C}$  est l'ensemble des cliques  $c$ , une clique étant un ensemble fini de sites deux à deux voisins et  $V_c$  est un potentiel agissant sur la clique  $c$ .

Le premier domaine d'application des champs de Markov concerne la modélisation et l'analyse de textures [CJ83, Gra87, Gim99]. Une texture est, dans ce cadre, considérée comme la réalisation d'un champ de Markov. Les paramètres du modèle (cliques et potentiels) sont alors caractéristiques de la texture considérée. Un voisinage étendu et une formulation générale des potentiels permet donc de modéliser fidèlement un grand nombre de textures réelles.

De nombreuses applications considèrent des modèles moins généraux pour pouvoir dériver des estimateurs des différents paramètres qui soient robustes et précis. Le but alors n'est pas tant de modéliser fidèlement une texture mais d'extraire un paramètre discriminant pour caractériser un objet ou un milieu par rapport à son environnement. Nous verrons par la suite ce type d'approche pour l'extraction et la caractérisation de milieux urbains.

Le second grand domaine concerne l'utilisation des champs de Markov comme modèles a priori pour résoudre des problèmes inverses tels que la segmentation ou la restauration d'image [Bes74, GG84, CJ93]. On se place ici dans un cadre bayésien. Si  $Y$  représente les données, on cherche alors à optimiser la loi a posteriori  $P(X|Y)$ , ce qui revient après application de la règle de Bayes à optimiser le

produit  $P(X)P(Y|X)$ . Le modèle a priori  $P(X)$  permet d'injecter des connaissances, telles que des propriétés d'homogénéité spatiale, sur la solution. Ce terme est souvent modélisé par un champ de Markov. Nous donnons dans le paragraphe suivant un exemple classique de cette approche pour la segmentation d'image. Le terme de vraisemblance  $P(Y|X)$  reflète les connaissances que l'on a sur les données (modèle du capteur, modèle des classes). Ce terme peut être vu comme un champ externe non homogène.

**Simulation et optimisation-** L'utilisation des champs de Markov requiert des outils de simulation. Dans le cadre des modèles de texture, nous avons besoin de générer des réalisations du modèle pour l'analyse comme pour la synthèse. Pour résoudre un problème inverse, nous devons optimiser la loi a posteriori c'est à dire estimer la réalisation qui maximise un certain critère, le critère le plus répandu étant le Maximum A Posteriori (MAP) qui consiste à maximiser la loi elle-même. Pour ce faire l'outil traditionnel est l'approche MCMC c'est à dire les méthodes de Monte Carlo par chaînes de Markov [Rob96]. L'algorithme de Metropolis-Hastings est, parmi ces méthodes, une des plus populaires. Cet algorithme itératif consiste à proposer une modification de la configuration courante suivant un certain noyau de proposition, et à accepter cette nouvelle configuration avec une certaine probabilité dépendant du noyau de proposition et du modèle  $\pi(X)$  à simuler :

**Algorithme 1. Metropolis-Hastings**

1 Initialiser la configuration  $X^{(0)}$  de façon aléatoire,  $n = 0$

2 Pour chaque pixel  $s$  de l'image :

– Tirer une nouvelle valeur  $new$  pour  $x_s$  ( $x_s$  ayant la valeur courante  $cur$ ) suivant la loi :

$$Q(X = (x_s = cur, x_t, t \neq s) \rightarrow X' = (x_s = new, x_t, t \neq s))$$

– Accepter cette nouvelle valeur avec la probabilité :

$$R = \min \left( 1, \frac{Q(X' \rightarrow X) \pi(X')}{Q(X \rightarrow X') \pi(X)} \right)$$

3 Si convergence stop sinon  $n \leftarrow n + 1$  et retour en 2.

Les deux algorithmes les plus répandus sont l'algorithme de Metropolis pour lequel :

$$Q(X = (x_s = cur, x_t, t \neq s) \rightarrow X' = (x_s = new, x_t, t \neq s)) = \frac{1}{\text{card}(\Lambda)}$$

et l'échantillonneur de Gibbs pour lequel les sites sont visités dans l'ordre lexicographique et :

$$Q(X = (x_s = cur, x_t, t \neq s) \rightarrow X' = (x_s = new, x_t, t \neq s)) = \pi(x_s | x_t, t \in V_s).$$

Pour obtenir le MAP, dans le cadre d'une optimisation, l'algorithme d'échantillonnage est intégré dans un recuit simulé :

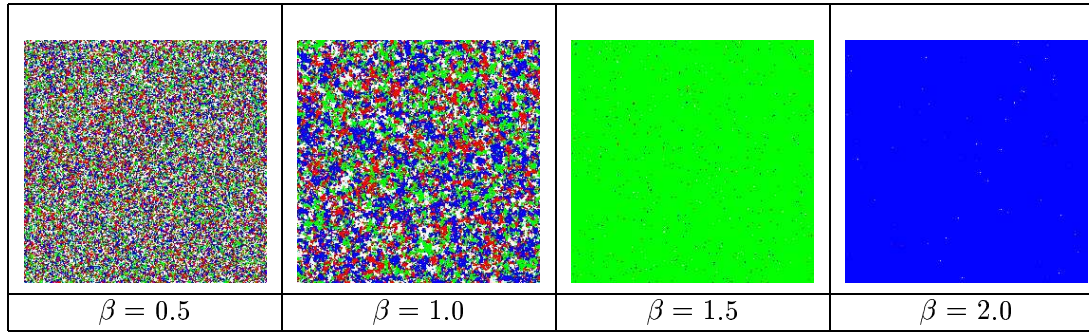
**Algorithme 2. Recuit simulé**

1 Initialiser la configuration  $X^{(0)}$  de façon aléatoire et la température  $T_0$ ,  $n = 0$

2 Faire une itération sur toute l'image de l'algorithme d'échantillonnage pour la loi  $\pi(X)^{1/T}$

3 Si convergence stop sinon  $n \leftarrow n + 1$ ,  $T_{n+1} = f(T_n)$  et retour en 2.

où  $f(T)$  est une loi de décroissance de la température (logarithmique en théorie, souvent géométrique en pratique).



TAB. 2.1 – Simulations du modèle de Potts

### 2.1.2 Modèle de Potts et attache aux données gaussienne

Nous développons ici un modèle classique pour la segmentation d'image en nous plaçant dans un cadre bayésien.

**Segmentation par un modèle de Potts-** Le modèle de Potts est largement utilisé comme a priori à des fins de segmentation [DE87, SR92]. Il permet, en effet, d'imposer des contraintes d'homogénéité spatiale sur l'image segmentée. C'est un modèle à interactions par paires. Un système de voisinage est défini (par exemple 4 ou 8 connexité). Les cliques sont alors constituées de paires de pixels voisins. La fonction potentielle pénalise alors les cliques non homogènes, c'est à dire les cliques dont les sites ont des étiquettes différentes :

$$P(X) = \frac{1}{Z} \exp - \left[ \sum_{c=\{s,t\} \in \mathcal{C}} \beta \delta(x(s) \neq x(t)) \right], \quad (2.6)$$

où  $X = (x(s), s \in S)$  est l'image segmentée,  $\mathcal{C}$  est l'ensemble des cliques,  $\beta$  le paramètre d'interaction et  $Z$  la fonction de partition.

Le tableau 2.1 montrent des simulations de modèles de Potts 4 connexe pour différentes valeurs du paramètres d'interaction. Pour de faibles valeurs de  $\beta$ , nous obtenons des configurations bruitées. A mesure que les interactions augmentent l'image se structure mais les configurations restent néanmoins bruitées. A partir d'une certaine valeur critique, un phénomène de transition de phase provoque la prédominance d'une classe sur les autres. Pour de fortes interactions, nous obtenons des configurations uniformes. Notons que le phénomène de transition de phase apparaît avant la disparition du bruit. L'homogénéité spatiale peut donc être obtenue avec une image uniforme. *En revanche, sans champ externe, les configurations constituées de plusieurs régions homogènes ne peuvent pas être obtenues avec le modèle de Potts.*

Pour définir la vraisemblance, nous effectuons l'hypothèse usuelle d'indépendance conditionnelle, ce qui permet d'écrire :

$$P(Y|X) = \prod_{s \in S} p(y(s)|x(s)). \quad (2.7)$$

Nous supposons que l'image segmentée contient  $n$  classes gaussiennes de paramètres  $(\mu_i, \sigma_i), i = 1, \dots, n$ . Nous avons alors :

$$p(y(s)|x(s) = i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp - \frac{(y(s) - \mu_i)^2}{2\sigma_i^2}. \quad (2.8)$$

Le modèle a posteriori est alors proportionnel au produit du modèle a priori et de la vraisemblance :

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \propto P(Y|X)P(X). \quad (2.9)$$

L'image segmentée est alors la configuration qui maximise l'a posteriori au sens d'un certain critère. Le critère le plus répandu est celui du Maximum A Posteriori, consistant à maximiser la loi a posteriori. Il peut être obtenu par un schéma de recuit simulé.

Dans la suite de ce document, nous comparerons le modèle de Potts avec un modèle a priori plus complexe. Nous verrons notamment que, si le modèle de Potts a de bonnes propriétés de régularisation (suppression du bruit dans une carte de segmentation), il a tendance à trop lisser les contours des objets, voire à supprimer les petits objets de forme allongée.

### 2.1.3 Les briques nécessaires

L'utilisation des champs de Markov pour résoudre un problème d'analyse d'image nécessite donc la mise au point de plusieurs briques. Nous pouvons séparer les différentes tâches comme suit :

- **Modélisation-** Cela concerne la définition du modèle a priori dans les approches bayésiennes ou celle du modèle de texture. Il s'agit de définir les cliques et les potentiels associés. Nous avons travaillé essentiellement sur des modèles markoviens gaussiens en ce qui concerne la texture, comme nous le verrons dans les paragraphes consacrés aux applications. Pour ce qui est des modèles a priori, notre contribution porte essentiellement sur le Chien modèle décrit dans le paragraphe 2.2.2. Dans les approches bayésiennes, cela concerne également la construction de la vraisemblance. Il s'agit de modéliser le capteur (restauration, segmentation) et de définir les classes et leurs paramètres (segmentation).
- **Estimation des paramètres-** Nous considérons ici l'estimation des paramètres associés à un modèle de texture ou au modèle a priori dans les approches bayésiennes. Pour ce qui est des textures, nous avons essentiellement traité de l'analyse urbaine (paragraphe 2.3.1). Nous avons également proposé un algorithme d'estimation des paramètres associés aux champs de Markov dont l'énergie dépend linéairement de ses paramètres (paragraphe 2.2.3). Un autre aspect est l'estimation des paramètres de la vraisemblance. Notre contribution est constituée d'une approche permettant de distinguer différentes classes dans un cadre fortement mélangé (paragraphe 2.2.1) et l'extension d'algorithmes classiques de classification ("K-means" et "Fuzzy C-means") intégrant l'estimation du nombre de classes.
- **Simulation, Optimisation-** Une fois le modèle défini, les paramètres estimés ou établis empiriquement par calibration, le résultat recherché est une configuration qui optimise un certain critère. Le critère le plus répandu est le Maximum A Posteriori, c'est à dire la configuration qui maximise la probabilité a posteriori. Pour obtenir cette configuration nous avons recours à un algorithme d'optimisation. Cette optimisation peut être optimale globalement comme pour le recuit simulé ou localement comme pour l'ICM. Ces algorithmes font appel à la théorie des chaînes de Markov comme par exemple les algorithmes MCMC. Ils sont fondés sur la simulation de la loi a posteriori éventuellement modifiée par un paramètre de température. La simulation d'une telle loi se fait par des algorithmes du type Metropolis-Hasting ou échantillonneur de Gibbs. La recherche d'outils de simulations plus efficaces en terme de coût en temps de calcul est un domaine encore ouvert. Nous avons entamé une étude sur une approche alternative aux méthodes MCMC. Cette approche est fondée sur la dynamique de Langevin qui permet de simuler des équations différentielles stochastiques convergeant vers une mesure donnée, par exemple un modèle markovien (voir paragraphe 2.2.4).

## 2.2 Quelques contributions

### 2.2.1 Estimation des classes : HVC et “K-means” entropique

#### Histogramme de la Variance Conditionnelle (HVC)

<sup>1</sup> Nous abordons le problème de l’estimation du nombre de classes et de leurs paramètres dans le cadre de la segmentation d’image. Nous supposons que les différentes classes sont définies à partir des niveaux de gris uniquement, c’est à dire que nous ne tenons pas compte de critères texturaux. Nous considérons également que nous avons une seule image de données. Nous proposons un algorithme pour estimer le nombre des classes et les paramètres qui leur sont associées. Le problème se ramène alors à l’estimation des composantes d’un mélange de distributions.

Une première approche consiste à rechercher les modes de l’histogramme de l’image. Néanmoins, cette approche peut conduire à une mauvaise classification dans le cas de distributions fortement mélangées. Supposons, en effet, que deux classes de l’image aient des moyennes proches l’une de l’autre, comparativement à leurs variances. Ces distributions seront alors représentées par un unique mode dans l’histogramme. Dans ce cas, un algorithme de détection de modes ne séparera pas ces deux classes. Nous montrons que, moyennant une propriété contextuelle inhérente aux images, l’utilisation d’une variance conditionnelle (variance de la loi des voisins d’un pixel conditionnellement à sa valeur) permet une meilleure séparation des modes.

Considérons un mélange de deux distributions  $P_1$  et  $P_2$ , dont les moyennes (resp. variances) sont notées  $\mu_1$  et  $\mu_2$  (resp.  $\sigma_1^2$  et  $\sigma_2^2$ ). Supposons que la probabilité de la classe 1 soit donnée par  $\pi$  et, par conséquent, la probabilité de la classe 2 par  $1 - \pi$ . Nous faisons l’hypothèse contextuelle suivante : *les voisins d’un pixel appartiennent à la même classe que ce pixel*. Cette hypothèse est vérifiée partout sauf sur les contours. Si nous ajoutons l’hypothèse classique concernant l’indépendance des pixels conditionnellement à leur classe, nous pouvons montrer que la variance conditionnelle  $\Sigma^2(y)$ , correspondant à la variance des pixels voisins d’un pixel ayant pour valeur  $y$ , peut s’écrire :

$$\Sigma^2(y) = A_1(y)\sigma_1^2 + A_2(y)\sigma_2^2 + A_1(y)A_2(y)(\mu_1 - \mu_2)^2. \quad (2.10)$$

où :

$$A_1(y) = \frac{\pi P_1(y)}{\pi P_1(y) + (1 - \pi)P_2(y)} \quad (2.11)$$

$$A_2(y) = \frac{(1 - \pi)P_2(y)}{\pi P_1(y) + (1 - \pi)P_2(y)} \quad (2.12)$$

Cette fonction est une parabole dont le maximum est compris entre  $\mu_1$  et  $\mu_2$ . Il permet de détecter la zone de conflit entre les deux classes (voir figure 2.1).

**Détection des modes par une analyse par espace d’échelle :** Nous présentons ici une méthode qui est notamment utilisée pour mettre en correspondance une courbe avec une somme de gaussiennes [GO94]. Un mode sur une courbe est défini comme un maximum local. Nous proposons d’utiliser une analyse par espace d’échelle, connue sous l’acronyme anglais SSA (“Scale Space Analysis”) pour détecter les modes de l’histogramme de la variance conditionnelle (HVC), qui représentent les zones de conflits entre classes. Cette méthode, proposée par Witkin [Wit84] a été appliquée pour détecter les modes de l’histogramme d’une image [Car87]. Considérons une séquence de données monovaluées et uniformément espacées, par exemple un histogramme. Considérons alors la convolution de ce signal par un noyau gaussien dont la variance varie dans un intervalle donné. Chaque convolution est alors associée à une résolution. A chaque résolution, nous calculons les points de passage à zéro de la dérivée seconde, qui

1. Nous résumons ici des travaux effectués à l’université catholique de Louvain (KUL). Bibliographie : [22]

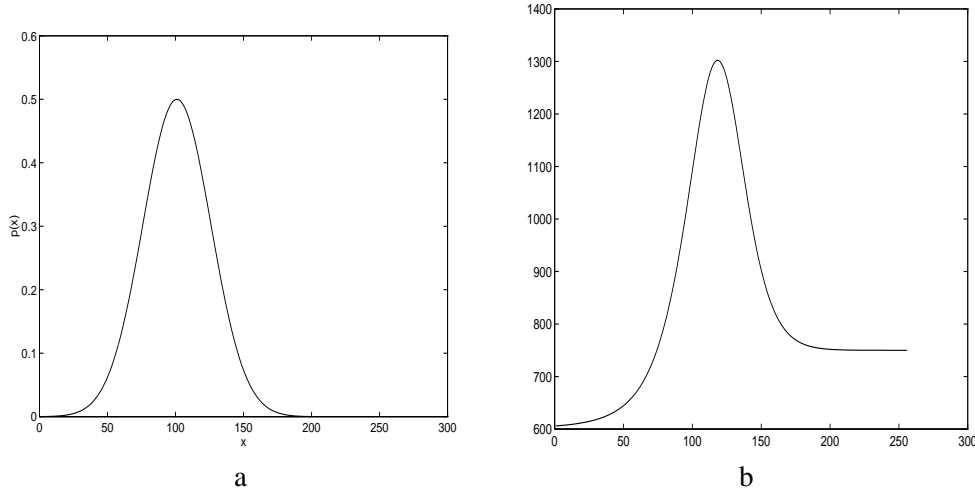


FIG. 2.1 – a) Somme de deux gaussiennes  $(\mu_1, \sigma_1) = (100, 600)$   $(\mu_2, \sigma_2) = (150, 750)$  et b) variance conditionnelle  $\Sigma^2(y)$

correspondent aux points d'inflexion. Sur l'histogramme d'une image, deux points d'inflexions successifs (passage négatif-positif de la dérivée seconde puis passage positif-négatif) permettent de définir une classe. Les intervalles entre les classes définissant les zones de conflit. Cette méthode a été utilisée pour définir automatiquement le terme d'attache aux données d'une modélisation markovienne, pour segmenter l'ALD (Adrénoleukodystrophy) à partir d'échos IRM (Images à Résonance Magnétique) [1].

Pour illustrer notre propos, considérons une distribution gaussienne, caractérisée par sa moyenne  $\mu$  et sa variance  $\sigma^2$  (voir figure 2.2.a). Si nous convoluons cette distribution par un filtre gaussien centré, caractérisé par sa variance  $\sigma_f^2$ , nous obtenons une gaussienne, dont la moyenne est donnée par  $\mu$  et la variance par  $\sqrt{\sigma^2 + \sigma_f^2}$ . Ainsi, lorsque la variance du filtre varie, le lieu des points  $y$  où la dérivée seconde s'annule est défini de la façon suivante :

$$y = \mu \pm \sqrt{\sigma^2 + \sigma_f^2}. \quad (2.13)$$

Dans ce cas, l'espace d'échelle est donc constitué de deux courbes divergentes (voir figure 2.2.b).

Considérons maintenant une distribution définie par la somme de deux gaussiennes (voir figure 2.2.c). La convolution par un noyau gaussien reste la somme de deux gaussiennes. Si les deux gaussiennes sont suffisamment séparées, nous obtenons une courbe bimodale et la dérivée seconde possède quatre zéros. Dans le cas contraire, la dérivée seconde ne présente que deux zéros. En haut de l'espace d'échelle (pour de faibles variances du noyau gaussien), nous avons donc quatre points. Lorsque  $\sigma_f^2$  croît, les deux zéros centraux se rapprochent et finissent par se confondre pour disparaître (voir figure 2.2.d). L'espace d'échelle de la somme de deux gaussiennes consiste donc en deux courbes divergentes et une arche entre les deux. Notons que si les deux gaussiennes initiales sont fortement mélangées, nous obtenons uniquement les deux courbes divergentes.

Cette SSA peut donc être utilisée pour estimer le nombre de modes dans un mélange. De plus, la localisation des arches permet d'obtenir une première estimation des paramètres des distributions associées aux modes et peut donc être utilisée comme initialisation pour un algorithme de mise en correspondance avec une somme de gaussiennes, tel que l'algorithme de Marquart.

Nous proposons d'utiliser cette technique pour détecter les différents modes d'un HVC. Nous n'avons pas besoin d'une hypothèse gaussienne mais supposons simplement que chaque mode est caractérisé par un point d'inflexion de chaque côté. L'utilisation d'une SSA permet d'éviter de mauvaises détectations

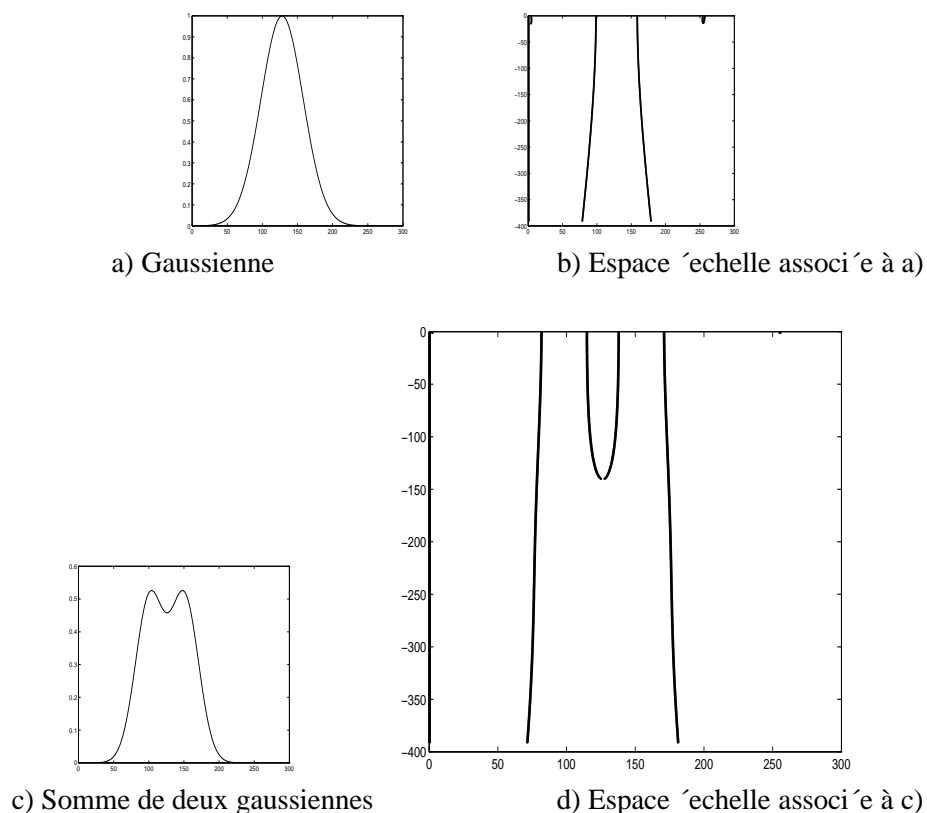


FIG. 2.2 – Gaussiennes et espaces échelle associés

dues au bruit. Considérons l'espace d'échelle associée avec un HVC donné. A partir de la convolution du HVC par un noyau gaussien de variance  $\sigma_f^2$ , nous calculons l'ensemble des points où la dérivée seconde s'annule. A partir de cet ensemble, nous définissons les sous-ensembles de niveaux de gris correspondant aux classes et aux zones de conflit entre classes.

L'algorithme se définit alors de la façon suivante : à partir des données, nous estimons d'abord la variance  $\Sigma^2(y)$  de la moyenne des voisins de  $s$  comme une fonction de  $x$ , où  $x$  représente le niveau de gris du pixel  $s$ . En pratique, le HVC est constituée de paliers plus ou moins constants, correspondant aux différentes classes, séparées par des modes de valeurs plus élevées, correspondant aux zones de conflit. Nous détectons alors les modes du HVC en utilisant une SSA. En pratique, nous convoluons le signal par un unique noyau gaussien. La variance de ce noyau est un paramètre donné par l'utilisateur. Plus cette variance est grande, moins nous détectons de classes. La valeur de ce paramètre dépend donc de la finesse de la classification désirée. Pour finir, nous estimons les passages par zéros de la dérivée seconde du résultat de la convolution. A partir de deux zéros définissant un mode, nous définissons une zone de conflit, ce qui permet ensuite de définir les différentes classes. Le domaine de validité de cette approche a été étudié dans le cas d'un mélange de deux gaussiennes et s'est avéré plus étendu que le domaine correspondant à la présence de deux modes dans l'histogramme.

**Validation avec une segmentation par une estimation itérative conditionnelle :** Pour étudier les performances de la méthode d'estimation proposée, nous l'insérons dans un algorithme de type EM ("expectation-maximisation"). Le principe de cet algorithme est d'alterner des séquences d'optimisation du modèle et d'estimation de ses paramètres. Ainsi, une initialisation de l'estimation est affinée après chaque étape d'optimisation, qui prend en compte l'information de type contextuel par l'intermédiaire



	HVC	it. 1	it. 2	it. 3	V. R.
$\mu_1$	70	84	84	84	85
$\sigma_1^2$	333	477	497	498	500
$\mu_2$	107	113	114	114	115
$\sigma_2^2$	233	439	487	489	500
$\mu_3$	147	145	144	144	145
$\sigma_3^2$	300	522	550	553	500
$\mu_4$	185	175	174	174	175
$\sigma_4^2$	400	497	504	504	500

TAB. 2.2 – Estimations des paramètres associées aux classes de l’image de synthèse #1 lors d’un ICE initialisé par un HVC pour le modèle de Potts. Les valeurs réelles sont portées dans la colonne V.R

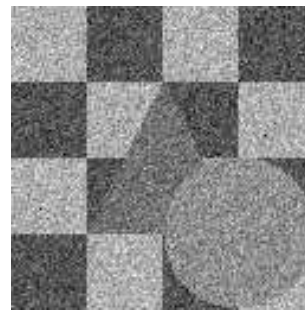
du terme de régularisation. Nous appliquons ici le principe de l’estimation itérative conditionnelle (ICE) :

- 1 Initialisation des paramètres de l’attache aux données par la méthode HVC par les estimateurs empiriques des moyennes et variances,
- 2 Générer  $n$  réalisations du champ des labels par un échantillonneur de Gibbs avec l’estimée courante des paramètres,
- 3 Actualiser les paramètres des classes conditionnellement au champ des labels,
- 4 Retourner à l’étape 2 jusqu’à ce que les paramètres associés aux classes soient stabilisés.

Nous considérons une image de synthèse pour laquelle nous connaissons donc le nombre de classes ainsi que leur moyenne et leur variance (voir figure 2.3.a). Cette image est bruitée par un bruit additif gaussien (voir figure 2.3.b).



a) Image de synthèse numéro 1  $256 \times 256$



b) Image a bruitée : SNB = 5dB

FIG. 2.3 – Image test de synthèse

Les résultats sont montrés avec le modèle de Potts comme *a priori*. La table 2.2 montre les valeurs des paramètres estimés pour chaque étape de l’ICE. Les résultats obtenus sur les images de synthèse prouvent les bonnes performances de la méthode de l’HVC. Une itération de la phase d’estimation est suffisante pour la convergence des estimées, ce qui revient à utiliser deux itérations de la phase d’optimisation.



FIG. 2.4 – Résultats obtenus pour l'image de synthèse # 1 avec un ICE initialisé par un HVC et un modèle de Potts

### “Cmeans” entropique

<sup>2</sup> Nous développons ici un algorithme de classification non-supervisé. Cet algorithme est une variation du “Fuzzy Cmeans” comprenant un terme permettant d’estimer le nombre de classes [Bez82, KK94]. Une première version de cette idée appliquée au “Kmeans” est développée dans [61] et sera reprise dans le paragraphe 2.3.3. Cette approche entre dans un cadre général où une fonctionnelle, contenant un terme de fidélité aux données à classer et un terme pénalisant la complexité de la représentation, est minimisée. Différents critères, parmi lesquels le critère AIC [Aka74, BL91], la compacité du nuage de points [HJ87, NC93] où le *Minimum Description Length* [Ris78] ont été proposés. Nous proposons ici un terme de type entropie.

**Algorithme du “Cmeans” entropique-** Nous considérons le cadre des ensembles flous pour définir les classes [Zad65]. Chaque ensemble (classe) est donc caractérisé par une fonction continue à valeurs dans  $[0,1]$ , c’est à dire que les pixels appartiennent aux différentes classes suivant un degré d’appartenance dans  $[0,1]$ . La classification est donc donnée par une matrice de partition  $\mathbf{U} = [u_{ij}]$ , où  $u_{ij}$  représente le degré d’appartenance du pixel  $i$  à la classe  $j$ . Chaque pixel est caractérisé par un vecteur de paramètres contenant l’information radiométrique et/ou texturale. La décision finale concernant la classification est prise une fois la convergence de l’algorithme obtenue en considérant pour chaque pixel la classe de plus fort degré d’appartenance.

Nous définissons une fonction de coût permettant d’estimer le nombre optimal de classes relativement à un critère entropique. Le but étant de représenter l’information avec un minimum de classes nous minimisons l’entropie de l’histogramme de l’image classifiée. Cela revient à considérer a priori suivant :

$$\sum_{i=1}^C p_i \log(p_i)$$

où  $p_i$  est la probabilité de la classe  $i$  et  $C$  le nombre de classes. Nous nous plaçons dans un cadre probabiliste. La probabilité a priori du pixel  $j$  est notée  $p_j$ . Si  $N$  est le nombre de pixels, nous avons  $p_j = \frac{1}{N}$ . Ce qui nous donne :  $p_i = \sum_{j=1}^N p_j p(j \in i)$ . Nous interprétons  $u_{ij}$  comme la probabilité que le pixel  $j$  appartienne à la classe  $i$ . Nous avons donc la contrainte suivante :  $\sum_{i=1}^C u_{ij} = 1, \forall j$ . La probabilité a priori de la classe  $i$  est donc donnée par :

$$p_i = \frac{1}{N} \sum_{j=1}^N u_{ij}. \quad (2.14)$$

La fonction de coût considérée est alors la suivante :

$$J = \sum_{i=1}^C \sum_{j=1}^N (u_{ij})^m d^2(x_j, c_i) - \alpha \sum_{i=1}^C p_i \log(p_i), \quad (2.15)$$

Cette fonction est minimisée sous la contrainte  $\sum_{i=1}^C u_{ij} = 1 \quad \forall j$ .  $m \in [0, \infty]$  caractérise le degré de flou,  $c_i$  est le centroïde de la classe  $i$ ,  $x_j$  le vecteur caractéristique du pixel  $j$  et  $d^2(x_j, c_i)$  la distance euclidienne entre le vecteur  $x_j$  et le vecteur  $c_i$ . Après plusieurs tests, nous choisissons empiriquement  $m = 2$ .  $C$  est le nombre initial de classes. Ce nombre diminue jusqu’à  $C_{opt}$  au cours de l’algorithme (classes vides). Par conséquent la méthode est non supervisée en considérant  $C$  égal au nombre de niveaux de gris. Le premier terme de  $J$  reflète l’homogénéité des classes et correspond

<sup>2</sup>. Nous résumons ici des travaux initiés à l’Institut Max Planck de Leipzig en collaboration avec G. Palubinskas et étendus dans la thèse d’A. Lorette à l’INRIA. Bibliographie : [61, 54, 56]

à l'algorithme du "Cmeans". Ce terme est minimal lorsque nous avons une classe par niveau de gris. Le second terme, terme d'entropie, est minimal quand tous les pixels appartiennent à la même classe (les autres classes étant vides).

Nous minimisons  $J$  alternativement par rapport à  $c_i$  et par rapport à  $u_{ij}$ . La contrainte est intégrée par des multiplicateurs de Lagrange, ce qui conduit au critère suivant :

$$J = \sum_{i=1}^C \sum_{j=1}^N (u_{ij})^2 d^2(x_j, c_i) - \alpha \sum_{i=1}^C p_i \log(p_i) - \sum_{j=1}^N \lambda_j \left( \sum_{i=1}^C u_{ij} - 1 \right). \quad (2.16)$$

Les équations de mise à jour du processus itératif sont obtenues en annulant les dérivées. Le degré d'appartenance de chaque pixel et la probabilité a priori de chaque classe varient faiblement à chaque itération, de sorte qu'à l'itération  $n_i$  et  $u_{ij}$  soient mis à jour de la manière suivante :

$$c_i^{(n)} = \frac{\sum_{j=1}^N [u_{ij}^{(n-1)}]^2 x_j}{\sum_{j=1}^N [u_{ij}^{(n-1)}]^2}, \quad (2.17)$$

$$u_{ij}^{(n)} = \frac{\frac{1}{d^2(x_j, c_i^{(n)})}}{\sum_{k=1}^C \frac{1}{d^2(x_j, c_k^{(n)})}} + \frac{\alpha \left( 1 + \log(p_i^{(n-1)}) - \frac{\sum_{k=1}^C \frac{1 + \log(p_k^{(n-1)})}{d^2(x_j, c_k^{(n)})}}{\sum_{k=1}^C \frac{1}{d^2(x_j, c_k^{(n)})}} \right)}{2N d^2(x_j, c_i^{(n)})}. \quad (2.18)$$

$\alpha$  décroît à chaque itération de sorte que durant les premières itérations le terme d'entropie permette de réduire le nombre de classes vers une valeur proche de  $C_{opt}$  et qu'ensuite la classification ne soit pas biaisée par ce terme.  $\alpha$  décroît de la façon suivante :

$$\alpha(n) = \alpha(0) \exp\left(-\frac{n}{\tau}\right), \quad (2.19)$$

où  $\tau$  est une constante de temps. La valeur initiale  $\alpha(0)$  est normalisée en fonction du bruit pour obtenir une classification indépendante du taux de bruit dans les données.  $\alpha(0)$  et  $\tau$  sont fixés une fois pour toutes et ne dépendent pas des données traitées.

Après avoir mis à jour les centroïdes et les degrés d'appartenance, nous mettons à jour les probabilités a priori des classes. Les classes dont la probabilité est inférieure à un seuil  $\epsilon$  sont supprimées.  $\epsilon$  doit être suffisamment petit (typiquement  $\epsilon = \frac{1}{N}$ ). Sa présence est due au caractère continu des degrés d'appartenance. Dans la version "Kmeans" de l'algorithme, ce seuil n'est pas nécessaire car les classes supprimées sont celles ne contenant aucun pixel.

L'algorithme complet est donc le suivant :

1. Initialisation du nombre de classes et des centroïdes de chaque classe
2. Mise à jour de la matrice de partition  $\mathbf{U}^{(n)}$  (voir équation 2.18)
3. Mise à jour des centroïdes de chaque classe (voir équation 2.17)
4. Mise à jour des probabilités a priori  $p$  de chaque classe (voir équation 2.14),  
Si  $p_i < \epsilon$ , supprimer la classe  $i$

5. si  $\|\mathbf{U}_{(n)} - \mathbf{U}_{(n-1)}\| > \xi$  alors  $n = n + 1$ , mettre à jour  $\alpha$  (voir équation 2.19) et retourner en 2. sinon aller en 6.
6. Assigner chaque pixel à la classe pour laquelle son degré d'appartenance est maximal.

**Résultats expérimentaux-** Pour réduire le temps de calcul le nombre initial de classes est réduit à 50. Cela revient à supposer que l'image contient moins de 50 classes. Pour les mêmes raisons, nous initialisons la classification par deux itérations du "fuzzy Cmeans" classique. Nous avons pris  $\tau = 10$  et  $\alpha(0) = 2$ . La figure 2.5 (a) est une image SPOT3 en zones rurales. Pour une telle image, il est bien difficile de prévoir le nombre de classes optimal. La classification que nous obtenons contient 9 classes. Le second test concerne une coupe IRM du cerveau (figure 2.6 (a)). Si les classes matière blanche, matière grise et liquide céphalo-rachidien peuvent être définies a priori, le cas du crâne et des tissus qui l'entourent est plus problématique, du fait de leur caractère très hétérogène du point de vue radiométrique. Nous obtenons un nombre de classes égal à 6 et une classification pour laquelle les classes d'intérêt sont bien représentées.

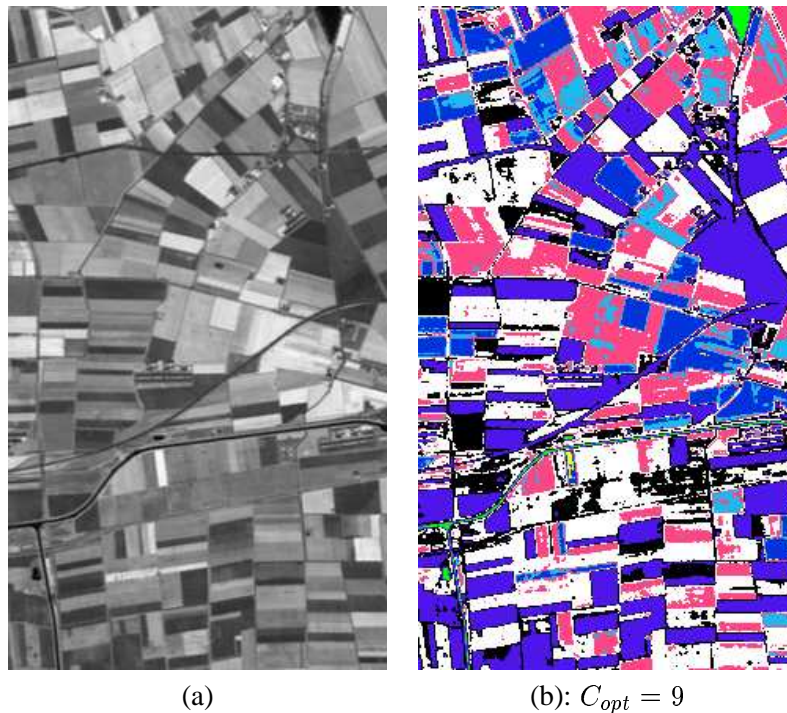


FIG. 2.5 – (a): image SPOT3 panchromatique (résolution 10 m) (b): Image classifiée.

## 2.2.2 Modèle a priori : le chien modèle

### Description du modèle

<sup>3</sup> Nous avons vu au début du chapitre 2 les limites du modèle de Potts pour la segmentation d'image. En résumé, ce modèle pénalise une composante connexe proportionnellement à la longueur de son contour. La vraisemblance, quant à elle, dépend de la surface d'un objet. Les composantes connexes de faible surface (comme le bruit) sont donc pénalisées, ce qui explique les propriétés régularisatrices du modèle. En revanche, les objets fins ou allongés (à fort rapport périmètre/surface) sont également

3. Nous résumons ici des travaux effectués à l'École Nationale Supérieure des Télécommunications en collaboration avec J.F. Mangin, E. Pechersky et M. Sigelle. Bibliographie : [19, 21]

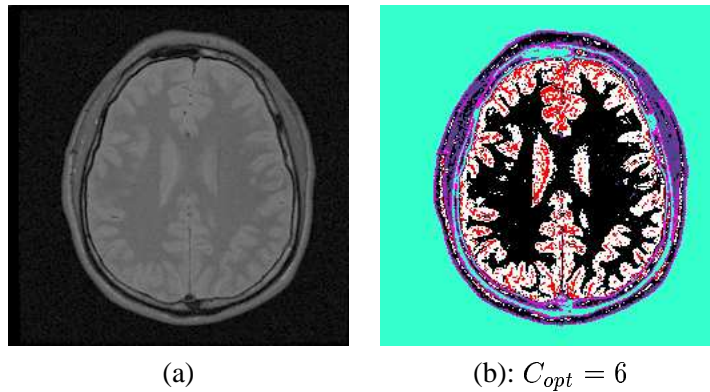


FIG. 2.6 – (a): coupe IRM du cerveau (b) Image classifiée.

pénalisées. Cela implique une sur-régularisation (perte des détails) du modèle de Potts. Nous avons proposé un modèle permettant de remédier à ce problème. Le modèle présenté dans ce paragraphe dépend de trois paramètres respectivement notés  $e$ ,  $l$  et  $n$  qui font référence aux contours (“edge”), aux lignes (“line”) et au bruit (“noise”). Le principe de construction du modèle se décompose en plusieurs étapes. Nous proposons en premier lieu un modèle binaire, étendu au cas  $m$ -aire par la suite. Nous choisissons un système de cliques : dans le cas présent, les cliques sont constituées des blocs carrés de trois pixels sur trois. Cette taille correspond à la taille minimale nécessaire à la définition de contours et de lignes. D’autre part, une taille plus grande accroît considérablement la complexité du modèle. Nous effectuons alors une classification des différentes configurations binaires possibles pour les cliques choisies. A chaque classe est alors associée un paramètre, valeur du potentiel correspondant. Le nombre des paramètres obtenus est encore trop grand pour être utilisable en pratique. En outre, le lien entre ces paramètres et les propriétés des réalisations n’est pas immédiat. La seconde étape consiste donc à relier ces paramètres avec des quantités facilement interprétables de l’image par l’intermédiaire d’une analyse des énergies locales. Des modèles similaires, fondés sur une classification des configurations  $3 \times 3$ , où chaque classe de configurations reçoit un potentiel différent, ont été proposés [WP85, CHL95]. De même, un modèle sur trame hexagonale est présenté dans [TB98]. L’originalité de notre approche tient dans la prise en compte des énergies locales induites par les contours et lignes pour différentes orientations.

**Construction du modèle binaire :** Considérons donc les  $\mathfrak{Z} = 512$  configurations binaires possibles sur une clique de 3 pixels par 3. Une première classification est obtenue en considérant que deux configurations sont équivalentes si elles se déduisent l’une de l’autre soit par la symétrie noir-blanc, soit par une rotation. Nous obtenons alors 51 classes d’équivalence dont un représentant, et le nombre d’éléments de chaque classe, sont donnés sur la figure 2.7.

A chaque classe est associée un paramètre définissant la valeur du potentiel des configurations de la classe. Ainsi, sous les hypothèses de symétrie du modèle, nous avons, pour un tel voisinage (cliques de  $3 \times 3$  pixels), cinquante et un degrés de liberté. L’énergie étant définie à une constante près, nous imposons un potentiel nul aux configurations uniformes ( $C(1) = 0$ ), ce qui ramène à cinquante le nombre de degrés de liberté. La construction du modèle consiste alors à imposer des contraintes en reliant les différents paramètres entre eux. Nous considérons que les états fondamentaux (états d’énergie minimale) sont données par les configurations uniformes. De sorte que la méthodologie de construction du modèle peut se résumer ainsi :

- Modéliser les contraintes sur les réalisations globales par les énergies locales et en déduire les équations sur les paramètres définis par la figure 2.7.
- Vérifier la cohérence du système d’équations et le résoudre.

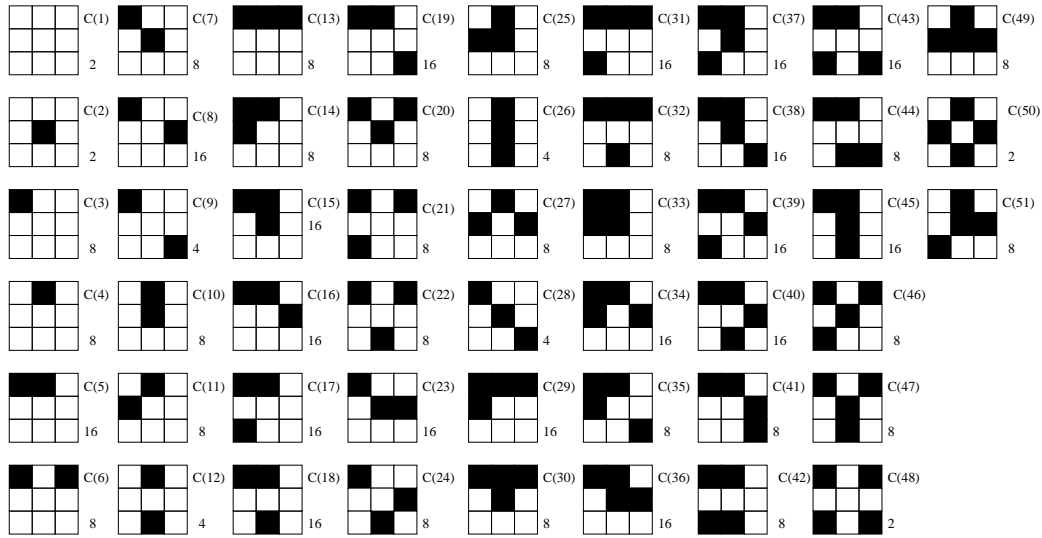


FIG. 2.7 – Classes d'équivalence des configurations binaires  $3 \times 3$  et nombre d'éléments dans chaque classe

– Vérifier qu'aucune réalisation n'est d'énergie négative.

Le premier type de contraintes concerne la pénalisation des contours. Nous imposons le coût d'un contour via une énergie par unité de longueur, notée  $e$ . Avec la taille élémentaire choisie (taille des cliques), nous avons huit directions possibles (voir figure 2.8). En considérant le modèle symétrique par rotation, à ces directions correspondent trois types de contours, à savoir vertical, oblique et diagonal. Nous imposons à chaque direction une même énergie par unité de longueur pour obtenir un modèle isotrope. Au trois types de contours sont alors associées les trois équations suivantes, représentées sur la figure 2.9 :

$$2C(13) = e \quad (2.20)$$

$$2C(3) + 2C(14) = \sqrt{2}e \quad (2.21)$$

$$C(5) + C(29) + C(3) = \frac{\sqrt{5}}{2}e \quad (2.22)$$

Ces contraintes sont définies pour les objets de largeur au moins égal à 3. Pour les autres objets, nous

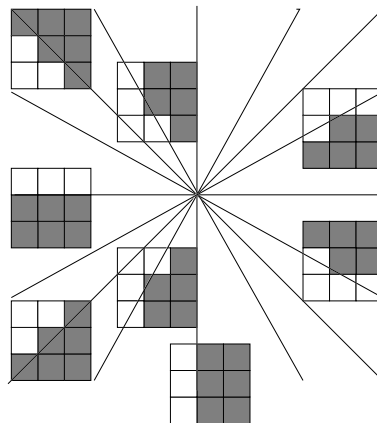


FIG. 2.8 – Orientations définies dans le chien-modèle

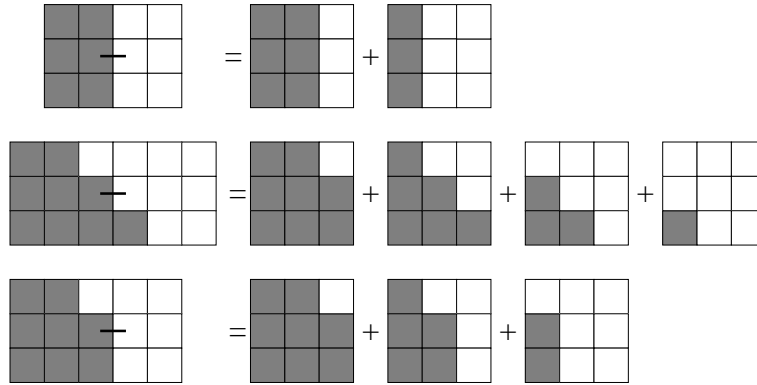


FIG. 2.9 – Equations correspondant aux contraintes de contours

d'éfinissons des contraintes similaires. Pour les objets d'épaisseur 1, nous introduisons un paramètre supplémentaire  $l$  d'éterminant l'énergie par unit'é de longueur des lignes, ce qui induit trois nouvelles équations (cf figure 2.10). Les objets d'épaisseur 2 et 3 sont d'éfinis comme ayant une énergie  $2e$  par unit'é de longueur correspondant aux bords droit et gauche. Les équations obtenues s'écrivent alors :

– pour les objets verticaux (et horizontaux) :

$$2C(13) + C(26) = l \quad (2.23)$$

$$4C(13) = 2e \quad (2.24)$$

– pour les objets diagonaux :

$$2C(3) + 2C(11) + C(28) = \sqrt{2}l \quad (2.25)$$

$$2C(3) + 2C(14) + 2C(35) = 2\sqrt{2}e \quad (2.26)$$

$$2C(3) + 4C(14) + C(9) = 2\sqrt{2}e \quad (2.27)$$

– pour les objets obliques :

$$C(3) + C(16) + C(23) + C(5) = \frac{\sqrt{5}}{2}l \quad (2.28)$$

$$C(3) + 2C(29) + C(19) + C(5) = \sqrt{5}e \quad (2.29)$$

$$2C(5) + 2C(29) + 2C(3) = \sqrt{5}e \quad (2.30)$$

Ces différentes contraintes se ramènent à onze équations faisant intervenir quatorze paramètres. Il nous faut maintenant vérifier que le système possède bien des solutions. Deux paramètres ( $C(3)$  et  $C(5)$ ) jouent un rôle particulier dans ce système puisqu'ils interviennent dans la plupart des équations. Nous



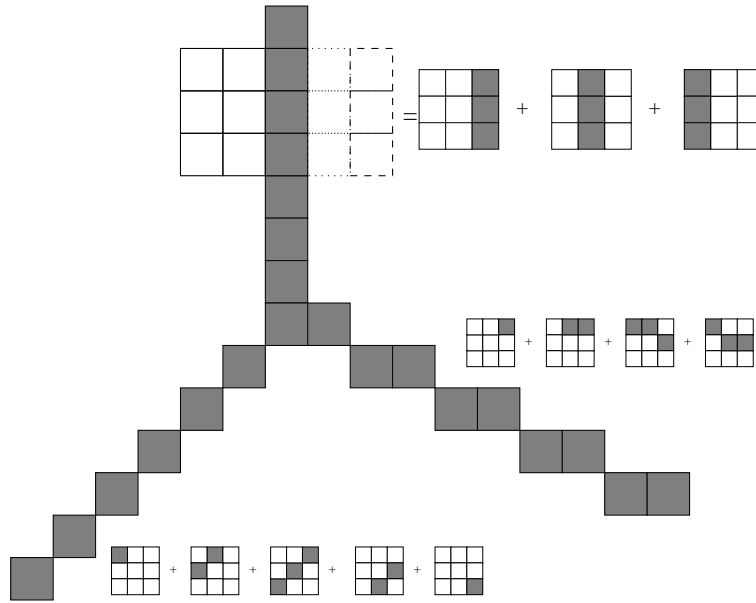


FIG. 2.10 – Définition des lignes dans le chien-modèle

exprimons alors les équations en fonction de ces deux paramètres pour obtenir le système suivant :

$$C(13) = \frac{e}{2} \quad (2.31)$$

$$C(14) = \frac{\sqrt{2}}{2}e - C(3) \quad (2.32)$$

$$C(26) = l - e \quad (2.33)$$

$$C(29) = \frac{\sqrt{5}}{2}e - C(3) - C(5) \quad (2.34)$$

$$2C(11) + C(28) = \sqrt{2}l - 2C(3) \quad (2.35)$$

$$C(35) = \frac{\sqrt{2}e}{2} \quad (2.36)$$

$$C(9) = 2C(3) \quad (2.37)$$

$$C(16) + C(23) = \frac{\sqrt{5}}{2}l - C(3) - C(5) \quad (2.38)$$

$$C(19) = C(3) + C(5) \quad (2.39)$$

Nous résolvons ce système en nous assurant qu'aucune configuration globale n'est d'énergie négative, ce qui est obtenu en considérant une solution pour laquelle aucune configuration locale n'est négative. Les configurations n'apparaissant pas dans le système d'équations sont considérées comme définissant le bruit et ont donc une énergie égale à  $n$  qui définit le troisième paramètre du modèle.

Pour des valeurs strictement positives des paramètres, nous avons uniquement les deux configurations uniformes comme états fondamentaux. En revanche si  $e = l = 0$ , toute configuration composée de bandes parallèles ou d'objets circulaires est un état fondamental.

L'inconvénient de ce modèle par rapport au modèle d'Ising réside essentiellement dans sa complexité. L'utilisation de cliques  $3 \times 3$  entraîne un voisinage de  $5 \times 5$  pixels. Le temps de calcul nécessaire à l'optimisation du modèle est donc accru par rapport à un simple modèle d'Ising 4 ou 8 connexe. En revanche, le modèle proposé est beaucoup plus souple puisqu'il fait la distinction entre bruit, lignes et

contours. Il repr esente donc un *a priori* plus fin sur la solution de la segmentation, ce qui sera d emontr e sur les r esultats exp erimentaux.

**Extension   un mod ele m-aire :** Nous  tendons le chien-mod ele binaire   un mod ele m-aire. De m eme que pour le cas binaire, nous d efinissons un co ut  nerg etique pour les contours, les lignes et le bruit respectivement.

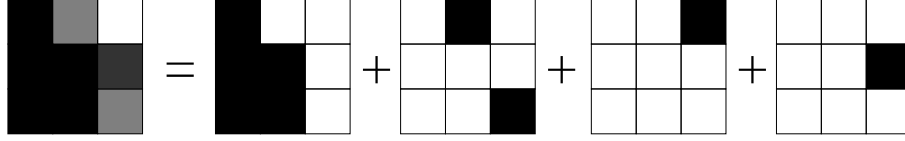


FIG. 2.11 – G en eralisation du mod ele binaire au cas m-aire

Soient  $X$  une configuration sur une clique  $3 \times 3$  et  $\lambda_i, i \in \{1, \dots, n\}$  les diff erents labels pr esents dans la configuration. Consid erons les diff erentes configurations binaires  $X_i$  obtenues en associant l' etat 1   tous les sites ayant pour label  $\lambda_i$  et l' etat 0 aux autres sites. Le potentiel  $V_b(X_i)$  associ e   la configuration  $X_i$  est alors celui correspondant au cas du mod ele binaire. Le potentiel associ e   la configuration  $X$  est alors donn e par la formule (voir figure 2.11) :

$$V(X) = \sum_{i \in \{1, \dots, n\}} V_b(X_i). \quad (2.40)$$

**Comparaison avec le mod ele de Potts :** Pour comparer le chien-mod ele avec le mod ele de Potts, nous allons les r ecrire sous une forme similaire, en utilisant la formulation suivante :

$$P_\Theta(X) = \frac{1}{Z(\Theta)} \exp - \langle \Theta, X \rangle = \frac{1}{Z(\Theta)} \exp \left[ - \sum_i \theta_i N_i(X) \right] \quad (2.41)$$

o  les  $N_i(X)$  sont des fonctions de la configuration  $X$ .

Les potentiels associ es au mod ele de Potts concernent deux sites et ne d ependent que de l' egalit e ou non des labels de ces sites. Nous pouvons donc  crire le mod ele de Potts de la fa on suivante :

$$P_\beta(X) = \frac{1}{Z(\beta)} \exp [-\beta N_0(X)] = \frac{1}{Z(\beta)} \exp [-\beta \#_X], \quad (2.42)$$

o   $N_0(X) = \#_X$  est le nombre de cliques h et erog enes de la configuration  $X$ .

Pour le chien-mod ele, le potentiel associ e   chaque configuration d'une clique est une combinaison lin eaire des trois param etres  $e, l$  et  $n$  :

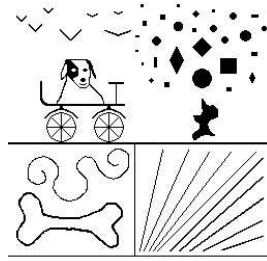
$$\forall i = 0, \dots, 51 \quad C(i) = \epsilon(i)e + \lambda(i)l + \eta(i)n. \quad (2.43)$$

La chien-mod ele peut donc  tre r ecrit comme suit :

$$P_{e,l,n}(X) = \frac{1}{Z(e,l,n)} \exp [-eN_0(X) - lN_1(X) - nN_2(X)] \quad (2.44)$$

o  :

$$\begin{aligned} N_0(X) &= \sum_{i=1, \dots, 51} \epsilon(i) \#_i(X), \\ N_1(X) &= \sum_{i=1, \dots, 51} \lambda(i) \#_i(X), \\ N_2(X) &= \sum_{i=1, \dots, 51} \eta(i) \#_i(X). \end{aligned}$$

FIG. 2.12 – **Image test** : 256 × 256

$\#_i(X)$  étant le nombre de configuration de type  $i$  dans la réalisation  $X$ .

Considérons l'image test de la figure 2.12. Nous estimons, sur l'image test, les paramètres du modèle de Potts et du chien-modèle au sens du maximum de vraisemblance, par une méthode qui sera décrite plus loin. Nous synthétisons ensuite les modèles avec les paramètres estimés. Les résultats sont montrés sur la figure 2.13. Bien entendu, les modèles considérées sont très générales et les simulations effectuées sont loin, visuellement, de l'image originale. Néanmoins, nous pouvons voir les propriétés captées par le modèle et injectées dans la simulation. La simulation du modèle de Potts donne une image représentant un bruit sur un fond uniforme. En effet, la seule propriété captée par le modèle de Potts est le nombre de cliques hétérogènes. Ces cliques peuvent correspondre à du bruit comme à des contours. Pour des raisons d'entropie, la simulation obtenue est composée d'un fond et de composantes connexes de petite taille et est donc peu pertinente du point de vue de la modélisation. En revanche, la simulation du chien-modèle a permis de capter le nombre de contours et de lignes dans l'image originale. Nous obtenons donc une réalisation beaucoup plus proche de l'image originale. Cela montre l'intérêt du chien-modèle qui permet de distinguer le bruit des structures importantes comme les contours et les lignes.

Cette propriété se retrouve lors de l'utilisation de ces modèles comme *a priori* dans des problèmes d'analyse d'image. Considérons le problème de la restauration de l'image binaire bruitée de la figure 2.14. Nous restaurons l'image dans un cadre bayésien avec successivement le modèle de Potts et le chien-modèle comme *a priori*. Les résultats obtenus sont montrés sur la figure 2.15. La restauration obtenue avec le chien-modèle est nettement meilleure. Il est à noter que même des détails aussi fins que les lignes sont bien restaurés. Le modèle de Potts conduit à une image encore bruitée, ce qui est naturel compte tenu de la simulation montrée sur la figure 2.13. En pratique, la restauration par un modèle de Potts se fait avec des interactions plus fortes, c'est notamment le cas de l'estimateur du maximum de pseudo-vraisemblance. Mais ceci induit un phénomène de sur-régularisation qui entraîne une perte des détails et de la précision des contours (voir figure 2.16).

### Propriétés d'isotropie de certains modèles à interactions multi-corps

<sup>4</sup> Le cadre général de ce travail concerne la recherche de critères motivant le choix d'un modèle a priori markovien particulier pour un problème et un type d'image données. Cette première approche s'attache à déterminer les propriétés d'isotropie de certains modèles. Pour ce faire, nous considérons la construction de Wulff de ces modèles à température nulle. Nous considérons les réalisations des modèles (binaires) à température nulle lorsque l'on fixe le rapport des pixels noirs relativement aux pixels blancs (ensemble canonique). Des études similaires ont été faites à basse température sur un nombre réduit de modèles [MS67, MS68, RD92]. Ici, nous nous restreignons à la température nulle mais considérons une classe de modèles beaucoup plus vaste. Nous montrons que, sous certaines hypothèses, la configuration

<sup>4</sup> Nous résumons ici une collaboration menée avec E. Pechevsky et soutenue par l'Institut franco-russe Lyapunov. Bibliographie : [25, 27]

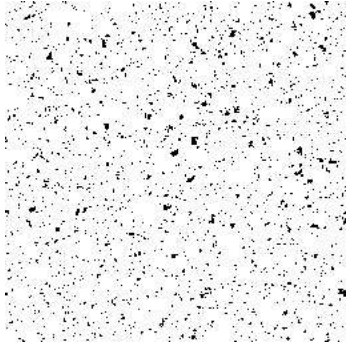
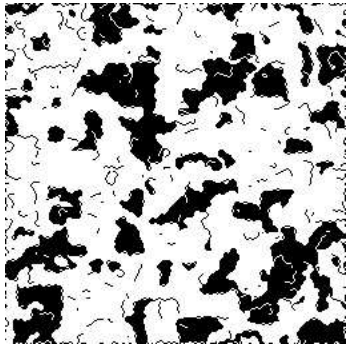
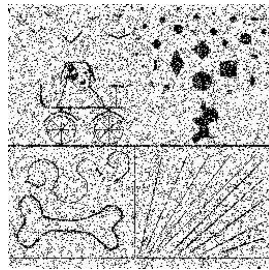
Modèle	$N_0$	Par. es.	$\langle N_0 \rangle$	$N_1$	Par. es.	$\langle N_1 \rangle$
<i>Potts</i>	8706	0.4981	8734			
<i>Chien</i>	3259	0.82	3261	1462	1.5	1464
Modèle	$N_2$	Par. es.	$\langle N_2 \rangle$			
<i>Potts</i>						
<i>Chien</i>	1101	1.6	1094			
Modèle	Echantillon correspondant aux paramètres estimés			Commentaires		
<i>Potts</i>				Les cliques hétérogènes sont représentées. La simulation est éloignée de l'image originale. Nous obtenons du bruit sur un fond uniforme.		
<i>Chien</i>				Les contours et les lignes sont représentés.		

FIG. 2.13 – Comparaison du modèle de Potts et du chien-modèle comme modèles a priori

FIG. 2.14 – Image test bruitée par un bruit de canal ( $\tau = 0.15$ )

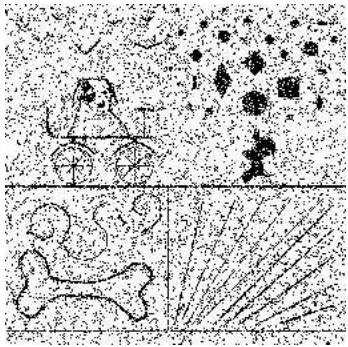
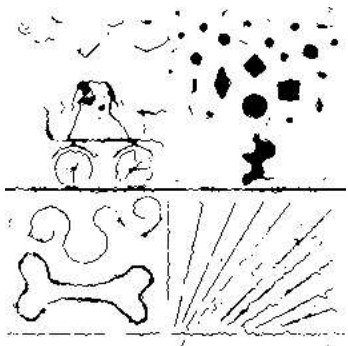
Modèle	Image restaurée	Commentaires
<i>Potts</i>		Le résultat est encore bruité. Nous ne pouvons obtenir une solution à la fois régularisée et ayant le bon nombre de cliques hétérogènes. Pour régulariser le résultat il faut accroître la valeur de $\beta$ .
<i>Chien</i>		Le chien-modèle est plus adapté à la restauration car il permet de contrôler la longueur des contours et des lignes. L'a priori modélise réellement certaines caractéristiques de l'image traitée.

FIG. 2.15 – Restauration d'une image binaire bruitée par le modèle de Potts et par le chien-modèle

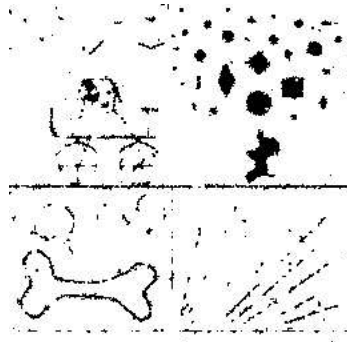


FIG. 2.16 – Restauration de la figure 2.14, sur-régularisation obtenue avec un modèle de Potts ( $\beta = 1.0$ )

obtenue est formée d'une goutte polygonale de la phase la moins représentée plongée dans un océan de la phase la plus représentée.

Pour écrire l'isotropie des modèles, nous proposons d'étudier la forme de la goutte. Les plus ou moins grandes déviations de cette forme par rapport au disque peuvent être interprétées comme un degré d'isotropie.

Nous considérons une classe de modèles ferromagnétiques binaires sur  $\mathbb{Z}$  ayant une longueur d'interaction inférieure ou égale à  $\sqrt{2}$ . Nous donnons une classification de ces modèles relativement à la forme de la goutte. Nous nous restreignons à une sous-classe définie par des conditions de régularité. Le résultat principal décrit un ensemble de polygones représentant les formes possibles de la goutte à la limite thermodynamique pour une température nulle. En particulier, cet ensemble contient le carré, résultat connu sur le modèle d'Ising et un polygone à seize côtés obtenu pour le chien-modèle. Nous appelons les formes obtenues *formes à l'échelle macroscopique*. La condition de régularité évite les cas où la forme obtenue est non-convexe ou même non connexe pour certains exemples particuliers.

Nous étudions les modèles sur  $\mathbb{Z}^2$  avec pour espace des spins  $X = \{0,1\}$ . Les seules fonctions potentielles pouvant être différentes de zéros sont :

$$\Phi_{W_t} : X^{W_t} \rightarrow \mathbb{R}, \quad (2.45)$$

où

$$W_0 = \{t = (t_1, t_2) \in \mathbb{Z}^2 : |t_i| \leq 1, i = 1, 2\}, \quad (2.46)$$

et  $W_t = W_0 + t$ . Un sous-ensemble  $W_t$  est appelé une *plaquette*. Nous considérons des modèles invariants par translation. Ainsi,  $\Phi_{W_0}(\mathbf{x}) = \Phi_{W_t}(T_t \mathbf{x}) = \Phi(\mathbf{x})$ , où  $T_t \mathbf{x}(u) = \mathbf{x}(t - u)$ , pour toute configuration  $\mathbf{x} : \mathbb{Z}^2 \rightarrow X$ , et tout  $t \in \mathbb{Z}^2$ .

La distribution de Gibbs est notée  $\Phi$ . Soit  $\Theta \subset \mathbb{Z}^2$  un volume fini et  $\mathcal{P}_\Theta$  l'ensemble de toutes les plaquettes dans  $\Theta$ . L'énergie d'une configuration  $\mathbf{x} : \Theta \rightarrow X$  est alors :

$$H(\mathbf{x}) = \sum_{W \in \mathcal{P}_\Theta} \Phi(\mathbf{x}_W). \quad (2.47)$$

La probabilité de Gibbs de  $\mathbf{x}$  dans le volume  $\Theta$  est :

$$P_{\Theta, \beta}(\mathbf{x}) = \frac{\exp\{-\beta H(\mathbf{x})\}}{Z_{\Theta, \beta}}, \quad (2.48)$$

où  $Z_{\Theta, \beta} = \sum_{\mathbf{y} \in X^\Theta} \exp\{-\beta H(\mathbf{y})\}$ . La distribution de Gibbs dans  $\mathbb{Z}^2$  est donnée par la limite thermodynamique de la spécification de Gibbs.

Nous donnons maintenant les hypothèses sur  $\Phi$ . Une *tuile* est un tableau de neuf chiffres :

$$\bar{r} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}, \quad (2.49)$$

où  $r_{ij} \in \{0,1\}$ . Soient  $\bar{v}_0 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$  et  $\bar{v}_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ . L'énergie étant définie à une constante additive près, nous posons :

$$\Phi(\bar{v}_0) = \Phi(\bar{v}_1) = 0. \quad (2.50)$$

En toute généralité, il y a neuf tuiles différentes et le même nombre de valeurs possibles pour  $\Phi$ . Néanmoins, nous imposons à  $\Phi$  d'être invariant par rapport aux symétries naturelles des tuiles, c'est à dire par rapport aux rotations de  $\bar{r}$  par  $\frac{\pi}{2}$  et aux symétries relativement aux axes horizontal et vertical. Ces transformations forment un groupe  $\hat{G}$ . Nous ajoutons à  $\hat{G}$  l'inversion de  $\bar{r}$  transformant chaque  $r_{ij}$  en  $1 + r_{ij} \pmod{2}$ . Soit  $G$  le groupe ainsi obtenu.

Nous supposons les conditions suivantes sur la fonction  $\Phi$  satisfaites :

**$\Phi 1$**  Pour tout  $g \in G$  et tout  $\bar{r}$

$$\Phi(\bar{r}) = \Phi(g(\bar{r})).$$

Cette condition r eduit l'ensemble des   valeurs possibles de  $\Phi$    51.

La condition suivante assure que les mod es soient de type ferromagn etiques. Soit  $k_1, k_2 \in \mathbb{Z}_+$  tels que  $k_i \geq 3, i = 1, 2$ . Consid erons un tableau  $\hat{s} = (s_{ij})_{\substack{1 \leq i \leq k_1, \\ 1 \leq j \leq k_2}}$ , o   $s_{ij} \in \{0, 1\}$ . Soit  $\mathcal{T}_{\hat{s}} = \{\bar{r}\}$  l'ensemble des tuiles  $\bar{r}$  qui peuvent  tre extraites de  $\hat{s}$ . Nous supposons que :

**$\Phi 2$**  Si  $\hat{s}$  n'est pas constant (0 ou 1) alors :

$$\sum_{\bar{r} \in \mathcal{T}_{\hat{s}}} \Phi(\bar{r}) > 0.$$

Il vient, du fait de (2.50) et  **$\Phi 2$** , que toute perturbation locale de la configuration  $\mathbf{x}_0(t) \equiv 0$  (ou  $\mathbf{x}_0(t) \equiv 1$ ),  $t \in \mathbb{Z}^2$ , a une  nergie finie et positive. Ainsi, les configurations  $\mathbf{x}_0(t) \equiv 0$  et  $\mathbf{x}_1(t) \equiv 1$  sont les uniques  tats fondamentaux p eriodiques. Il est  galement ais e de v erifier que la condition de Peierls est satisfaite. Il existe donc une temp erature critique s eparant le cas d'un  tat de Gibbs unique du cas d'au moins deux  tats de Gibbs.

Notre but est de caract eriser les  tats fondamentaux des mod es satisfaisant aux conditions  **$\Phi 1$**  et  **$\Phi 2$**  pour l'ensemble canonique. La preuve que nous donnons est restreinte aux mod es que nous appelons *r eguliers*. N anmoins, nous conjecturons qu'une classification similaire est valide pour tous les mod es v erifiant les deux conditions cit ees.

Consid erons les tuiles suivantes :

$$\begin{aligned} \bar{u}_0 &= \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \\ \bar{u}_1^1 &= \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \bar{u}_1^2 &= \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \\ \bar{u}_2^1 &= \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \bar{u}_2^2 &= \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}. \end{aligned}$$

Nous utilisons les notations suivantes :

$$\begin{aligned} \tilde{\mathcal{U}} &= \{\bar{u}_0, \bar{u}_1^1, \bar{u}_1^2, \bar{u}_2^1, \bar{u}_2^2\}. \\ \mathcal{U} &= G\tilde{\mathcal{U}}. \end{aligned} \tag{2.51}$$

Le  $p$ -contour (contour contenant uniquement des tuiles de  $\mathcal{U}$ )  $\Omega_{\mathbf{x}}$  est *r egulier* s'il est compos e de tuiles appartenant    $\mathcal{U}$ . Soit  $\mathfrak{M}$  la classe des mod es satisfaisant aux conditions  **$\Phi 1$**  et  **$\Phi 2$** .

Un mod e de  $\mathfrak{M}$  est dit *r egulier* si pour toute configuration  $\mathbf{x}$  nous avons la propri et e suivante :

si  $t, s \in \Lambda_n$  appartiennent   une composante connexe de  $\Omega_{\mathbf{x}}$  alors il existe une configuration  $\mathbf{y}$  telle qu'une composante connexe de  $\Omega_{\mathbf{y}}$  contienne  $s$  et  $t$ , toutes les tuiles de cette composante appartiennent    $\mathcal{U}$  et  $H(\mathbf{y}) \leq H(\mathbf{x})$ .

Soit  $\mathfrak{M}_r$  la classe des mod es r eguliers.

Nous donnons une classification de la forme macroscopique des gouttes pour les modèles de cette classe. Pour formuler le théorème correspondant, nous introduisons les notations suivantes :

$$\begin{aligned} E_0 &= 2\Phi \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \\ E_{1/2} &= \Phi \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} + \Phi \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} + \Phi \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \\ E_1 &= 2 \left( \Phi \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} + \Phi \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \right), \end{aligned} \quad (2.52)$$

et

$$e_{1/2} = \frac{E_{1/2}}{E_0}, \quad e_1 = \frac{E_1}{E_0}. \quad (2.53)$$

Il vient d'après **Φ2** que  $E_0 > 0$ ,  $E_{1/2} > 0$  et  $E_1 > 0$ . Ceci est aisément prouvé en considérant les configurations  $\mathbf{y}_0, \mathbf{y}_{1/2}$  et  $\mathbf{y}_1$  sur un large volume rectangulaire comme suit. Pour un grand  $n$  considérons les fonctions  $g_0^i, g_{1/2}^i, g_1^i$ ,  $i = 1, 2$ , définies sur  $[-1, 1]$ :

$$\frac{dg_\varepsilon^i}{dt} = -\varepsilon, \quad g_\varepsilon^1(0) = \frac{m}{n}, \quad g_\varepsilon^2(0) = -\frac{m}{n}, \quad (2.54)$$

où  $\varepsilon \in \{0, \frac{1}{2}, 1\}$  et  $m$  est un entier fixé et supérieur à 5. Considérons les configurations  $\mathbf{y}_\varepsilon$  sur  $\frac{1}{n}\mathbb{Z}^2$  définies par :

$$\mathbf{y}_\varepsilon(t) = \begin{cases} 0, & \text{if } t = (t_1, t_2) \in \Lambda_n \text{ and } g_\varepsilon^1(t_1) \leq t_2 \leq g_\varepsilon^2(t_1), \\ 1, & \text{otherwise.} \end{cases} \quad (2.55)$$

Du fait de **Φ2** chaque  $\mathbf{y}_\varepsilon$  a une énergie positive. Il est aisée de vérifier que si  $n$  est suffisamment grand alors  $4nE_\varepsilon$  donne la principale contribution à l'énergie de  $\mathbf{y}$  et par suite  $E_\varepsilon > 0$ .

L'indice de  $E_\varepsilon$  et  $e_\varepsilon$  signifie que la dérivée des fonctions de l'équation (2.54) vaut  $-\varepsilon$  et que la contribution principale de l'énergie des configurations correspondantes définies par l'équation (2.55) est donnée par  $E_\varepsilon$ .

Nous partitionnons l'espace  $\mathfrak{M}_r$  par les régions suivantes définies par les valeurs de  $e_{1/2}, e_1$  :

$$\begin{aligned} A_{17} &= \left\{ (e_{1/2}, e_1) : e_{1/2} \geq \frac{3}{2}, e_1 \geq 2 \right\}, \\ A_{15} &= \left\{ (e_{1/2}, e_1) : e_1 \leq 2e_{1/2} - 1, 1 \leq e_1 \leq 2 \right\}, \\ A_{45} &= \left\{ (e_{1/2}, e_1) : e_1 \leq e_{1/2}, e_1 \leq 1 \right\}, \\ A_{13} &= \left\{ (e_{1/2}, e_1) : e_1 \geq 2e_{1/2} - 1, 1 \leq e_{1/2} \leq \frac{3}{2} \right\}, \\ A_{23} &= \left\{ (e_{1/2}, e_1) : e_1 \geq e_{1/2}, e_{1/2} \leq 1 \right\}, \\ A_{35} &= \left\{ (e_{1/2}, e_1) : e_1 \leq \frac{4}{3}e_{1/2}, e_1 \geq e_{1/2}, e_1 \geq 2e_{1/2} - 1 \right\}, \\ A_{36} &= \left\{ (e_{1/2}, e_1) : e_1 \geq \frac{4}{3}e_{1/2}, e_{1/2} \leq \frac{3}{2} \right\}. \end{aligned} \quad (2.56)$$

Chaque région  $A_j$  contient tous les modèles de  $\mathfrak{M}_r$  ayant des valeurs de  $(e_{1/2}, e_1)$  dans le domaine correspondant.

Nous écrivons la forme macroscopique des gouttes en déterminant leur contour dans le domaine  $\{(\lambda_1, \lambda_2) : \lambda_2 \geq \lambda_1 \geq 0\}$ . Ces contours sont définis comme une fonction  $d$  dont le graphe est compris entre l'axe  $\lambda_1 = 0$  et la diagonale  $\lambda_1 = \lambda_2$ . Nous considérons les gouttes d'aire unitée. Dès lors, la portion



de la goutte dans le domaine étudié a une aire de  $\frac{1}{8}$ . Les coefficients  $\delta$  du théorème suivant sont choisis de sorte que l'aire de la goutte soit égale à 1. Ils peuvent être aisément calculés mais pour ne pas surcharger les formules, nous omettons leur forme explicite. Il est clair que les valeurs de  $\delta$  sont différentes suivant les domaines, nous ne le rappelons pas par des indices pour alléger les notations.

**Théorème 1.** *Pour les modèles de  $\mathfrak{M}_r$ , les formes macroscopiques des gouttes sont données par :*

1. dans  $A_{17}$

$$d(\lambda_1) = 2\delta, \text{ si } \lambda_1 \in [0, 2\delta] \quad (2.57)$$

2. dans  $A_{45}$

$$d(\lambda_1) = -\lambda_1 + 2\delta, \text{ si } \lambda_1 \in [0, \delta e_1]. \quad (2.58)$$

3. dans  $A_{15}$

$$d(\lambda_1) = \begin{cases} 2\delta, & \text{si } \lambda_1 \in [0, 2\delta(e_1 - 1)], \\ -\lambda_1 + 2\delta e_1, & \text{si } \lambda_1 \in [2\delta(e_1 - 1), \delta e_1], \end{cases} \quad (2.59)$$

4. dans  $A_{13} \cap A_{35}$

$$d(\lambda_1) = \begin{cases} 2\delta, & \text{si } \lambda_1 \in [0, 4\delta(e_{1/2} - 1)], \\ -\frac{1}{2}\lambda_1 + 2\delta e_{1/2}, & \text{si } \lambda_1 \in [4\delta(e_{1/2} - 1), 4\delta(e_1 - e_{1/2})], \\ -\lambda_1 + 2\delta e_1, & \text{si } \lambda_1 \in [4\delta(e_1 - e_{1/2}), \delta e_1]. \end{cases} \quad (2.60)$$

5. dans  $A_{13} \cap A_{36}$

$$d(\lambda_1) = \begin{cases} 2\delta, & \text{si } \lambda_1 \in [0, 4\delta(e_{1/2} - 1)], \\ -\frac{1}{3}\lambda_1 + 2\delta e_{1/2}, & \text{si } \lambda_1 \in [4\delta(e_{1/2} - 1), \frac{4}{3}\delta e_{1/2}], \end{cases} \quad (2.61)$$

6. dans  $A_{23} \cap A_{36}$

$$d(\lambda_1) = -\frac{1}{2}\lambda_1 + 2\delta, \text{ si } \lambda_1 \in [0, \frac{4}{3}\delta], \quad (2.62)$$

7. dans  $A_{23} \cap A_{35}$

$$d(\lambda_1) = \begin{cases} -\frac{1}{2}\lambda_1 + 2\delta e_{1/2}, & \text{si } \lambda_1 \in [0, 4\delta(e_1 - e_{1/2})], \\ -\lambda_1 + 2\delta e_1, & \text{si } \lambda_1 \in [4\delta(e_1 - e_{1/2}), \delta e_1], \end{cases} \quad (2.63)$$

Ce résultat est illustré par la figure 2.2.2, où les formes macroscopiques sont données pour les régions  $A_{ij}$ .

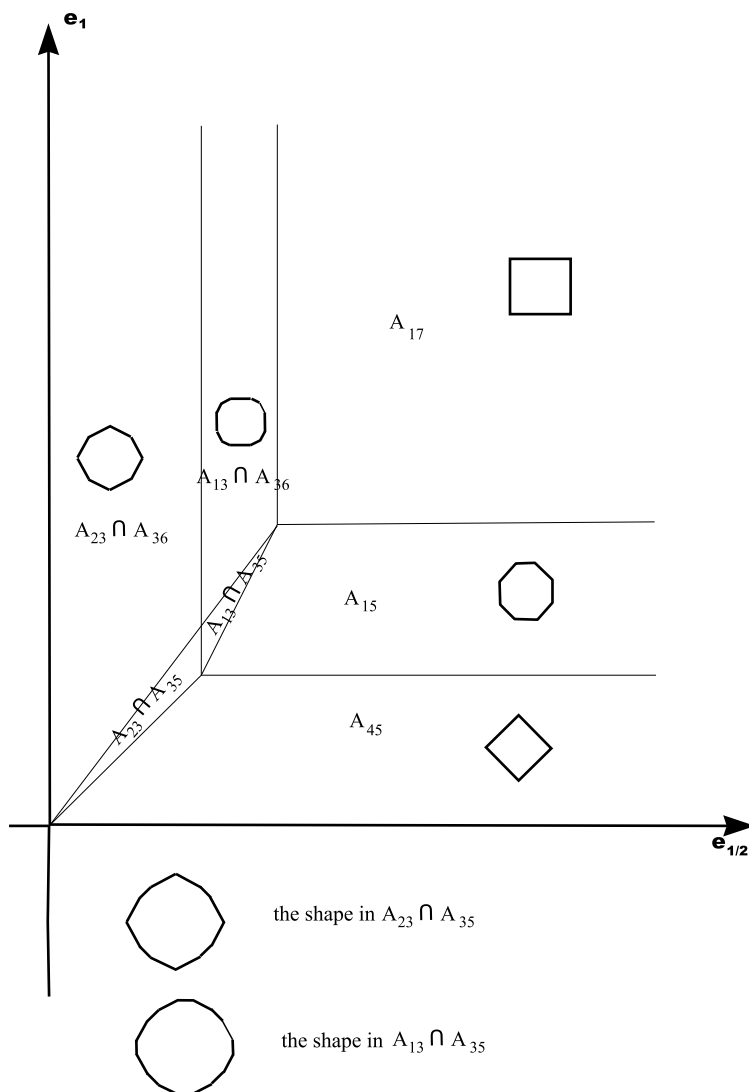


FIG. 2.17 – Formes obtenues pour les différents modèles

Les indices dans les notations des régions proviennent du problème de programmation linéaire que nous avons résolu pour obtenir les formes. Dans ce problème interviennent sept coefficients et deux relations entre eux, chaque coefficient correspondant à la pente d'une droite définissant la forme. Les deux entiers en indice correspondent aux pentes à partir desquelles les contours de la forme correspondante sont construits.

Notons que dans  $A_{17}$ ,  $A_{45}$  et  $A_{23} \cap A_{36}$ , la forme des polygones ne dépend pas de l'énergie des contours. Nous appelons ces polygones des polygones *purs*. Dans les autres régions, la forme du polygone dépend de l'énergie. Ils peuvent alors être considérés comme un mélange de polygones purs.

Une forme familière de goutte est obtenue dans  $A_{17}$ . Le modèle d'Ising, dont les coordonnées dans le plan  $(e_{1/2}, e_1)$  sont  $(\frac{3}{2}, 2)$  appartient à cette région. La forme la "plus isotrope" est un polygone à seize cotés. Le chien-modèle dont les coordonnées dans le plan  $(q_2, e_1)$  sont  $(\frac{\sqrt{5}}{2}, \sqrt{2})$  appartient à la région correspondante à ce polygone.

Dans cette étude, nous avons considéré certains modèles que nous avons définis comme réguliers. Des exemples de modèles non réguliers ainsi que deux conditions suffisantes de régularité sont données dans [27].

### 2.2.3 Estimation des paramètres : un algorithme de type MVMCMC

<sup>5</sup> Nous écrivons ici un algorithme, fondé sur une approche MCMC, permettant d'obtenir des estimateurs des paramètres d'un champ markovien au sens du Maximum de Vraisemblance (MV). L'énergie du champ markovien est supposée dépendre linéairement des paramètres. Cette approche est dérivée des travaux de Geyer et Thompson [GT92]

Soit  $P_\Theta$  un champ aléatoire défini sur  $S$  et paramétré par le vecteur  $\Theta \in \mathcal{B}$  (Nous supposons que  $P_\Theta$  est un champ de Gibbs dont l'énergie est linéaire par rapport à  $\theta$ ). Il s'écrit alors :

$$P_\Theta(Y) = \frac{1}{Z(\Theta)} \exp - \langle \Theta, Y \rangle = \frac{1}{Z(\Theta)} \exp \left[ - \sum_i \theta_i N_i(Y) \right] \quad (2.64)$$

où les  $N_i(Y)$  sont des fonctions de la configuration  $Y$ . Pour l'écriture, nous considérons un espace d'état continu  $\Lambda$ . Les résultats sont identiques en discret en changeant les intégrales en somme. La fonction de partition  $Z(\Theta)$  s'écrit alors :

$$Z(\Theta) = \int_{\Lambda^S} \exp \left[ - \sum_i \theta_i N_i(X) \right] dX, \quad (2.65)$$

Nous considérons que nous connaissons les données  $Y$ . Nous cherchons le modèle qui correspond le mieux aux données. La log-vraisemblance est définie par :

$$\log P(Y|\Theta) = \log \left[ \frac{1}{Z(\Theta)} \exp - \sum_i \theta_i N_i(Y) \right], \quad (2.66)$$

$$\log P(Y|\Theta) = - \sum_i \theta_i N_i(Y) - \log Z(\Theta). \quad (2.67)$$

Les estimateurs au sens du MV sont obtenus en maximisant la log-vraisemblance. Nous avons alors :

$$\forall i, \frac{\partial \log P(Y|\Theta)}{\partial \theta_i} (\hat{\Theta}) = 0, \quad (2.68)$$

et donc :

$$\forall i, -N_i(Y) + \frac{\int_{\Lambda^S} N_i(X) \exp \left[ - \sum_i \hat{\theta}_i N_i(X) \right] dX}{\int_{\Lambda^S} \exp \left[ - \sum_i \hat{\theta}_i N_i(X) \right] dX} = 0, \quad (2.69)$$

où  $\hat{\theta}_i$  est l'estimateur au sens du MV de  $\theta_i$ .

Notons par  $\langle a(x) \rangle_\Theta$ , l'espérance de  $a(x)$  relativement à  $\mathcal{B}$ . nous avons donc :

$$\forall i, \langle N_i(X) \rangle_{\hat{\Theta}} = N_i(Y). \quad (2.70)$$

Pour évaluer la log-vraisemblance, nous devons calculer la fonction de partition. Les dérivées partielles de la log-vraisemblance nécessitent le calcul des différents moments  $\langle N(X) \rangle_\Theta$ . Malheureusement, le calcul de ces quantités n'est pas envisageable analytiquement. Nous pouvons cependant estimer les quantités  $\langle N(X) \rangle_\Theta$  en échantillonnant la distribution. Néanmoins, échantillonner la distribution pour chaque valeur de  $\Theta$  est inconcevable du point de vue du temps de calcul. Nous introduisons donc l'échantillonnage préférentiel

5. Nous résumons ici des travaux faits en collaboration avec Robin Morris, Josiane Zerubia et Marc Berthod. Bibliographie : [23, 24]

6. *Importance sampling* en anglais

**Echantillonnage préférentiel-** L'échantillonnage préférentiel permet d'estimer les moments statistiques correspondant à  $P_\Theta$  à partir d'échantillons de la loi  $R_\Theta$ .

Considérons, en premier lieu, la fonction de partition :

$$Z(\Theta) = \int_{\Omega} \exp \left[ - \sum_i \theta_i N_i(X) \right] dX, \quad (2.71)$$

alors :

$$\begin{aligned} Z(\Theta) &= \int_{\Omega} \exp \left[ - \sum_i (\theta_i - \psi_i) N_i(X) \right] \exp \left[ - \sum_i \psi_i N_i(X) \right] dX \\ Z(\Theta) &= \int_{\Omega} \exp \left[ - \sum_i (\theta_i - \psi_i) N_i(X) \right] Z(\Psi) dP_{\Psi}(X). \end{aligned} \quad (2.72)$$

Pour tout couple  $(\Theta, \Psi)$ , le rapport des fonctions de partition est donc donné par :

$$\frac{Z(\Theta)}{Z(\Psi)} = E_{\Psi} \left( \exp \left[ - \sum_i (\theta_i - \psi_i) N_i(X) \right] \right), \quad (2.73)$$

où  $E_{\Psi}$  représente l'espérance par rapport à la loi  $\Psi$ .

La fonction de partition correspondant à  $P_\Theta$  peut donc être estimée à partir d'un échantillonnage de  $P_{\Psi}$ . Nous devons uniquement échantillonner la loi de paramètre  $\Psi$  pour obtenir un estimateur du rapport  $\frac{Z(\Theta)}{Z(\Psi)}$  pour tout  $\Theta$  en calculant l'espérance donnée par la formule (2.73).

Considérons maintenant la log-vraisemblance. L'estimateur au sens du MV est donné par le vecteur  $\Theta$  qui maximise la formule (2.66), ce qui est équivalent à minimiser l'expression suivante :

$$- \log P_{\Theta}(Y) = \sum_i \theta_i N_i(Y) + \log \frac{Z(\Theta)}{Z(\Psi)}. \quad (2.74)$$

Les dérivées partielles de la fonction de partition s'écrivent :

$$\begin{aligned} \frac{\partial Z(\Theta)}{\partial \theta_i} &= \int_{\Omega} -N_i(X) \exp \left[ - \sum_j (\theta_j - \psi_j) N_j(X) \right] Z(\Psi) dP_{\Psi}(X) \\ &= -Z(\Psi) E_{\Psi} \left( N_i(X) \exp \left[ - \sum_j (\theta_j - \psi_j) N_j(X) \right] \right). \end{aligned} \quad (2.75)$$

Nous avons donc :

$$\frac{\partial - \log P_{\Theta}(Y)}{\partial \theta_i} = N_i(Y) - \frac{E_{\Psi} \left( N_i(X) \exp \left[ - \sum_j (\theta_j - \psi_j) N_j(X) \right] \right)}{E_{\Psi} \left( \exp \left[ - \sum_j (\theta_j - \psi_j) N_j(X) \right] \right)}. \quad (2.76)$$

A partir d'un échantillonnage de  $R_{\Psi}$ , nous pouvons donc théoriquement estimer la log-vraisemblance de  $P_\Theta$  et de ses dérivées partielles pour tout  $\Theta$ . Le même type de calcul permet de calculer le hessien et nous avons :

$$\begin{aligned} \frac{\partial^2 - \log P_{\Theta}(Y)}{\partial \theta_i \partial \theta_j} &= \frac{E_{\Psi} (N_i(X) N_j(X) \exp - [\sum_k (\theta_k - \psi_k) N_k(X)])}{E_{\Psi} (\exp - [\sum_k (\theta_k - \psi_k) N_k(X)])} \\ &\quad - \frac{E_{\Psi} (N_i(X) \exp - [\sum_k (\theta_k - \psi_k) N_k(X)]) E_{\Psi} (N_j(X) \exp - [\sum_k (\theta_k - \psi_k) N_k(X)])}{E_{\Psi} (\exp - [\sum_k (\theta_k - \psi_k) N_k(X)])^2}. \end{aligned} \quad (2.77)$$

Considérons une image  $Y$  à partir de laquelle nous voulons obtenir l'estimateur au sens du MV de  $\Theta$  en considérant  $\mathcal{B}$  comme modèle. A partir de l'image, nous pouvons estimer la valeur des différentes quantités  $N(Y)$ . Ensuite, nous pouvons échantillonner la loi  $\mathcal{R}_\Psi$  pour une valeur de  $\Psi$  donnée. A partir des échantillons obtenus, les différentes espérances impliquées dans les formules (2.73) et (2.76) peuvent être estimées. Alors, pour tout  $\Theta$ , nous pouvons estimer la log-vraisemblance et ses dérivées. Un algorithme d'optimisation, de type descente de gradient ou gradient conjugué, permet alors d'obtenir une estimation de  $\Theta$  au sens du MV, lorsque la log-vraisemblance est convexe.

Néanmoins, si deux jeux de paramètres  $\Theta$  et  $\Psi$  sont trop éloignés, l'estimation des espérances sera peu précise. En effet, la robustesse de cette estimation nécessite que le recouvrement entre les deux distributions  $P_\Theta$  et  $P_\Psi$  soit suffisamment important. En pratique, la méthode décrite n'est valide que dans un voisinage du paramètre  $\Psi$ . Durant l'optimisation, lorsque la valeur courante de  $\Theta$  s'éloigne trop de  $\Psi$ , nous devons ré-échantillonner le modèle en utilisant la valeur courante des paramètres.

**Algorithme d'estimation-** Considérons l'estimation courante des paramètres  $\hat{\Theta}$  et un échantillonnage de  $P_{\hat{\Theta}}$ . Nous pouvons estimer le gradient et le hessien de la log-vraisemblance en  $\hat{\Theta}$ . Nous calculons ensuite les directions conjuguées. Suivant chaque direction conjuguée, nous définissons un intervalle sur lequel l'estimation est robuste. Nous maximisons alors la log-vraisemblance sur chacun de ces intervalles. Le processus est alors itéré en générant de nouveaux échantillons (avec la valeur courante de  $\hat{\Theta}$ ) jusqu'à convergence.

L'algorithme s'écrit alors :

1. Calculer les statistiques de l'image  $N_i(Y)$ .
2. Initialiser l'estimation  $\hat{\Theta}_0, n = 0$ .
3. Échantillonner la distribution avec la valeur courante des paramètres  $P_{\hat{\Theta}_n}$ .
4. Estimer le gradient et le hessien de la log-vraisemblance en  $\hat{\Theta}_n$ , avec les équations (2.76) et (2.78).
5. Calculer les directions conjuguées  $\Delta$ .
6. Pour chaque direction conjuguée, définir un intervalle de recherche pour lequel l'estimation est supposée robuste.
7. Calculer  $\hat{\Theta}_{n+1}$  en maximisant la log-vraisemblance sur chaque intervalle par le principe de la section d'or.
8. Si  $\|\hat{\Theta}_{n+1} - \hat{\Theta}_n\| > \epsilon$  alors  $n = n + 1$  et retourner en 3, où  $\epsilon$  est un seuil choisi suivant la précision désirée.

**Validation-** Nous considérons différents modèles markoviens pour valider l'approche proposée. Le modèle de Potts peut être reformulé sous la forme de l'équation (2.64):

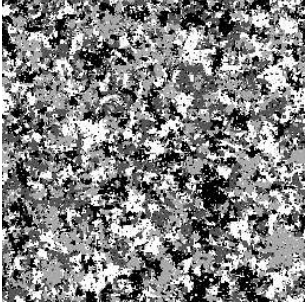
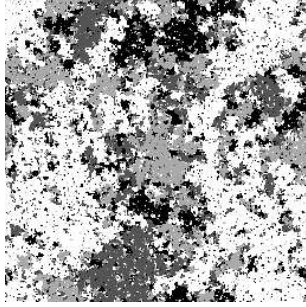
$$P_\beta(X) = \frac{1}{Z(\beta)} \exp[-\beta N_0(X)] = \frac{1}{Z(\beta)} \exp[-\beta \#X], \quad (2.78)$$

où  $N_0(X) = \#X$  est le nombre de cliques non homogènes de la configuration  $X$ . Le modèle ne dépend que d'un paramètre, ce qui simplifie l'algorithme puisque nous n'avons plus à calculer les directions conjuguées. Le tableau 2.3 montre les estimées obtenues à partir de réalisations correspondant à différentes valeurs de  $\beta$ .

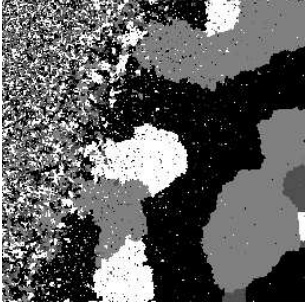
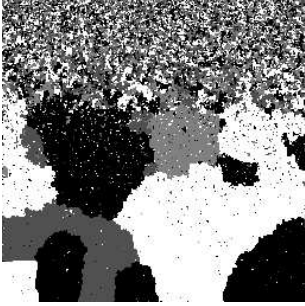
Nous pouvons étendre l'approche au cas d'un modèle de Potts non-stationnaire. Soit un modèle de Potts dont le paramètre d'interaction  $\beta$  dépend de la localisation de la clique. Nous supposons pour simplifier que cette dépendance est linéaire :

$$\beta_{c=\{x_{i,j}, x_{p,q}\}} = a \left( \frac{i+p}{2} \right) + b \left( \frac{j+q}{2} \right) + c. \quad (2.79)$$

Pour estimer  $\beta$ , il nous faut donc estimer  $a$ ,  $b$  et  $c$ .

Paramètres	$\beta = 0.53$ ( $N_0 = 48070$ )	$\beta = 0.5493$ ( $N_0 = 35699$ )
Estimées	$\hat{\beta} = 0.529$ ( $\langle N_0 \rangle = 48072$ )	$\hat{\beta} = 0.5488$ ( $\langle N_0 \rangle = 35699$ )
Echantillon		

TAB. 2.3 – Estimation du paramètre du modèle de Potts

Paramètres	$c$	0.3	$N_0$	26603	$c$	0.5	$N_0$	29171
	$b$	0.005	$N_1$	1703201	$b$	0.005	$N_1$	1816390
	$a$	0.002	$N_2$	2777495	$a$	0.0	$N_2$	3704620
Estimées	$\hat{c}$	0.3002	$\langle N_0 \rangle$	27960	$\hat{c}$	0.535	$\langle N_0 \rangle$	30495
	$\hat{b}$	0.0052	$\langle N_1 \rangle$	1771015	$\hat{b}$	0.0052	$\langle N_1 \rangle$	1859293
	$\hat{a}$	0.0025	$\langle N_2 \rangle$	2757615	$\hat{a}$	0.0003	$\langle N_2 \rangle$	3760871
Echantillon								

TAB. 2.4 – Estimation des paramètres d’une version non-stationnaire du modèle de Potts

Ce modèle peut s’écrire sous la forme de l’équation (2.64) :

$$P_{a,b,c}(X) = \frac{1}{Z(a,b,c)} \exp [-cN_0(X) - bN_1(X) - aN_2(X)] \quad (2.80)$$


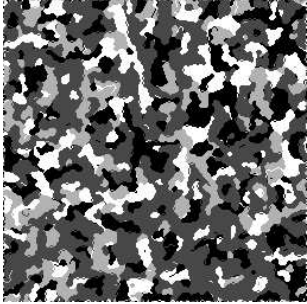
avec :

$$N_0(X) = \#_X, \text{ le nombre de cliques h'et'rogènes}$$

$$N_1(X) = \sum_{\text{cliques h'et'rogènes}} \frac{j+q}{2} = N_0(X) \left\langle \frac{j+q}{2} \right\rangle_{\text{cl. h'et.}}$$

$$N_2(X) = \sum_{\text{cliques h'et'rogènes}} \frac{i+p}{2} = N_0(X) \left\langle \frac{i+p}{2} \right\rangle_{\text{cl. h'et.}}$$

Le tableau 2.4 montre des échantillons de ce modèle et les paramètres estimés sur ces échantillons.

Paramètres	$e$	0.2	$N_0$	32950	$e$	0.4	$N_0$	21089
	$l$	0.4	$N_1$	11724	$l$	0.8	$N_1$	1710
	$n$	0.6	$N_2$	49708	$n$	1.0	$N_2$	2880
Estimées	$\hat{e}$	0.2008	$\langle N_0 \rangle$	32857	$\hat{e}$	0.3905	$\langle N_0 \rangle$	21358
	$\hat{l}$	0.4038	$\langle N_1 \rangle$	11669	$\hat{l}$	0.7843	$\langle N_1 \rangle$	1585
	$\hat{n}$	0.5997	$\langle N_2 \rangle$	49606	$\hat{n}$	0.9993	$\langle N_2 \rangle$	2947
Echantillon								

TAB. 2.5 – Estimation des paramètres du chien-modèle

Considérons pour finir le chien-modèle. Le potentiel associé à chaque configuration sur une clique est une combinaison linéaire des trois paramètres  $e$ ,  $l$  et  $n$ :

$$\forall i = 0, \dots, 51 \quad C(i) = \epsilon(i)e + \lambda(i)l + \eta(i)n. \quad (2.81)$$

La distribution résultante s'écrit sous la forme :

$$P_{e,l,n}(X) = \frac{1}{Z(e,l,n)} \exp[-eN_0(X) - lN_1(X) - nN_2(X)] \quad (2.82)$$

où :

$$\begin{aligned} N_0(X) &= \sum_{i=1, \dots, 51} \epsilon(i) \#_i(X), \\ N_1(X) &= \sum_{i=1, \dots, 51} \lambda(i) \#_i(X), \\ N_2(X) &= \sum_{i=1, \dots, 51} \eta(i) \#_i(X). \end{aligned}$$

$\#_i(X)$  étant le nombre de configurations de type  $i$  dans la réalisation  $X$ .

Les résultats obtenus sont résumés dans le tableau 2.5.

### 2.2.4 Optimisation : Equations Différentielles Stochastiques

<sup>7</sup> Le défaut des méthodes stochastiques réside dans le temps de calcul nécessaire à l'optimisation. Les méthodes MCMC consistent à proposer une nouvelle configuration à partir de la configuration courante, celle-ci étant acceptée avec une certaine probabilité. Lorsqu'une proposition est refusée, les calculs nécessaires à son évaluation relativement à la configuration courante peuvent être considérés comme perdus. L'échantillonneur de Gibbs permet un taux d'acceptation égal à 1 mais nécessite bien souvent des calculs lourds pour le noyau de proposition. Nous étudions ici une alternative aux méthodes MCMC par l'intermédiaire des équations différentielles stochastiques. Le problème pour lequel nous évaluons les deux approches est la restauration d'image, ou plus précisément le débruitage. Nous considérons un processus de diffusion avec interactions et la dynamique de Langevin associée. Ce processus de Markov est stationnaire et réversible pour la mesure de Gibbs associée au modèle de restauration. Pour simuler la dynamique de Langevin, nous considérons des processus de Markov discrets dans le temps, appelés approximation d'Euler et approximation explicite forte de Taylor. Le principe général consiste à coupler une approximation de la dynamique de Langevin avec un critère d'optimisation fournissant un estimateur de l'espérance de la mesure de Gibbs ou du minimum global de l'hamiltonien. Ces outils sont comparés à la dynamique de Metropolis-Hastings, notamment pour un faible nombre d'itérations.

**L'approche par Equation Stochastique Différentielle-** Comme pour l'approche bayésienne fondée sur les champs de Gibbs, nous considérons un hamiltonien défini par la somme d'un terme d'interactions, le modèle a priori, et un terme d'attache aux données, modélisant la vraisemblance. Nous considérons un  $\Phi$ -modèle pour ses propriétés de préservation des discontinuités.

Soit la trame de l'image  $\Lambda \subset \mathbb{Z}^2, |\Lambda| = m$  et un espace d'état continu  $S \subset \mathbb{R}$ . Supposons que  $S$  est un sous-ensemble compact de  $\mathbb{R}^1$ , alors  $\Omega = S^m$  est l'espace des configurations. L'hamiltonien du modèle a la forme suivante :

$$H(X, Y) = \Phi_1(X) + \Phi_2(X, Y), \quad (2.83)$$

où  $X, \theta \in \Omega$ ,  $X = \{X_i, i \in \Lambda\}$ , et  $Y = \{Y_i, i \in \Lambda\}$  est une configuration fixée (les données),

$$\Phi_1(X) = \beta \sum_{(i,j) \in \Lambda^2: |i-j|=1} U_1(X_i - X_j), \quad (2.84)$$

$$\Phi_2(X, \theta) = \lambda \sum_{i \in \Lambda} (X_i - \theta_i)^2, \quad (2.85)$$

$\beta > 0$  et  $\lambda > 0$  sont les paramètres du modèle.

Le modèle test considérée est le suivant :

$$U_1^{(2)}(X_i - X_j) = -\frac{1}{1 + \frac{(X_i - X_j)^2}{d^2}}. \quad (2.86)$$

L'idée générale de la méthode est, tout comme pour l'approche par MCMC, de construire l'image résultat comme la configuration limite  $\hat{X} \in \Omega$  d'un schéma itératif  $\{\hat{X}_n\}$ . La  $n$ -ième configuration de ce schéma est déterminée par la configuration précédente  $\hat{X}_{n-1}$  et l'hamiltonien du modèle. La construction de ce processus s'effectue en trois étapes :

1. Nous construisons, en premier lieu, un processus de diffusion dans le temps avec interactions, continu sur l'espace des mesures, et qui converge vers la mesure de Gibbs définie par l'hamiltonien  $H(X, Y)$  de l'équation (2.83).

2. Nous considérons ensuite une approximation discrète dans le temps de ce processus par un processus de Markov discret.

---

<sup>7</sup> Ce travail est le résultat d'une collaboration avec E. Zhizhina de l'IITP dans le cadre d'un projet soutenu par l'institut Lyapunov.



3. Pour finir, nous définissons un estimateur qui optimise en un certain sens l'hamiltonien considéré. Le processus discret est couplé avec un schéma particulier (recuit simulé ou moyennage) pour obtenir la configuration qui minimise l'hamiltonien (nommé critère du MAP) ou permet d'obtenir l'espérance de la distribution de Gibbs (que nous nommons critère EXP).

Soit le processus de diffusion  $X(t) = \{X_i(t), i \in \Lambda\}$ , correspondant à la première étape de notre construction. Considérons pour un  $\sigma > 0$  quelconque donné un processus stationnaire  $\{X(t), t \geq 0\}$  sur  $\Omega$  ayant pour mesure invariante :

$$d\mu_\sigma(X) = \frac{e^{-\frac{2}{\sigma^2}H(X,\theta)}}{Z_\Lambda(\sigma)} d\mu_0(X), \quad (2.87)$$

où  $\mu_0 = \times_{i \in \Lambda} \nu_0^{(i)}$  est le produit des distributions uniformes  $\nu_0$  sur  $S$ . Ce processus est décrit par son générateur  $\mathcal{L}$  qui est un opérateur dans l'espace de Hilbert  $\mathcal{L}_2(\Omega, d\mu_\sigma)$  des fonctions sur  $\Omega$ :

$$L_\sigma f = \frac{1}{2}\sigma^2 \Delta f - \nabla H \cdot \nabla f = \frac{\sigma^2}{2} \sum_{i \in \Lambda} \frac{\partial^2 f}{\partial x_i^2} - \sum_{i \in \Lambda} \frac{\partial H}{\partial x_i} \cdot \frac{\partial f}{\partial x_i}, \quad f \in \mathcal{L}_2(\Omega, d\mu_\sigma).$$

L'opérateur  $L_\sigma$  satisfait la condition de l'équilibre minutieux<sup>8</sup>, ce qui est équivalent au fait que l'opérateur  $L_\sigma$  est auto-adjoint dans l'espace  $\mathcal{L}_2(\Omega, d\mu_\sigma)$  [Lig85]. Par conséquent, l'opérateur  $\mathcal{L}$  génère un processus stochastique réversible  $X^\sigma(t)$  sur  $\Omega$  relativement à  $\mu_\sigma$ . Ce processus est appelé la dynamique de Langevin et est obtenu comme solution de l'équation différentielle stochastique suivante :

$$dX^\sigma(t) = a(X^\sigma(t))dt + \sigma dW(t), \quad t \geq 0, \quad (2.88)$$

où :

$$a(X) = a(X, Y) = -\nabla_X H(X, Y)$$

est un terme de dérive déterministe dépendant des données  $\theta$ ,  $dW(t)$  est un terme de diffusion,  $W = \{W(t), t \geq 0\}$  est le processus de Wiener à  $m$  dimensions. Pour simplifier les notations, nous omettons  $\sigma$ , de sorte que  $X \equiv X^\sigma$  par la suite. La solution de l'équation (2.88) peut être écrite comme suit :

$$X_i(t) = X_i(s) + \int_s^t a_i(X(u), \theta) du + \sigma \int_s^t dW(u), \quad i = 1, \dots, m \quad (2.89)$$

avec  $m = |\Lambda|$ ,  $0 < s < t$ .

**Algorithmes-** Pour simuler le processus, nous devons discrétiser le temps. Nous présentons deux schémas de discrétisation.

• *L'approximation d'Euler-* Nous considérons une discrétisation de l'intervalle de temps  $(0, t)$ :

$$\tau(\delta) = \{\tau_n, \quad n = 0, 1, 2, \dots, n_t\}$$

par un pas temporel  $\delta_n = \tau_{n+1} - \tau_n = \delta$ . Le processus d'approximation :

$$Z_n = \{Z_i(n)\}, \quad i = 1, \dots, m; \quad n = 0, 1, \dots, n_t$$

possède le même état initial  $X(0)$  que le processus  $X(t)$ , et est construit par le schéma itératif suivant :

$$\begin{aligned} Z_i(0) &= X_i(0), \\ Z_i(n+1) &= Z_i(n) + a_i(Z(n), Y) \delta_n + \sigma (W(\tau_{n+1}) - W(\tau_n)). \end{aligned} \quad (2.90)$$

8. "Detailed balance condition" en anglais

9. "drift term" en anglais

$W(\tau_{n+1}) - W(\tau_n)$  est simulée en échantillonnant une loi normale centrée de variance  $\delta$ , notée  $\mathcal{N}(0, \delta)$ .

**Remarque.** Remarquons que le processus  $Z(n)$  peut être défini par la probabilité de transition suivante :

$$P(Z(n) \rightarrow Z(n+1)) = \frac{k^m}{(2\pi\delta_n\sigma^2)^{m/2}} \exp \left\{ -\frac{1}{2\delta_n\sigma^2} \sum_{i=1}^m (Z_i(n+1) - Z_i(n) - a_i(Z(n), Y)\delta)^2 \right\},$$

où  $k^m$  est une constante de normalisation, et  $\sigma$  un paramètre donné qui peut être interprété comme la température de la mesure de Gibbs. Cette formule fait référence à un échantillonnage parallèle. Les résultats restent valides dans le cadre de l'échantillonnage séquentiel utilisé pour les simulations.

• *L'approximation forte de Taylor-* En appliquant la formule de Taylor stochastique [PK92], nous obtenons d'après l'équation (2.89) un processus appelé schéma d'approximation forte de Taylor :

$$\begin{aligned} Z_i(0) &= X_i(0), \\ Z_i(n+1) &= Z_i(n) + a_i(Z(n), \theta) \delta_n + \frac{1}{2} a_i(Z(n), \theta) a'_i(Z(n), \theta) \delta_n^2 + \\ &\quad \frac{1}{4} a''_i(Z(n), \theta) \sigma^2 \delta_n^2 + a'_i(Z(n), \theta) \sigma \Delta V_i(n) + \sigma \Delta W_i(n), \end{aligned} \tag{2.91}$$

Avec :

$$\Delta V_i(n) = \frac{1}{2} \left( \xi_i^{(1)} + \frac{1}{\sqrt{3}} \xi_i^{(2)} \right) \delta_n^{3/2}, \quad \Delta W_i(n) = \xi_i^{(1)} \sqrt{\delta_n},$$

$\xi_i^{(1)}, \xi_i^{(2)}$ ,  $i = 1, \dots, m$  sont des variables aléatoires indépendantes, identiquement distribuées (v.a.i.i.d.) par  $\mathcal{N}(0, 1)$ .

Nous considérons également une modification de l'approximation forte de Taylor, appelée approximation forte de Taylor explicite :

$$\begin{aligned} \tilde{Z}_i(0) &= X_i(0), \\ \tilde{Z}_i(n+1) &= \tilde{Z}_i(n) + \frac{1}{2} \left( a_i(\gamma(n), \theta) + a_i(\tilde{Z}(n), \theta) \right) \delta_n + \sigma \Delta W_i(n), \end{aligned} \tag{2.92}$$

avec

$$\gamma(n) = \tilde{Z}(n) + a(\tilde{Z}(n), \theta) \delta_n + \sigma \Delta \tilde{W}(n),$$

et

$$\Delta W_i(n) = \xi_i^{(1)} \sqrt{\delta_n}, \quad \Delta \tilde{W}_i(n) = \xi_i^{(2)} \sqrt{\delta_n},$$

$\xi_i^{(1)}, \xi_i^{(2)}$  sont des v.a.i.i.d. par  $\mathcal{N}(0, 1)$ .

Pour effectuer les différents tests, nous avons considéré cette seconde approximation, en plus du schéma d'ordre 1 (approximation d'Euler). Ce second schéma permet d'éviter le calcul de dérivées d'ordre supérieur de l'hamiltonien qui peuvent induire des instabilités numériques.

**Convergence et propriétés d'ergodicité-** Nous énonçons les propriétés de convergence des processus d'approximation donnés par les équations (2.90) à (2.92). Supposons que pour tout  $i = 1, \dots, m$

1)  $a_i(X, Y) \in C(S^{2m})$  est une fonction continue de  $X$  et de  $Y$ ,

2)  $a_i(X, Y)$  ne dépend des configurations  $X$  et  $\theta$  que dans un voisinage du site  $i$ :

$$a_i(X, Y) = a_i(X_j, Y_j), \quad j : |i - j| \leq c_0$$

pour une constante  $c_0$ .

Notons que, d'après notre définition des  $a_i(X, Y)$  et de  $H(X, Y)$ , où

$$a_i(X, Y) = -\frac{\partial}{\partial X_i} H(X, Y),$$

ces deux hypothèses sont vérifiées.

**Théorème 1 (convergence forte).**

$$\text{Soit } \delta = \max_{k=1, \dots, n_t} \delta_k.$$

alors, sous les hypothèses 1)-2) sur les fonctions  $a_i(X, Y)$ ,  $i = 1, \dots, m$ , le processus d'approximation (2.90) converge fortement vers le processus  $X(t)$  à l'ordre  $1/2$  c'est-à-dire, pour tout  $t$ , nous avons la propriété suivante :

$$\max_i E(|X_i(t) - Z_i(n_t)|) \leq C\delta^{1/2} \quad (2.93)$$

$C$  étant une constante positive qui ne dépend pas de  $\delta$  (mais dépend de  $t$ ).

Par application du développement de Taylor-Ito, et en considérant des restrictions plus fortes sur les dérivées des fonctions  $a_i(X, Y)$ , nous avons un résultat similaire pour les processus (2.91)-(2.92) aux ordres 1.5 et 1.0 respectivement.

**Corollaire (Convergence uniforme faible).** Pour chaque  $f \in C(S^m)$  et chaque  $t > 0$ , nous avons :

$$\max_{X \in S^m} \sup_{0 \leq s \leq t} |E_X f(Y([s/\delta])) - E_X f(X(s))| \rightarrow 0 \quad \text{as } \delta \rightarrow 0, \quad (2.94)$$

où  $E_X$  est l'espérance si  $X$  est l'état initial.

**Diagnostic de convergence-** Nous comparons la vitesse de convergence des schémas étudiés avec l'algorithme de Metropolis-Hastings. Pour ce faire, nous effectuons  $n$  simulations indépendantes et considérons l'itération  $m$ . Pour tester la convergence, nous comparons les statistiques des  $m_0$  échantillons suivants, obtenus avec les différentes simulations indépendamment, avec les statistiques de l'ensemble défini par l'union des échantillons obtenus pour les  $n$  simulations. Ces statistiques sont calculées sur un seul pixel.

Considérons un pixel donné. Soit  $x_k^i$  sa valeur à l'itération  $k$  pour la simulation  $i$ . Considérons maintenant l'ensemble  $X_m^i = \{x_k^i, k = m, \dots, m + m_0\}$  pour un  $m_0$  fixé. Nous supprimons de chacun de ces ensemble les  $\alpha\%$  valeurs les plus faibles et les  $\alpha\%$  valeurs les plus fortes. Soient  $A_m^i$  et  $B_m^i$  les valeurs minimale et maximale des ensembles résultants. Considérons maintenant l'ensemble  $X_m = \cup_{i=1, \dots, n} X_m^i$  auquel on enlève les  $\alpha\%$  valeurs les plus faibles et les  $\alpha\%$  valeurs les plus fortes. Soient  $A_m$  et  $B_m$  les valeurs minimale et maximale de cet ensemble. Nous définissons le rapport suivant :

$$R_m = \frac{B_m - A_m}{\sum_{i=1, \dots, n} A_m^i/n - \sum_{i=1, \dots, n} B_m^i/n}$$

Si l'algorithme converge, nous avons :

$$\lim_{m \rightarrow \infty} R_m = 1$$

Nous traçons  $R_m$  en fonction de  $m$  et considérons que la convergence est atteinte lorsque  $R_m$  est proche de 1.

Nous comparons les approximations de la dynamique de Langevin avec l'algorithme de Metropolis-Hastings. La figure 2.18 montre un exemple sur un pixel de l'évolution de  $R_m$  en fonction de  $m$

( $m_0 = 100$ ) pour le modèle étudié. Nous pouvons remarquer que l'approximation d'Euler montre la convergence la plus rapide et l'algorithme de Metropolis-Hastings la plus lente. Notons qu'une itération de l'algorithme de Metropolis-Hastings et de l'approximation d'Euler nécessite le même temps de calcul alors que l'approximation forte de Taylor explicite nécessite un temps de calcul double. Ceci montre l'intérêt de considérer la dynamique de Langevin en concurrence des algorithmes de type MCMC pour échantillonner un modèle de type champ de Gibbs.

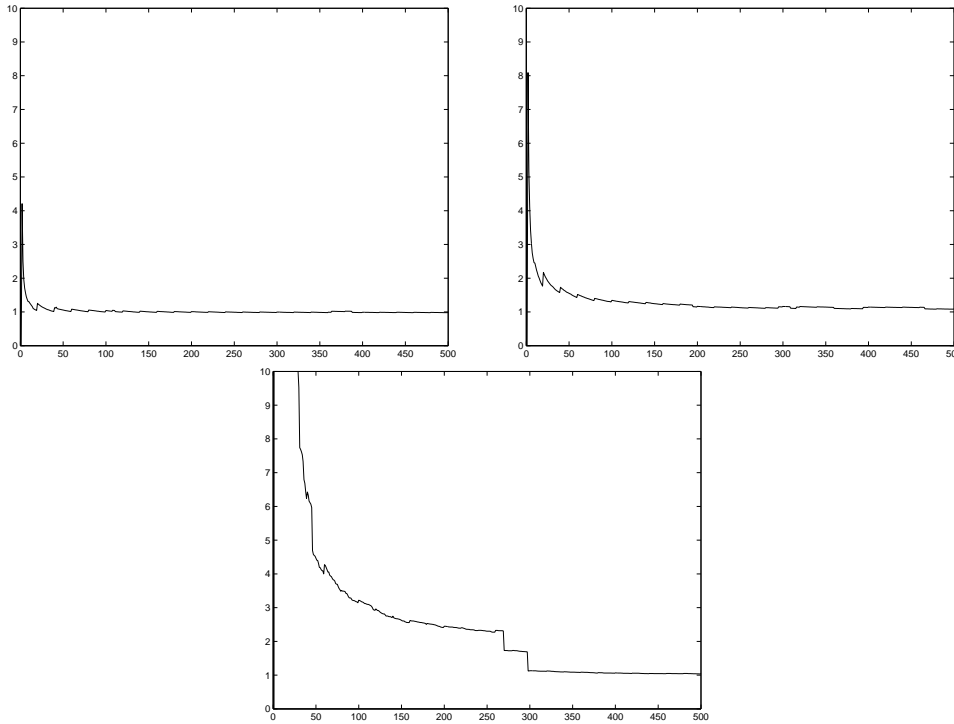


FIG. 2.18 –  $R_m$  en fonction de  $m$  ( $m_0 = 100$ ) pour respectivement l'approximation d'Euler, l'approximation explicite forte de Taylor et l'algorithme de Metropolis-Hastings

**Application à la restauration d'image** Nous considérons maintenant le problème de la restauration d'image pour lequel un simple échantillonnage de la mesure de Gibbs n'est pas suffisant, puisque la solution est obtenue par l'optimisation du modèle. Nous écrivons deux critères d'optimisation pour lesquels nous donnons des estimateurs fondés sur les dynamiques étudiées.

Le premier critère considéré est le critère MAP :

$$\hat{X}_1 = \arg \min_X H(X, \theta) \quad (2.95)$$

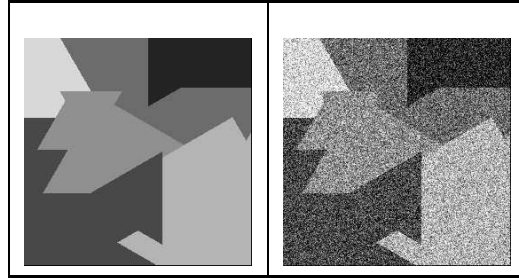
Pour estimer le MAP nous appliquons un recuit simulé en faisant décroître le paramètre au cours des itérations. Il a été montré que, pour la dynamique de Langevin, la solution de l'équation (2.88) converge vers  $\hat{X}_1$  quand  $t \rightarrow \infty$  pour  $\sigma = \sigma(t) = \frac{1}{\sqrt{\ln(t+2)}}$  [PK92]. La valeur optimale (au sens de la vitesse de convergence) du pas de discrétisation  $\delta$  dépend de la valeur de  $\sigma$ . Nous introduisons donc également un schéma de décroissance sur  $\delta$ . En pratique, nous avons utilisé deux schémas exponentiels de décroissance.

De par les résultats énoncés précédemment, nous pouvons considérer un second critère (EXP) fondé sur l'estimation de l'espérance :

$$\hat{X}_2 = E_\mu(X) \quad (2.96)$$

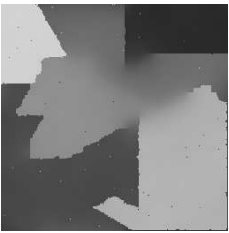
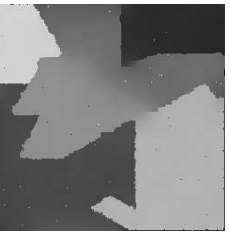
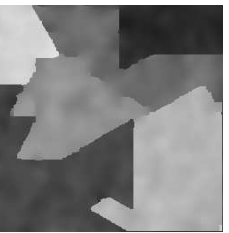
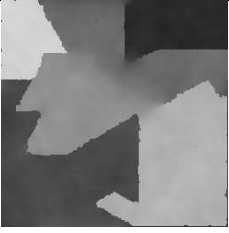

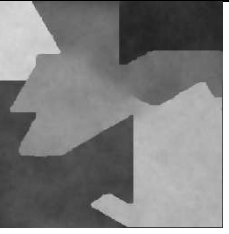
Pour obtenir l'estimée  $\hat{X}_2$ , nous considérons un nombre d'itérations de la dynamique de Langevin, pour un  $\sigma$  donné, suffisant pour atteindre la convergence. Nous effectuons alors un certain nombre d'itérations supplémentaires pour obtenir plusieurs échantillons de la mesure. L'estimée  $\hat{X}_2$  est alors obtenue par moyennage sur ces échantillons.

**Résultats sur une image test-** Considérons, en premier lieu, l'image constituée de zones uniformes du tableau 2.6.a à laquelle est ajoutée un bruit gaussien centré d'écart type 50 (voir le tableau 2.6.b).



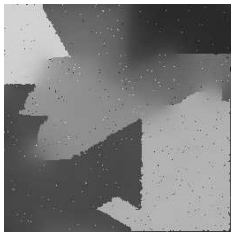
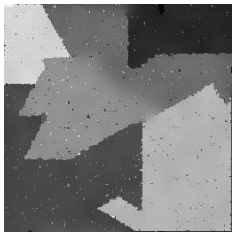


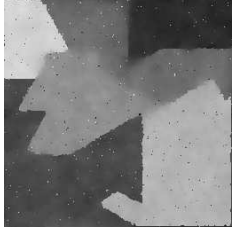
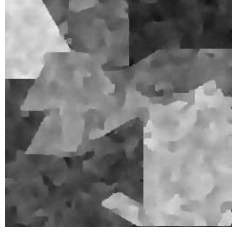
TAB. 2.6 – a (à gauche): Première image test, b (à droite): Image bruitée  $\sigma = 50$

Nous comparons les résultats obtenus avec les deux critères (MAP et EXP) par l'algorithme de Metropolis-Hastings (AMH) et les dynamiques de Langevin discrétisées (DLD) pour différents nombres d'itérations (voir les tableaux 2.7 pour 10000 itérations et 2.8 pour 1000 itérations).

critère MAP :			
	Euler $\sigma : 1000 \rightarrow 0.001$ $\delta : 1000 \rightarrow 0.1$ $H_f = -245404$	Taylor $\sigma : 1000 \rightarrow 0.001$ $\delta : 1000 \rightarrow 0.1$ $H_f = -245385$	Metropolis $T : 1000 \rightarrow 0.001$ $H_f = -245018$
			
critère EXP :	Euler $T = 1.0, \delta = 10$ $H_f = -244928$	Taylor $T = 1.0, \delta = 10$ $H_f = -244954$	Metropolis $T = 0.5$ $H_f = -241139$

TAB. 2.7 – Résultat des différents algorithmes et critères pour un grand nombre d'itérations (10000),  $\beta = 2.0$ ,  $d = 10$ ,  $\lambda = 0.0001$

La convergence par les DLD est donc plus rapide que l'AMH pour les deux critères. De plus, les minima locaux obtenus par les deux types d'approches pour un faible nombre d'itérations ont des propriétés différentes. Avec les DLD, les résultats obtenus après 1000 itérations sont très proches de ceux

critère MAP :	 Euler $\sigma : 10 \rightarrow 0.01$ $\delta : 1000 \rightarrow 5$ $H_f = -242820$	 Taylor $\sigma : 10 \rightarrow 0.01$ $\delta : 1000 \rightarrow 5$ $H_f = -241719$	 Metropolis $T : 500 \rightarrow 0.001$ $H_f = -221939$
critère EXP :	 Euler $T = 1.0, \delta = 20$ $H_f = -237813$	 Taylor $T = 1.0, \delta = 20$ $H_f = -238169$	 Metropolis $T = 0.5$ $H_f = -229612$

TAB. 2.8 – Résultat des différents algorithmes et critères pour un petit nombre d’itérations (1000),  $\beta = 2.0$ ,  $d = 10$ ,  $\lambda = 0.0001$


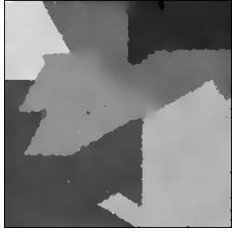
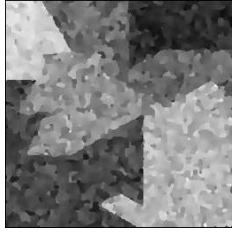
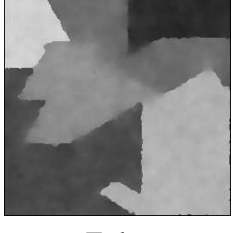
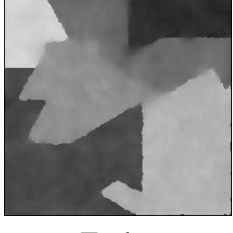
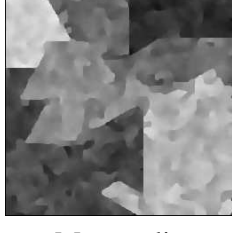
obtenus avec 10000 itérations sauf en certains points isolés contenant un bruit résiduel, ce phénomène étant moins important pour le critère EXP. Avec l’AMH, nous n’obtenons pas ce bruit résiduel mais les zones sont beaucoup moins homogènes avec 1000 itérations qu’avec 10000 itérations. Pour supprimer ce bruit résiduel, il suffit d’appliquer un filtre médian en post-traitement. Le tableau 2.9 présente le résultat précédent après une étape de filtrage.

Le filtrage améliore de façon évidente les résultats de DLD, mais reste quasiment sans effet sur les résultats de l’AMH. Pour finir, notons que l’approximation de Taylor n’améliore pas les résultats obtenus par l’approximation d’Euler.

Ces différents commentaires se retrouvent sur les valeurs des énergies obtenues pour les différentes expérimentations et synthétisées dans les tableaux 2.10 et 2.11.

Ces courbes montrent également la convergence plus rapide des DLD. Même si l’optimum global n’est pas atteint, les valeurs de l’énergie obtenues par les DLD sont plus faibles que par l’AMH. En outre, le filtre médian réduit ces valeurs pour une valeur proche de l’optimum global pour les DLD.

**Tests sur des données réelles-** Considérons d’abord une image radar ERS1, 4-vues, montrée sur la figure 2.19. Le bruit contenu dans cette image (bruit de chatoiement) est un bruit multiplicatif, corrélé. Néanmoins, nous conservons ici le modèle quadratique additif, sachant que la littérature fait état de résultats satisfaisant avec ce modèle simple; le but étant ici de comparer les dynamiques de Langevin et de Metropolis sur cet exemple très bruité pour un faible nombre d’itérations (respectivement 3000 et 300 pour les tableaux 2.12 et 2.13). Si nous considérons à la fois la qualité visuelle et la valeur de l’énergie, l’AMH fournit des résultats légèrement meilleurs pour 3000 itérations. Avec un filtre médian en post-traitement, les résultats sont équivalents. Le point principal est que si nous réduisons le nombre d’itérations à 300, le schéma DLD est plus robuste et produit un résultat à peine dégradé par rapport à celui obtenu avec 3000 itérations, ce qui n’est pas le cas pour l’AMH. De plus, une fois encore, un filtre

critère MAP :	 Euler $\sigma : 10 \rightarrow 0.01$ $\delta : 1000 \rightarrow 5$ $H_f = -243499$	 Taylor $\sigma : 10 \rightarrow 0.01$ $\delta : 1000 \rightarrow 5$ $H_f = -243147$	 Metropolis $T : 500 \rightarrow 0.001$ $H_f = -219963$
critère EXP :	 Euler $T = 1.0, \delta = 20$ $H_f = -241021$	 Taylor $T = 1.0, \delta = 20$ $H_f = -241233$	 Metropolis $T = 0.5$ $H_f = -230552$

TAB. 2.9 – Résultat des différents algorithmes et critères pour un petit nombre d'itérations (1000) suivi par un filtre médian  $3 \times 3$ ,  $\beta = 2.0$ ,  $d = 10$ ,  $\lambda = 0.0001$

médian améliore le résultat du DLD et nous obtenons une valeur de l'énergie très proche de la valeur optimale.

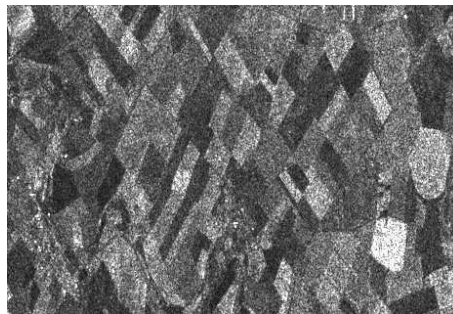
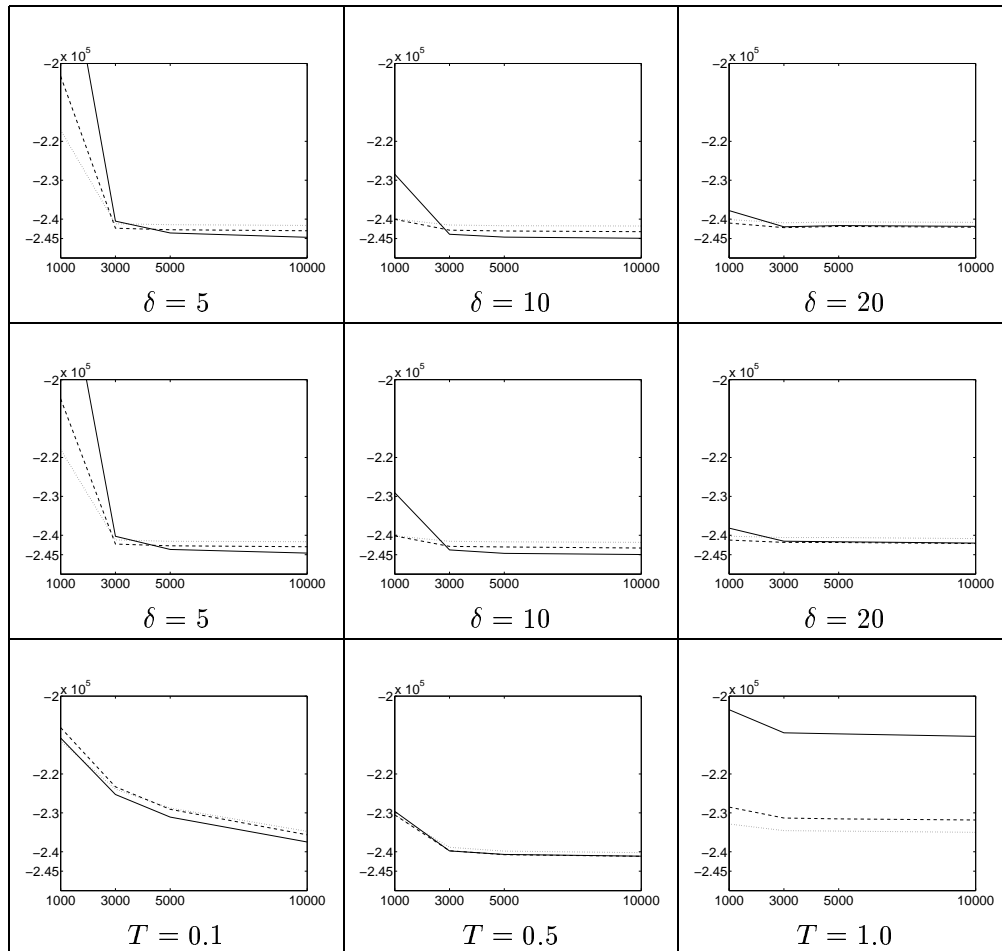


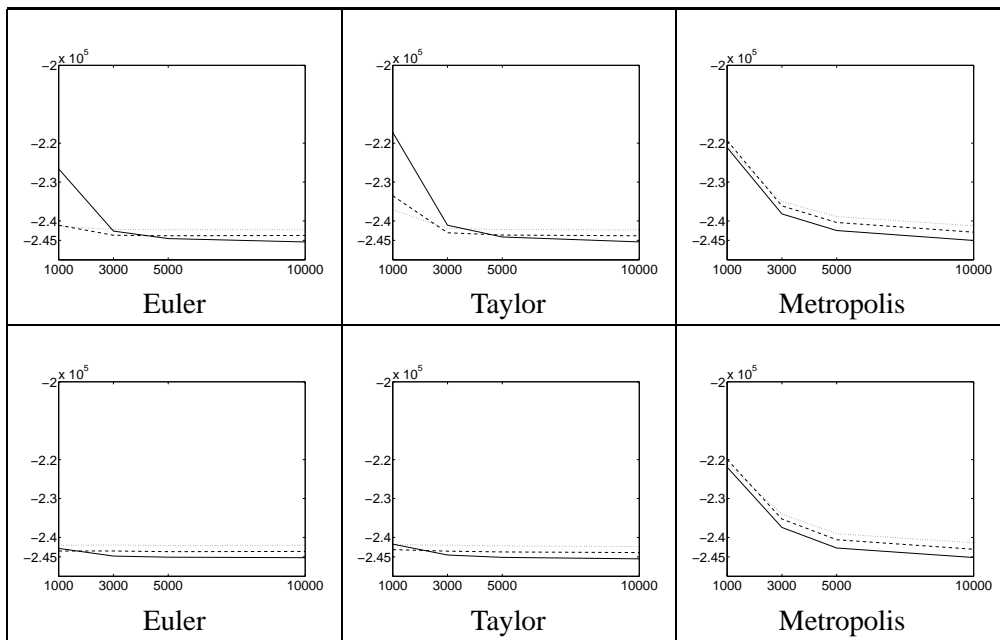
FIG. 2.19 – Image ERS1 4-vues4

Pour finir, nous considérons l'image de Lena à laquelle nous ajoutons un bruit gaussien d'écart type 50 (voir le tableau 2.14). Nous considérons uniquement 50 itérations ce qui conduit à un temps CPU de 8 secondes sur un processeur de 1 GHz pour une image  $256 \times 256$ . Ici encore, le DLD donne de meilleurs résultats, le critère EXP étant légèrement meilleur (voir les tableaux 2.15 et 2.16).

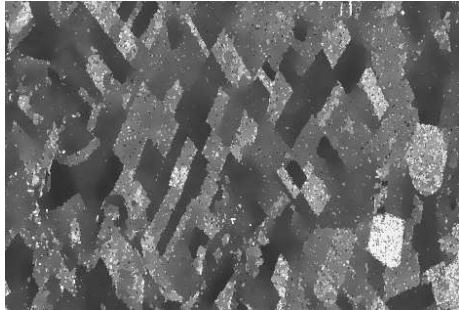
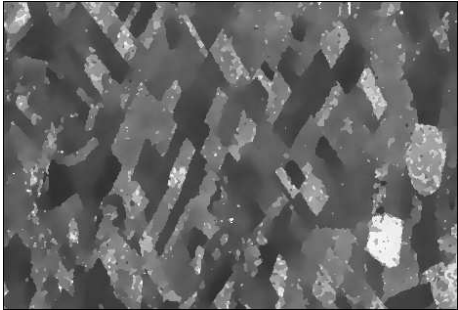
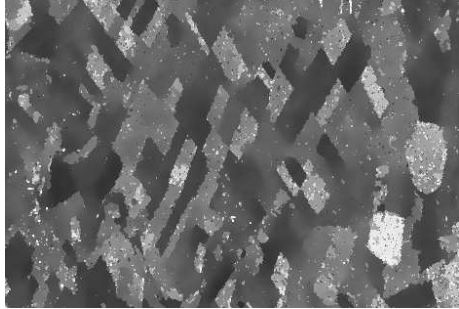
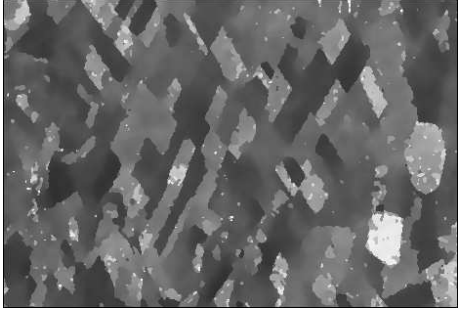
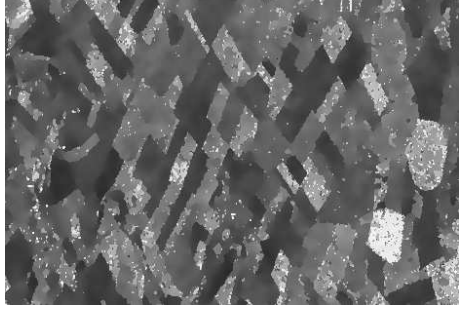
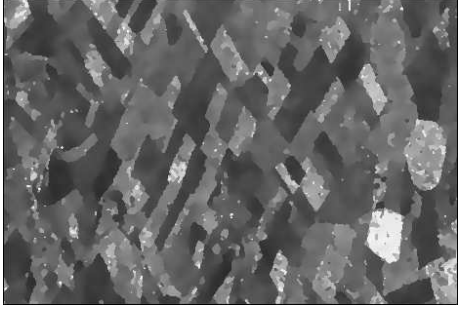
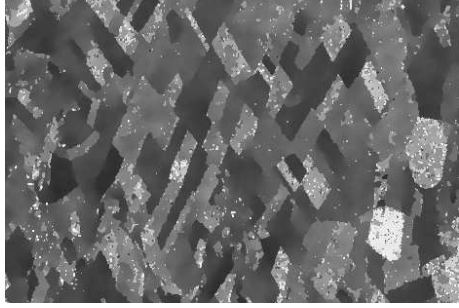
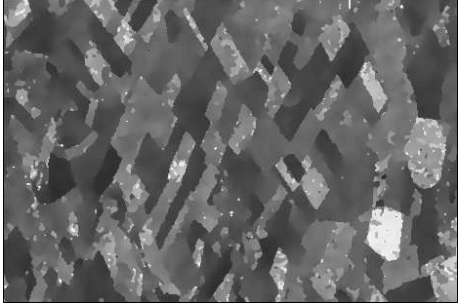


TAB. 2.10 – Energie en fonction du nombre d'itérations pour le critère EXP et respectivement Euler (ligne du haut), Taylor (ligne du milieu) et Metropolis (ligne du bas) sans filtrage (courbes pleines) et après un filtre médian  $3 \times 3$  (courbes en tirets) et  $5 \times 5$  (courbes en pointillés)

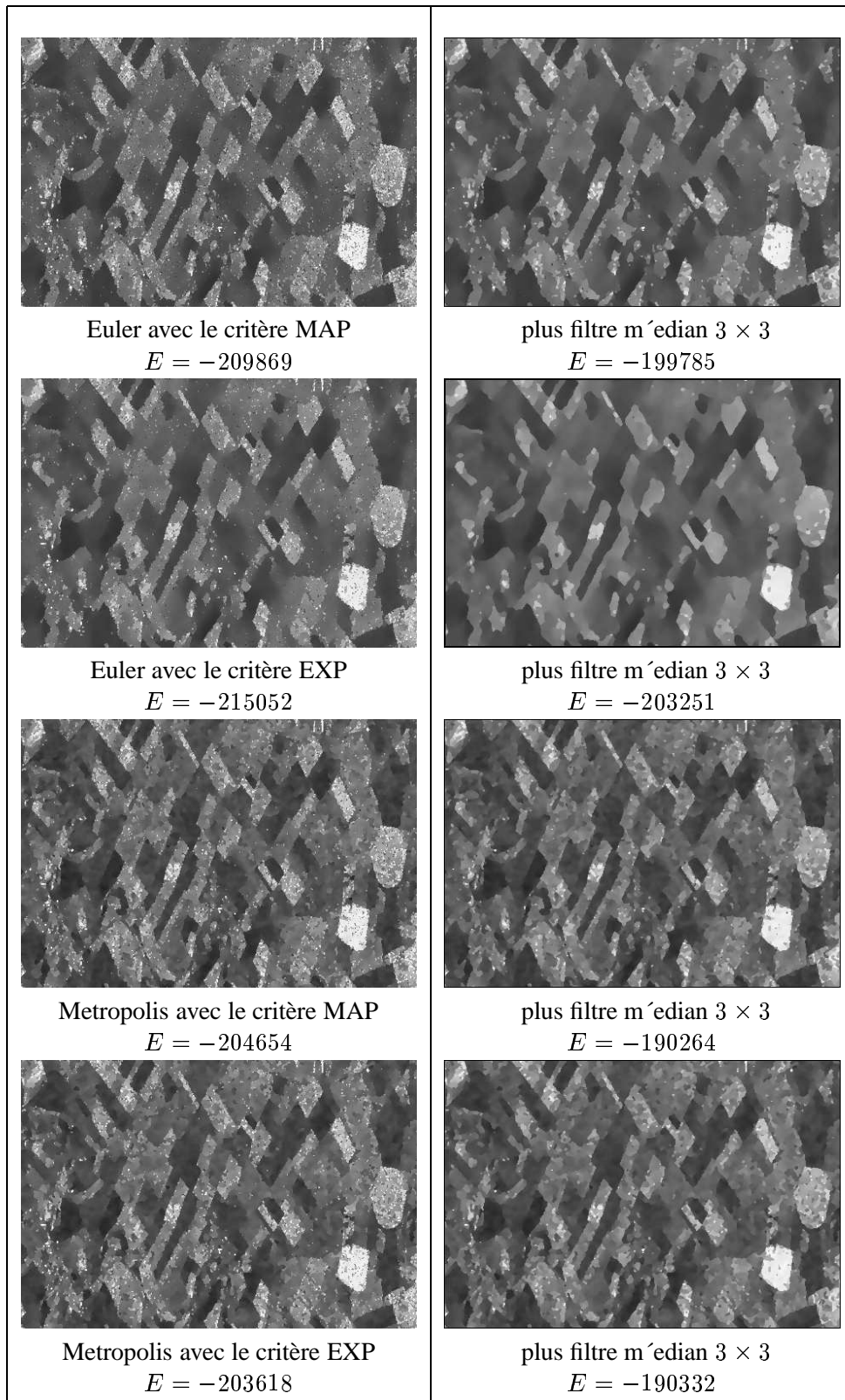




TAB. 2.11 – *Energie en fonction du nombre d'itérations pour le critère MAP : ligne du haut, pour  $T = 1000 \rightarrow 0.001$  (Metropolis) et  $T = 10 \rightarrow 0.01, \delta = 1000 \rightarrow 0.01$  (Euler et Taylor), et ligne du bas, pour  $T = 500 \rightarrow 0.001$  (Metropolis)  $T = 10 \rightarrow 0.01, \delta = 1000 \rightarrow 5.0$  (Euler et Taylor) sans filtrage (courbes pleines) et après un filtre médian  $3 \times 3$  (courbes en tirets) et  $5 \times 5$  (courbes en pointillés)*

	
<p>Euler avec le critère MAP <math>E = -213845</math></p>	<p>plus filtre médian <math>3 \times 3</math> <math>E = -201382</math></p>
	
<p>Euler avec le critère EXP <math>E = -216915</math></p>	<p>plus filtre médian <math>3 \times 3</math> <math>E = -203780</math></p>
	
<p>Metropolis avec le critère MAP <math>E = -220075</math></p>	<p>plus filtre médian <math>3 \times 3</math> <math>E = -202661</math></p>
	
<p>Metropolis avec le critère EXP <math>E = -218675</math></p>	<p>plus filtre médian <math>3 \times 3</math> <math>E = -203780</math></p>

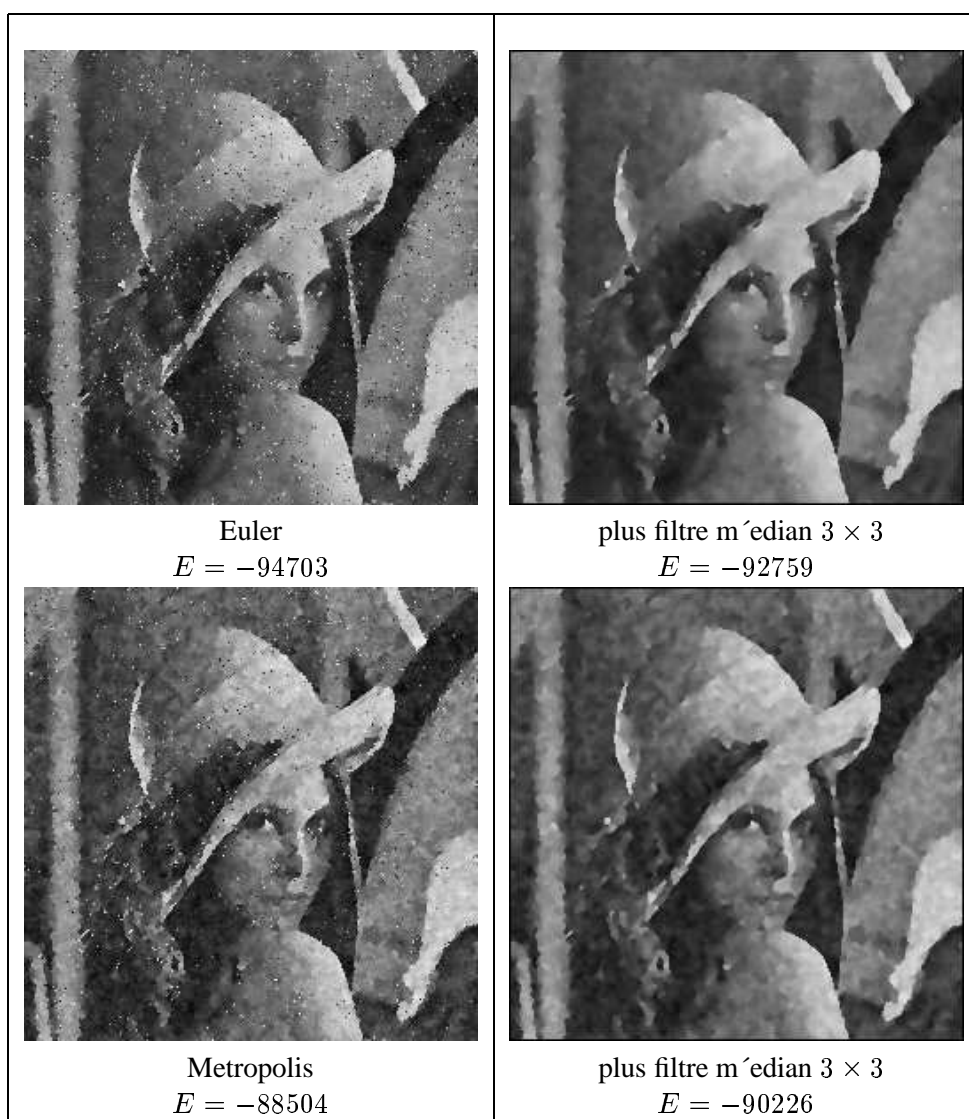
TAB. 2.12 – Résultats sur l'image de la figure 2.19 pour 3000 itérations



TAB. 2.13 – Résultats sur l'image de la figure 2.19 pour 300 itérations



TAB. 2.14 – Image de Lena  $256 \times 256$  et sa version bruitée ( $\sigma = 50$ )



TAB. 2.15 – Résultats sur l'image du tableau 2.14 pour 50 itérations avec le critère MAP



TAB. 2.16 – Résultats sur l'image du tableau 2.14 pour 50 itérations avec le critère EXP.

## 2.3 Quelques applications traitées

### 2.3.1 La texture

#### La variance conditionnelle

<sup>10</sup> Nous abordons ici l'utilisation des champs de Markov en tant que modèle de texture. L'application traitée consiste à extraire un masque urbain à partir d'une image satellitaire. Différentes approches ont été proposées dans la littérature pour résoudre ce problème. On retrouve les matrices de cooccurrence [Har79], qui s'avèrent bien adaptées sur les images LANDSAT mais moins performantes à des résolutions meilleures [BP90], la morphologie mathématique [SM94] qui permet une extraction grossière du masque urbain [Ser89], des techniques de fusion d'informations alliant imageries optique et radar [HG91], ou encore les analyses multi-échelle [WMCL97].

Nous avons repris et généralisé une approche développée dans notre thèse qui consiste à modéliser la texture par un champ markovien gaussien 4-connexe isotrope [34]. Les paramètres de ce champ nous ont permis de discriminer la texture urbaine de façon satisfaisante comme le montre l'étude comparative dans [Gou96]. L'approche a été validée sur des images SPOT à 10m de résolution. Avec des données plus résolues, par exemple SPOT V, de nouvelles structures apparaissent clairement (serres, vergers,...). Ces structures sont à l'origine de fausses alarmes dans l'approche isotrope. Le travail résumé ici (cf thèse d'A. Lorette [Lor99]) prend donc en compte une analyse directionnelle pour pallier ce défaut.

Nous considérons huit modèles gaussiens à voisinage directionnel, c'est à dire que chaque site possède deux voisins définis par les deux sites les plus proches suivant une direction donnée. Les lois conditionnelles locales sont alors gaussiennes et leur variance est un indice de l'urbain. Le biais introduit par l'anisotropie de la trame est corrigé par une procédure de renormalisation. Le paramètre de texture retenu est une combinaison des variances conditionnelles des huit modèles directionnels. Nous classifions ensuite le paramètre de texture estimé en chaque point par l'algorithme décrit dans le paragraphe 2.2.1 et régularisons le résultat par un modèle de Potts. Un post-traitement permet d'éliminer les dernières fausses alarmes. L'approche a été validée sur des images SPOT II, des simulations SPOT V, et des images radar ERS1.

**Les modèles-** Nous définissons huit modèles markoviens directionnels (cf figure 2.20) à voisinage monodimensionnel.

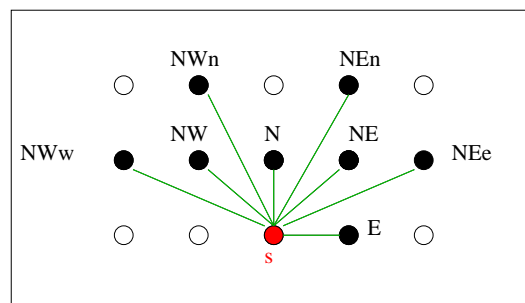


FIG. 2.20 – Les huit directions

Pour chaque modèle associée à une direction  $d$ , le voisinage d'un pixel  $s$  est l'ensemble des deux plus proches voisins de  $s$  dans cette direction. La probabilité conditionnelle en un point sachant ses voisins

10. Nous résumons ici des travaux initiés à l'École Nationale Supérieure des Télécommunications en collaboration avec M. Sigelle, et pleinement développés et étendus dans la thèse d'A. Lorette, sur un financement et en collaboration avec le CNES. Bibliographie : [56, 34, 53, 55]

s'écrit alors :

$$P(X_s | X_r, r \in V^d(s)) = \frac{1}{Z_{V^d(s)}} \exp \left( -\beta^{(d)} \left( \sum_{r \in V^d(s)} (X_s - X_r)^2 + \lambda^{(d)} (X_s - \mu)^2 \right) \right) \quad (2.97)$$

$V^d(s)$  étant le voisinage de  $s$  dans la direction  $d$ .  $Z_{V^d(s)}$  est la fonction de partition.

En supposant des conditions aux bords toriques, nous pouvons remarquer à partir de l'équation (2.97) que la probabilité conditionnelle  $P(X_s | X_r, r \in V^d(s))$  dépend uniquement de la moyenne  $m_s^d$  des deux voisins de  $s$  :

$$P(X_s | X_r, r \in V^d(s)) = \frac{1}{Z'_{V^d(s)}} \exp \left( -\beta^{(d)} (2 + \lambda^{(d)}) \left( X_s - \frac{2m_s^d + \lambda^{(d)}\mu}{2 + \lambda^{(d)}} \right)^2 \right) \quad (2.98)$$

$$= P(X_s | m_s^d) \quad (2.99)$$

De plus, la probabilité conditionnelle  $P(X_s | m_s^d)$  suit une loi gaussienne définie par :

$$P(X_s | m_s^d) \equiv \mathcal{N} \left( \frac{2m_s^d + \mu\lambda^{(d)}}{2 + \lambda^{(d)}}, \frac{1}{2\beta^{(d)}(2 + \lambda^{(d)})} \right) \quad (2.100)$$

Les indices texturaux considérées sont les variances conditionnelles  $\frac{1}{\beta} = \frac{1}{2\beta^{(d)}(2 + \lambda^{(d)})}$  estimées pour chacun des huit modèles par la méthode dite des *queues de comète* (le nom provient de la forme des matrices conditionnelles estimées) [7]. Pour une distribution donnée, ces paramètres ne dépendent pas de  $m_s^d$ . En outre, la méthode des *queues de comètes* nous permet d'estimer le paramètre  $\lambda$  qui sera utile pour l'étape de renormalisation. Une heuristique permet de prendre en compte la non-stationarité de la moyenne locale et évite ainsi le phénomène de flou dû à la fenêtre d'estimation. Cette heuristique consiste à ne considérer dans l'estimation de la variance conditionnelle que les distributions les plus représentatives, c'est à dire celles correspondant aux moyennes des voisins les plus caractéristiques de la région d'étude.

**Normalisation des paramètres-** Pour pallier l'anisotropie de la trame carrée, nous normalisons les paramètres estimés. Les modèles considérés ont un voisinage mono-dimensionnel. La distance entre deux voisins dépend de la direction étudiée. Nous considérons une résolution plus fine pour laquelle chacune des résolutions des directions étudiées peut être obtenue par décimation. C'est à dire que la distance entre deux voisins peut être approximée par la distance du réseau fin multipliée par un entier. En renormalisant le modèle, nous pouvons, à partir de l'estimation du paramètre suivant une direction, obtenir le paramètre du modèle équivalent sur le réseau fin. Les paramètres des différentes directions sont donc ramenés à une résolution identique.

**Résultats expérimentaux-** Les estimées des huit paramètres obtenues à partir de la figure 2.21 sont montrées sur la figure 2.22. Les zones urbaines sont caractérisées par une forte réponse dans toutes les directions. Notons que les frontières entre les régions sont nettes grâce à l'heuristique décrite. Le paramètre final retenu est défini par la valeur moyennes des deux valeurs centrales. Cette combinaison a été obtenue par expérimentation et confirmée par une analyse en composantes principales.

**Segmentation et post-traitements-** Le paramètre retenu est classifié par l'algorithme de "fuzzy C-means" avec prise en compte d'un terme d'entropie décrit au paragraphe 2.2.1. Une segmentation est ensuite obtenue par un modèle de Potts. A ce niveau, certaines fausses alarmes demeurent. Pour les supprimer, nous considérons la variance conditionnelle la plus faible parmi les huit directions. L'image



FIG. 2.21 – *Toreilles: simulation SPOT V*

obtenue est classifiée, puis filtrée par un filtre médian. Nous obtenons alors une image de marqueurs. Les zones urbaines sont alors constituées des composantes connexes de la première segmentation ayant une intersection non vide avec l'image de marqueurs. Un exemple du paramètre de texture retenu sur une simulation SPOT V est donné sur la figure 2.23 (a). L'image de marqueurs est sur la figure 2.23 (b) et le résultat final sur la figure 2.23 (c). La robustesse de l'approche est montrée par les résultats obtenus sur une image SPOT 3 de Cayenne (voir figure 2.24) et sur une image ERS 1 de Macapa (voir figure 2.25). Les résultats sont donc précis pour différents capteurs, ce qui semble montrer que le paramètre de texture retenu est caractéristique de la structure urbaine à des résolutions allant de 2-3 mètres à 10-20 mètres, quelque soit le capteur.

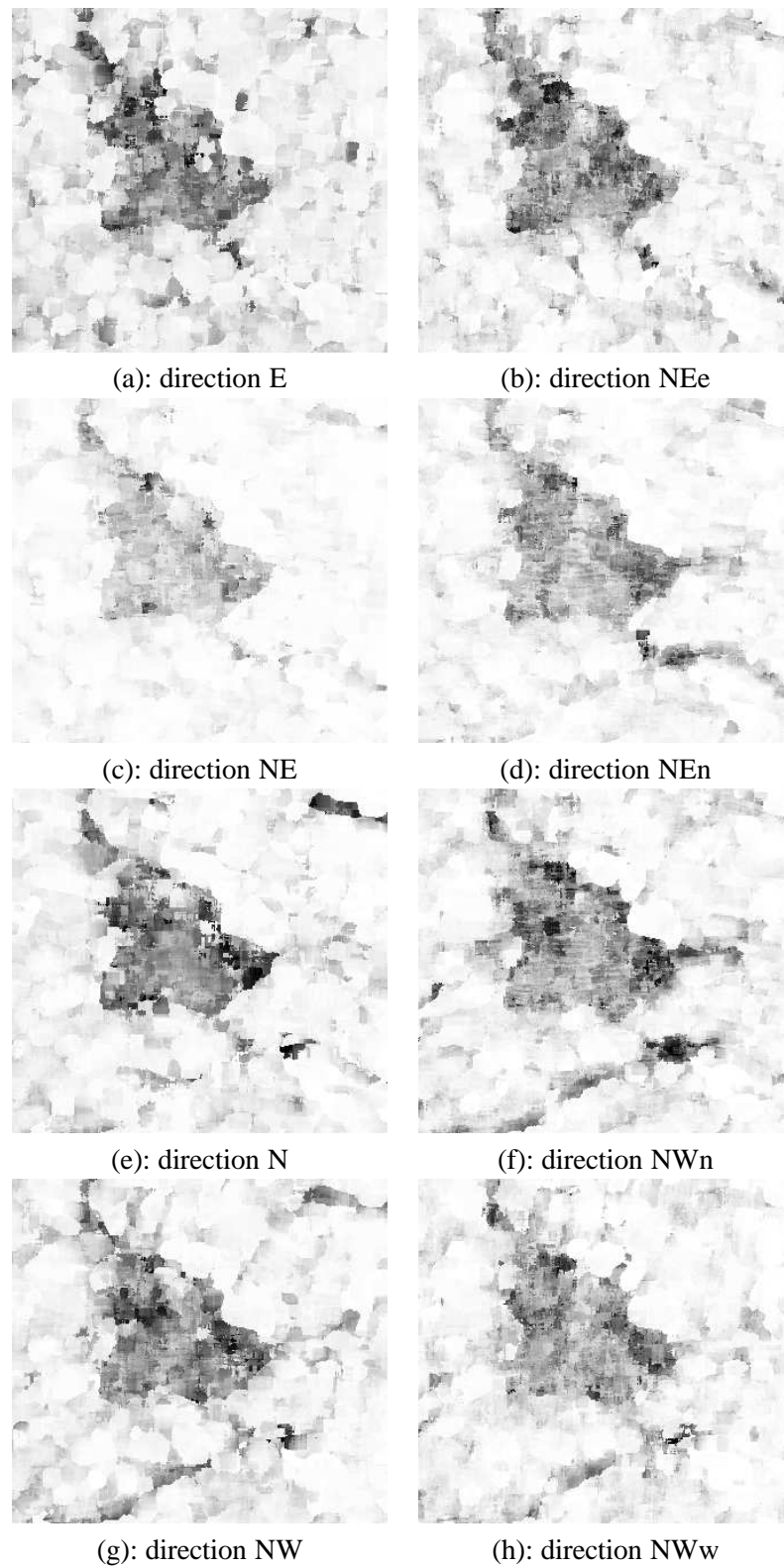
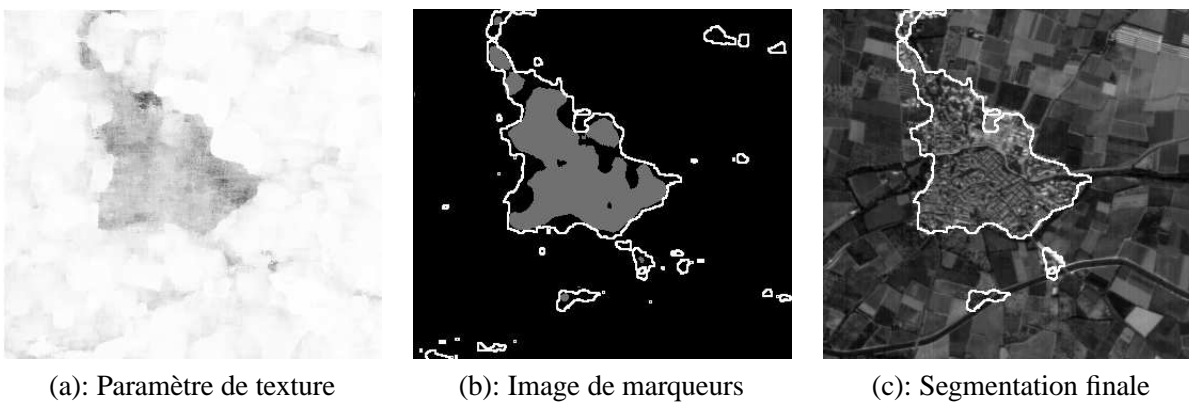
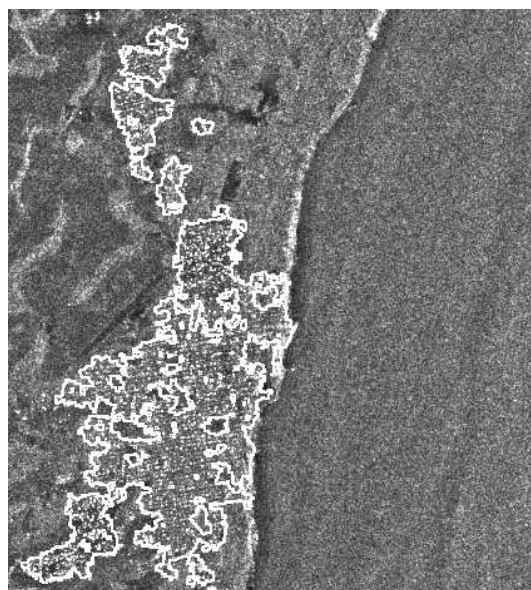


FIG. 2.22 – Paramètres de texture estimés (résultat en inversion vidéo)

FIG. 2.23 – *Toreilles: simulation SPOT V*FIG. 2.24 – *Cayenne (département de La Guyanne): image SPOT III*FIG. 2.25 – *Macapá (Brésil): image radar ERS 1*

### Données de grande dimension

<sup>11</sup> Les images hyperspectrales, par opposition aux images monospectrales, sont des images de télédétection ayant la particularité de fournir à la fois un échantillonnage spatial et un échantillonnage spectral. Pour chaque échantillon spatial, elles ne fournissent plus une valeur scalaire, mais un ensemble de valeurs (qui peut être vu comme un vecteur), correspondant aux mesures du rayonnement faites dans un certain nombre de longueurs d'onde. Ces images constituent par conséquent un "pavé" hyperspectral dans lequel chaque point peut être indexé par ses deux coordonnées spatiales et sa coordonnée spectrale (ie. la longueur d'onde correspondante).

Ces images ont un nombre de canaux en général bien supérieur à 10. Chacun de ces canaux est associé à une longueur d'onde du spectre allant du visible à l'infrarouge. Par rapport aux images multispectrales (dont le nombre de canaux est supérieur à 1 et inférieur à 10), ce découpage spectral plus fin autorise des mesures radiométriques plus précises, et permet donc une caractérisation plus exacte du matériau présent sur le terrain. Ceci autorise en particulier des applications liées à la reconnaissance du sol en géologie, à l'aide à l'agriculture de précision, etc. . .

Dans ce travail, on s'intéresse à l'analyse des textures présentes dans des images hyperspectrales tirant parti à la fois des relations spatiales et spectrales de celles-ci, visant ainsi à ajouter l'information spectrale aux méthodes classiques d'analyse de texture prenant en compte les variations spatiales des niveaux de gris.

L'application dans laquelle s'inscrit ce travail est la classification des zones urbaines. En effet, ces zones sont constituées de nombreux matériaux, ce qui est la source d'une grande variabilité de réflectance, qui se traduit à la fois entre différents pixels et à l'intérieur d'un même pixel dont la réflectance peut alors être due à un mélange de plusieurs composantes "pures".

De plus, une partie de ces matériaux est aussi caractéristique de certains paysages naturels, donnant à certains pixels urbains les mêmes signatures spectrales que ces paysages naturels. De ce fait, l'étude des réponses radiométriques de ces zones est insuffisante pour aboutir à un résultat satisfaisant. Il est donc nécessaire d'étudier les zones urbaines à travers leur texture.

**Modèle markovien gaussien-** Nous avons appliqué le modèle markovien gaussien en imagerie monodimensionnelle pour l'analyse de la texture urbaine. Nous étendons cette approche en définissant un modèle à deux dimensions, que l'on appelle MGMRF (en anglais "*Multivariate Gaussian Markov Random Field*"), sur les données vectorielles que constituent les observations aux différentes longueurs d'onde d'un même site. Les interactions entre les valeurs à différentes longueurs d'onde ne sont pas modélisées explicitement, mais cependant elles existent, et découlent des paramètres qui sont estimés à partir des données. Il n'est donc pas nécessaire de faire une hypothèse concernant la portée de ces interactions.

Soit une image  $x$  de dimensions spatiales  $N_h \times N_v$ , ayant un nombre de bandes  $N_b$ . Cette image est supposée être une réalisation d'un champ  $X$ . Nous considérons un champ à 2 dimensions (les dimensions spatiales) composé de variables vectorielles de dimension égale au nombre de bandes (les réponses spectrales). La modélisation markovienne implique que la probabilité qu'un vecteur aléatoire  $x_s$  en un site  $s$  prenne la valeur  $x_s$  connaissant le reste de l'image, est égale à la probabilité en ce site connaissant ses voisins. La modélisation markovienne gaussienne ajoute une contrainte sur cette probabilité, qui est supposée être de la forme suivante :

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z} \exp \left\{ -\frac{1}{2} \left\| x_s - \sum_{t \in V_s} \theta_{t-s} x_t \right\|_{\Sigma}^2 \right\} \quad (2.101)$$

la notation  $\|a\|_{\Sigma}^2$  signifiant  $a^t \Sigma^{-1} a$ , les  $\theta_i$  étant des matrices de transformation, ou paramètres d'interac-

11. Nous résumons ici des travaux de la thèse de G. Rellier [Rel02], menés en collaboration avec Alcatel Space Cannes et cofinancés par la région PACA. Bibliographie : [64, 69].

tion et  $\Sigma$  la matrice de covariance conditionnelle. Ce sont les paramètres du modèle.

Cette loi est donc une loi gaussienne dont la covariance est constante et la moyenne d'épnd du voisinage. Cette expression est valable si l'on considère des données centrées, c'est-à-dire de moyenne nulle. Dans le cas de moyennes non nulles, l'expression est légèrement modifiée :

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z} \exp \left\{ -\frac{1}{2} \left\| x_s - \sum_{t \in V_s} \theta_{t-s} x_t - (I_{N_b} - \sum_{t \in V_s} \theta_{t-s}) \mu \right\|_{\Sigma}^2 \right\} \quad (2.102)$$

où  $I_{N_b}$  désigne la matrice identité, et  $\mu$  la moyenne du champ.

Par la suite, on suposera que la texture homogène considérée a été centrée, la généralisation étant triviale. La formulation (2.101) permet d'écrire :

$$x_s = \sum_{t \in V_s} \theta_{t-s} x_t + u_s \quad (2.103)$$

avec  $u_s$  bruit blanc gaussien, centré, de matrice de covariance  $\Sigma$ .

De plus, pour des contraintes de cohérence et de stationnarité du champ, on doit avoir  $\theta_{i-s} = \theta_{s-i}$ , ce qui conduit à :

$$x_s = \sum_{k=0}^K \theta'_k m_s^k + u_s \quad (2.104)$$

où  $k$  désigne chaque type de voisinage (horizontal, vertical, ...),  $K$  le nombre de types de voisins,  $m_s^k$  la moyenne des voisins de type  $k$ , et  $\theta'_k = \theta_k n_k$ , avec  $n_k$  le nombre de voisins de type  $k$ . Un type de voisins désigne, ici, l'ensemble des voisins  $t$  d'un pixel  $s$  ayant la même contribution dans le modèle, ce qui se traduit par le même paramètre  $\theta_{s-t}$ .

Pour obtenir la formule de la loi globale, on revient sur la formule (2.103) :

$$u_s = x_s - \sum_{t \in V_s} \theta_{t-s} x_t \quad (2.105)$$

Soit  $x$  l'image entière réorganisée en un vecteur colonne,  $x = (x_{(0,0)}^t \dots x_{(N_v, N_h)}^t)^t$ , de même  $u = (u_{(0,0)}^t \dots u_{(N_v, N_h)}^t)^t$ . Il vient alors :

$$u = Ax \quad (2.106)$$

où la structure de  $A$  est déterminée par les paramètres  $\theta$  et le système de voisinage considéré.

Le vecteur  $u$  suit une loi normale, donc le vecteur  $x$  également. Nous pouvons écrire :

$$P(x) = \frac{1}{Z(S)} \exp \left( -\frac{1}{2} x^t S^{-1} x \right) \quad (2.107)$$

avec  $S = E\{xx^t\} = A^{-1}E\{uu^t\}$  et  $Z(S)$  la constante de normalisation.

**Caractéristiques des espaces de grande dimension-** Après avoir présenté ce modèle dans sa forme générale, on s'intéresse plus particulièrement à son application dans le cadre de la classification des images AVIRIS à 224 bandes spectrales.

Le processus de classification tel qu'on l'envisage est constitué dans un premier temps d'une phase d'apprentissage durant laquelle les classes à distinguer sont définies par un opérateur, puis caractérisées par l'estimation de leurs paramètres.

Le nombre de paramètres du modèle markovien gaussien est variable, selon l'ordre du voisinage, son anisotropie, ou les contraintes appliquées à la matrice  $\theta$ . Dans le cas de données AVIRIS à 224 bandes, l'estimation d'un nombre aussi élevée de paramètres est très difficile, d'autant plus que le nombre d'échantillons disponibles pour effectuer le calcul ne peut excéder quelques milliers. Or, plus le nombre de paramètres est élevée, plus le phénomène de Hughes [Hug68] est présent, entraînant des instabilités du processus d'estimation.

Deux solutions peuvent être apportées à ce problème. D'une part, il est possible de choisir des critères et des méthodes permettant de réduire la dimension des observations de façon à améliorer estimation et classification. D'autre part, il est possible de simplifier le modèle de façon à ce que le nombre de paramètres à estimer diminue radicalement.

Dans le cas de l'application qui nous intéresse, la simplification consistera à donner une certaine forme *a priori* aux matrices de transformation  $\theta_i$  :

$$\theta_i = \text{diag}(a_i) = a_i \text{Id} \quad (2.108)$$

Ceci consiste à considérer que l'on a d'une part indépendance entre les bandes dans les interactions entre pixels, et d'autre part que ces interactions sont identiques dans les différentes longueurs d'onde du spectre. Cela se traduit par une même vitesse de décroissance des corrélations dans les différentes bandes.

Cette hypothèse a également pour conséquence de simplifier la forme de la probabilité conditionnelle :

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z} \exp \left\{ -\frac{1}{2} \|x_s - \sum_{t \in V_s} a_{t-s} x_t\|_{\Sigma}^2 \right\} \quad (2.109)$$

Nous pouvons alors écrire :

$$P(x_s | \{x_t\}, t \in V_s) = \frac{1}{Z'} \exp \left\{ -\sum_{t \in V_s} \alpha_{t-s} \|x_s - x_t\|_{\Sigma}^2 - \beta \|x_s\|_{\Sigma}^2 \right\} \quad (2.110)$$

Pour que le champ  $X$  de loi (2.107) existe, il faut que la matrice  $S$  soit définie positive, par conséquent il faut que  $A$  soit définie positive si l'on considère que  $\Sigma$  l'est. Cette dernière hypothèse est vraisemblable, si l'on considère un nombre d'échantillons suffisamment élevée.

Par la suite, on prendra la formulation de la probabilité conditionnelle suivante :

$$P(x_s | \{m_k\}) = \frac{1}{Z} \exp \left\{ -\frac{1}{2} \|x_s - \sum_k a'_k m_s^k\|_{\Sigma}^2 \right\} \quad (2.111)$$

où les  $a'_k$  sont obtenus à partir des  $a_k$  comme les  $\theta'$  à partir des  $\theta$  (cf. équation (2.104)).

Nous pouvons montrer qu'une condition nécessaire à l'existence du champ markovien gaussien de paramètres  $\{a'_k\}$  est que  $\sum_{k=0}^{K-1} a'_k < 1$ , où  $K$  désigne le nombre de voisins [Haz00].

**Classification de texture-** In fine, le but de cette analyse de texture est la classification, en particulier celle de zones urbaines, qui sont texturées (par opposition aux zones rurales, qui sont plus homogènes à la résolution considérée).

Nous souhaitons définir une méthode pour laquelle l'influence de la radiométrie et de la texture sont équivalentes. Pour ce faire, il paraît plus judicieux de passer par des méthodes où des images de texture sont extraites et ajoutées aux images initiales de radiométrie. L'algorithme de classification proposée ici est divisé en deux phases. La première phase est un pré-traitement qui consiste en une réduction de la dimension des données d'une part, et en une extraction des paramètres de texture d'autre part. La deuxième phase est la classification elle-même.

Comme nous venons de l'expliquer, le pré-traitement est l'étape d'extraction de caractéristiques. Cette étape est constituée de deux sous-étapes bien distinctes, qui ont chacune un but précis. La première sous-étape consiste à chercher un sous-espace de l'espace initial dans lequel projeter les données. Son but est de limiter les problèmes liés à l'estimation statistique dans des espaces de grande dimension. L'autre sous-étape est l'extraction des paramètres de texture nécessaires à l'analyse, accompagnée également d'une réduction de dimension.

La sous-étape de réduction de la dimension est la phase clé de notre analyse. Beaucoup de méthodes ont été développées dans ce cadre ou adaptées dans ce but, la plus connue étant l'analyse en composantes principales. Cette méthode vise à déterminer le sous-espace qui minimise l'erreur quadratique totale pour une dimension donnée. Cette méthode est très utilisée en raison de son efficacité dans le cas général. Cependant, dans le cadre de la classification, d'autres méthodes, utilisées seules ou en combinaison avec l'ACP, sont plus performantes. Parmi celles-ci, on peut retenir les méthodes de poursuite de projection.

La poursuite de projection est une méthode visant à rechercher un sous-espace dans lequel un indice de projection est maximisé [FT74].

Cette méthode est intéressante pour deux raisons :

- le phénomène de Hughes est atténué du fait que tous les calculs sont faits dans le sous-espace projeté,
- l'indice de projection peut être adapté selon le traitement visé.

Initialement [Hub85], la poursuite de projection était utilisée dans des contextes non paramétriques, afin de trouver des angles de projection pour lesquels les données seraient "bien" représentées. Les indices de projection étaient alors des mesures liées à l'histogramme du nuage de points.

Plus récemment, cette méthode a été utilisée dans un contexte paramétrique dans le cadre de la classification d'image hyperspectrale [JL99]. Ayant choisi un modèle gaussien non-contextuel, les auteurs proposent d'appliquer une approximation de la poursuite de projection afin de trouver le sous-espace maximisant la distance de Bhattacharyya entre les classes définies au préalable à travers des échantillons d'apprentissage.

Nous avons proposé une variante de ce dernier algorithme, fondée sur la modélisation par MGMRF, utilisant une distance de Bhattacharyya liée au modèle. Néanmoins, dans cette application, nous nous contentons de l'indice de projection de Jimenez et Landgrebe [JL99] pour ce qui concerne la réduction de dimension des images de radiométrie. Quant à l'indice de projection optimisé pour les images de textures, nous y reviendrons dans le paragraphe suivant.

Le critère qui dirige la phase de classification est le maximum de vraisemblance. Soit  $X$  l'image pré-traitée, destinée à être classifiée. En chaque pixel  $s$ , la mesure vectorielle est constituée de la mesure radiométrique  $\hat{\mu}_s$ , et de la mesure de texture  $x_r^s$ .

Considérons un problème à  $L$  classes  $C_l$ ,  $l = 0 \dots L - 1$ . La règle de décision du maximum de vraisemblance (MV) consiste à assigner le label  $\hat{\lambda}_s$  au pixel  $s$  de la façon suivante :

$$\hat{\lambda}_s = \arg \max_{l \in \{0, L-1\}} P_l(x^s) = \arg \max_{l \in \{0, L-1\}} P_l(x_r^s, x_t^s) \quad (2.112)$$

où  $P_l(x^s) = P(x^s / \lambda_s = l)$ .

Or, nous pouvons écrire :  $P(x_r^s, x_t^s) = P_l(x_r^s / x_t^s) P_l(x_t^s)$ . En outre, nous faisons l'hypothèse que la mesure de radiométrie est indépendante de la texture, connaissant le label du pixel observé, c'est-à-dire :  $P_l(x_r^s / x_t^s) = P_l(x_r^s)$ . Par conséquent,  $P(x_r^s, x_t^s) = P_l(x_r^s) P_l(x_t^s)$ . Et, il vient :

$$\hat{\lambda}_s = \arg \max_{l \in \{0, L-1\}} P_l(x_r^s) P_l(x_t^s) \quad (2.113)$$

La partie radiométrie du vecteur  $X^s$  est supposée suivre une loi conditionnelle normale  $\mathcal{N}(\mu, \Gamma_l)$ . En ce qui concerne la partie texture de ce vecteur, nous y reviendrons après avoir écrit les paramètres

choisis.

**Extraction de paramètres de texture-** Nous avons donc proposé un modèle d'image texturée multidimensionnelle adaptée aux images hyperspectrales. Ce modèle va servir à écrire les textures, et donc à les caractériser. Les paramètres associés à ce champ sont de deux types : la matrice de covariance conditionnelle  $\Sigma$ , et des paramètres d'interaction qui, dans le cas particulier du MGMRF que nous avons écrit, sont réduits à un vecteur, le vecteur  $a$ .

Nous voulons avoir, en chaque point, une estimation des paramètres de texture. Or, la texture n'est pas une notion pixélique, mais se situe au niveau d'un voisinage local. Nous calculons donc en chaque point les paramètres de texture correspondant à une fenêtre centrée sur ce point.

Le MMSE ("*Minimum Mean Square Error*" en anglais) est l'estimateur minimisant l'erreur quadratique moyenne.

Le MMSE obéit au principe d'orthogonalité, c'est-à-dire que l'erreur d'estimation  $y$  est orthogonale aux données servant à l'estimation. Cela se traduit par l'équation suivante :

$$E\{x_r^t u_s\} = 0, \forall s \neq r \quad (2.114)$$

En explicitant  $u_s$  et en développant, l'équation (2.114) devient :

$$E\{x_r^t x_s\} - \sum_{t \in V_s} a_{t-s} E\{x_r^t x_t\} = 0, \forall s \neq r \quad (2.115)$$

En notant  $c_d = E\{x_s^t x_{s+d}\}$ , l'équation peut se réécrire :

$$c_d - \sum_{t \in V_0} a_t c_{d+t} = 0, \forall d \neq 0 \quad (2.116)$$

Il faut autant d'équations que d'inconnues pour résoudre le système, un choix naturel est donc de prendre le système formé des équations (2.116) pour  $d \in \mathcal{K}$

Pour estimer les paramètres  $a_k$ , il suffit donc d'estimer les coefficients  $c_d$  et  $c_{d+t}$  du champ nécessaires à la résolution du système d'équations, et de le résoudre.

Cette méthode suppose que, dans une fenêtre centrée sur un point, seul un type de texture est représenté. Ceci peut être supposé vrai si la fenêtre est assez petite, mais il est préférable de traiter des tailles de fenêtres suffisamment grandes. La raison en est que si l'on travaille avec des fenêtres trop petites, l'estimation n'est plus fiable (d'une part, du fait que nous pouvons avoir des configurations très particulières ou peu caractéristiques du champ à certains endroits, et d'autre part du fait du phénomène de Hughes).

Malheureusement, en augmentant la taille des fenêtres, nous augmentons la possibilité d'y trouver plusieurs textures. Pour remédier à ce problème, nous considérons une heuristique.

L'heuristique proposée ici est fondée sur le fait que le problème apparaît quand les moyennes des textures sont distantes. Le principe est simple, il consiste à vérifier avant le calcul des paramètres de texture que nous ne sommes pas en présence de deux zones de moyennes éloignées, en déterminant d'abord deux nuages de points au moyen de l'algorithme "*k-means*", puis en testant la distance entre ces deux nuages. Si la distance entre les centres de ces nuages est supérieure à un seuil dépendant de la variance des nuages, alors nous supposons être en présence de deux zones distinctes, et le calcul des paramètres de texture se fait sur le nuage ayant le plus grand nombre de points. Nous ne prenons en compte que les cas où nous avons deux textures, au plus, dans une fenêtre. Les paramètres pertinents pour la classification de la texture urbaine sont donnés par la matrice  $\Sigma$ .

**Retour sur le calcul de l'indice de projection pour la poursuite de projection-** L'attribut de texture conservé est donc la matrice de covariance conditionnelle  $\Sigma$  estimée, c'est-à-dire que nous avons  $\frac{N(N-1)}{2}$  paramètres distincts. Comme pour les bandes de radiométrie, il est nécessaire d'effectuer une réduction



de dimension pour les paramètres de texture. Cette réduction est effectuée par un algorithme de poursuite de projection, dont nous décrivons l'indice de projection.

Lorsque nous avons traité le cas des bandes de radiométrie, nous avons choisi la distance de Bhattacharyya entre les classes prédéfinies comme indice de projection à maximiser. Cette distance est facilement calculable dans le cas de données gaussiennes, et nous la calculons pour les paramètres dont nous disposons.

Dans le cas des paramètres de texture que nous considérons, la modélisation gaussienne n'est pas adaptée. Nous approximations  $\hat{\Sigma}$  par une variable  $\Sigma_w$  telle que  $\frac{N^2}{\sum_s \lambda_s^2} \Sigma_w$  suive une loi de Wishart à  $\frac{N^2}{\sum_s \lambda_s^2}$  degrés de liberté.

La distance de Bhattacharyya entre deux distributions de  $\hat{\Sigma}_1$  et  $\hat{\Sigma}_2$ , telles que  $n_1 \hat{\Sigma}_1$  suive une distribution de Wishart de paramètres  $(\Sigma_1, n_1)$  et que  $n_2 \hat{\Sigma}_2$  suive une distribution de Wishart de paramètres  $(\Sigma_2, n_2)$ , peut se calculer comme suit :

$$B_{12} = \frac{n_1^{\frac{p n_1}{4}} n_2^{\frac{p n_2}{4}} |\Sigma_{12}|^{\frac{n_1+n_2}{4}}}{|\Sigma_1|^{\frac{n_1}{4}} |\Sigma_2|^{\frac{n_2}{4}}} \quad (2.117)$$

où  $\Sigma_{12}^{-1} = \frac{1}{2} (n_1 \Sigma_1^{-1} + n_2 \Sigma_2^{-1})$ .

L'algorithme de réduction de dimension des paramètres de texture est une poursuite de projection s'appuyant sur la distance Bhattacharyya donnée par l'équation (2.117). De même que pour la réduction de dimension des bandes de radiométrie, on va chercher à maximiser la distance entre les distributions des paramètres des échantillons d'apprentissage.

L'algorithme consiste en une technique de poursuite de projection par croissance d'espace. A partir d'un espace vide, on augmente cet espace pas à pas d'une dimension, jusqu'à la dimension désirée. A chaque itération, nous ajoutons à la base courante un vecteur directeur de la droite qui, dans l'orthogonal de la base courante, maximise la distance de Bhattacharyya entre les distributions des paramètres de texture des classes prédéfinies.

**Algorithme de classification-** Dans l'équation (2.113), on note que, pour chaque classe  $l$ , on a deux lois de probabilités,  $P_l(X_r^s = x_r^s)$  et  $P_l(X_t^s = x_t^s)$ , que l'on note  $P_l(x_r^s)$  et  $P_l(x_t^s)$ .

La première loi est celle des pixels de radiométrie, directement mesurés par l'instrument. Cette loi est modélisée par une loi normale.

En ce qui concerne la deuxième loi, nous disposons déjà d'une approximation de la loi conditionnelle  $P_l(x_t^s/a)$ . Il est difficile de modéliser la loi du vecteur  $a$ ,  $p(a)$ , puisque le vecteur est soit issu d'une optimisation numérique sur laquelle nous avons peu de contrôle (MV), soit le quotient de deux fonctions de variables aléatoires (les matrices de corrélation) dont les lois ne sont pas des lois simples.

Nous remarquons que les valeurs de  $a$  sont très concentrées autour des valeurs réelles, ou du moins que la variance est assez faible, spécialement pour les estimations faites sur des images réelles, car elles sont très proches de la limite du domaine. Aussi allons-nous confondre la probabilité conditionnelle avec la probabilité intégrée sur le domaine de  $a$ .

Revenons sur la règle de décision (2.113). Si nous prenons deux fois l'opposé de la log vraisemblance, il vient :

$$\hat{\lambda}_s = \arg \min_{l \in \{0, L-1\}} (-2 \log P_l(x_r^s) - 2 \log P_l(x_t^s)) \quad (2.118)$$

Or :

$$-2 \log P_l(x_r^s) = (x_r^s)^t \Sigma_l^{-1} x_r^s + \log |\Sigma_l| \quad (2.119)$$

Et nous faisons l'approximation suivante :



FIG. 2.26 – 3 bandes extraites d'un collage de 4 textures urbaines AVIRIS.

$$-2 \log P_l(x_l^s) = \text{tr} \left( \Sigma_l^{-1} \hat{\Sigma} \right) - (n_l - p - 1) \log \left| \hat{\Sigma} \right| + n_l \log |\Sigma_l| \quad (2.120)$$

où  $x_l^s$  est le résultat de la réorganisation de  $\hat{\Sigma}$  en un vecteur colonne, et  $n_l$  le nombre de degrés de liberté.

Par conséquent, la règle de décision finale est la suivante :

$$\hat{\lambda}_s = \arg \min_{l \in \{0, L-1\}} \left( (x_r^s)^t \Sigma_l^{-1} x_r^s + \text{tr} \left( \Sigma_l^{-1} \hat{\Sigma} \right) - (n_l - p - 1) \log \left| \hat{\Sigma} \right| + (n_l + 1) \log |\Sigma_l| \right) \quad (2.121)$$

Il est à noter que, dans les applications sur des images réelles, le nombre de degrés de liberté choisi est bien inférieur à la valeur calculée théoriquement, car les champs que nous traitons sont beaucoup plus irréguliers que de vrais MGMRF. Ceci a pour effet de rendre les mesures de  $\hat{\Sigma}$  plus variables. Or, quand le nombre de degrés de liberté est grand, cela signifie que la distribution est plus “piquée” autour de la moyenne, et toute valeur qui s’en éloigne donne une probabilité très faible.

**Résultats sur des images AVIRIS-** Il reste donc à tester l’algorithme de classification proposée sur des données hyperspectrales réelles. Les images sur lesquelles nous travaillons sont des extraits d’une image du capteur AVIRIS prise au dessus de la région de Moffett Field, en Californie. La méthode présentée ici est testée, ainsi que deux autres méthodes de référence. Nous appelons  $ACP - n$  la méthode de classification consistant à réduire l’image en gardant les  $n$  composantes principales, puis à classifier selon le maximum de vraisemblance (MV), en supposant les données gaussiennes sans interaction. Nous appelons  $PP - n$  la méthode proposée par Jimenez et Landgrebe [JL99] consistant à rechercher  $n$  bandes par poursuite de projection, avec comme indice de projection la distance de Bhattacharyya calculée sur les bandes de radiométrie, suivie du même algorithme de classification que pour  $ACP - n$ . Nous comparons également avec la méthode  $PP - n$  suivie d’une régularisation markovienne (par un modèle de Potts) obtenue par recuit simulé. Nous notons cette dernière méthode  $PPR - n$ . Enfin, la méthode que nous avons présentée sera notée  $PP - (n, m)$ , dans le cas où l’on a d’une part réduit la dimension à  $n$  bandes (par la même méthode que dans le cas précédent), et d’autre part extrait les images de paramètres  $\hat{\Sigma}$  calculées sur  $m$  bandes.

La figure 2.26 représente une mosaïque de quatre types différents de texture urbaine. Nous testons les différents algorithmes sur cette image dans un premier temps, pour avoir une idée de la qualité des résultats que nous pouvons obtenir, même si les transitions entre textures qui sont introduites artificiellement sont bien plus abruptes que celles qui existent naturellement dans les images réelles. Donc sur ce type d’image, notre méthode aura un handicap du fait de l’utilisation d’une fenêtre d’analyse.

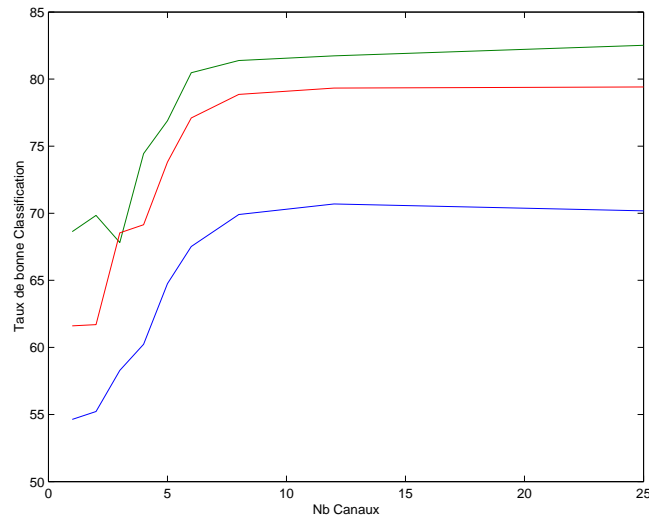


FIG. 2.27 – Classification par différentes méthodes de la mosaïque de textures AVIRIS. En bleu :  $PP - N$ , en rouge :  $PPR - N$ , en vert :  $PP - NN$ .

Les tests sont réalisés sur un sous-ensemble de 25 bandes spectrales parmi les 224 disponibles, prises dans la même zone (visible). Nous comparons les différents algorithmes pour plusieurs valeurs de  $n$ , la dimension finale pour la radiométrie. La valeur de la dimension finale pour la texture est choisie empiriquement égale à  $n$ , car celle-ci donne les meilleurs résultats.

Nous constatons sur la figure 2.27, où les résultats de classification sur la mosaïque de textures AVIRIS sont fournis, que la méthode proposée donne de meilleurs taux de bonne classification, y-compris par rapport à la méthode avec régularisation par modèle de Potts. Pourtant, nous remarquons que les résultats obtenus par la méthode proposée reflètent 2 problèmes. D'une part, la solution n'est pas régulière, et d'autre part, la classification est mauvaise au niveau des zones de transition entre textures. Le premier problème peut être surmonté, par exemple en utilisant une régularisation markovienne. Quant au second, la mise au point d'une meilleure heuristique permettant de détecter les bords reste nécessaire.

Un autre test est mené sur une vraie sous-image de la même image AVIRIS. L'image de test est l'image de la figure 2.28. De la même façon que pour le test précédent, nous utilisons un sous-ensemble des bandes disponibles (ici 40 bandes prises dans le domaine visible). Nous avons défini deux sortes de tissus urbains différents, et trois classes non-urbaines. Nous opérons la classification sur ces cinq classes et le résultat est validé pour les deux classes de zones urbaines qui nous intéressent, sur des zones de validation différentes de celles d'apprentissage. Les zones d'apprentissage et de validation ont été déterminées grâce à des cartes de l'USGS de la région. La taille des ensembles d'apprentissage est de 361 échantillons pour les zones urbaines, et oscille entre 182 et 559 pour les autres. Les courbes de la figure 2.29 donnent l'évolution du coefficient  $Kappa$  ( $\kappa$ ) de la classification pour les différentes méthodes :  $PP - n$ ,  $PP - (n, m)$ ,  $ACP - n$ ,  $PPR - n$ , quand  $n$  évolue. Le coefficient  $Kappa$  est une valeur comprise entre 0 et 1 permettant de mesurer la qualité d'une classification, 1 correspondant à une classification sans erreur.

Pour des raisons de temps de calcul et de limitation de la capacité de mémoire, la classification  $PP - (n, m)$  est une classification  $PP - (n, 3)$ , où nous avons gardé une matrice de covariance conditionnelle calculée sur 3 bandes spectrales. En examinant la courbe de classification, nous remarquons avant tout que la courbe  $PP - (n, 3)$  est bien au-dessus des autres, pour presque toute dimension de l'espace dans lequel la classification est faite. Les valeurs de  $Kappa$  sont assez élevées pour la méthode proposée,

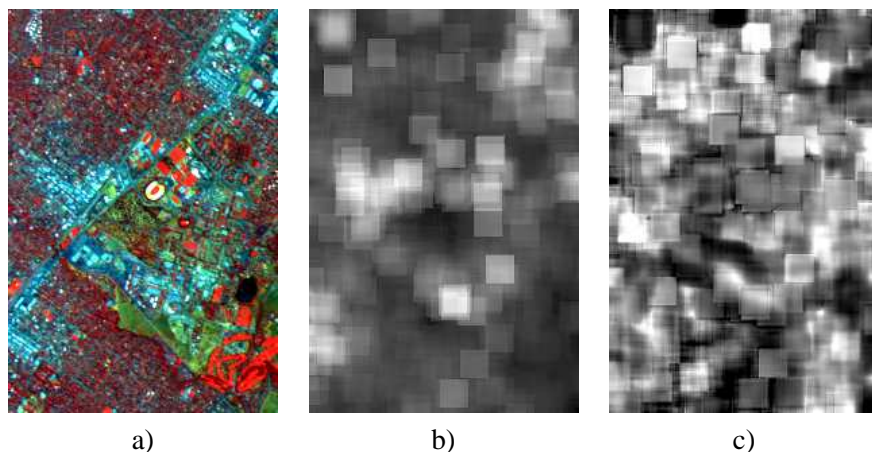


FIG. 2.28 – a) Extrait de 3 bandes visibles de l’image AVIRIS de Moffett Field, b) variance conditionnelle  $\sigma$  extraite d’une bande de cette image, c) paramètre d’interaction  $a$ , extrait d’une bande de cette image.

puisque elles restent supérieures à 0.8, alors que les autres méthodes ne passent pas au-dessus de cette valeur, y-compris la méthode  $PPR - n$ , qui donne de meilleurs résultats que les deux autres méthodes non régularisées.

Comme le montre la figure 2.30, la classification par la méthode proposée apporte une sorte de régularisation fondée sur la texture, même si nous trouvons encore un nombre important de pixels isolés.

Enfin, nous montrons un dernier exemple de classification, cette fois sur une scène entière de l’image AVIRIS de Moffett Field (qui contient 4 scènes de 512 lignes de 614 pixels). Dans cette scène, nous avons défini deux types de zones urbaines, les grands bâtiments et les zones résidentielles, et cinq classes non-urbaines. La première image (a) de la figure 2.31 est un affichage de trois des bandes de l’image sur laquelle nous travaillons. Le test est effectué sur un sous-ensemble de 40 des bandes visibles de l’image originale. Les trois autres images sont les résultats de classification par la méthode  $PP - 10$ , la méthode  $PPR - 10$ , et la méthode  $PP - 10,3$ . L’indice  $Kappa$  des classifications par les deux premières méthodes est  $\kappa_{PP-10} = 0.55$  et  $\kappa_{PPR-10} = 0.62$ , alors que l’indice correspondant à la méthode proposée est nettement supérieur  $\kappa_{PP-10,3} = 0.92$ . Malgré cette grande différence, comme dans les résultats précédents, cette dernière méthode donne des résultats visuels qui ne sont pas toujours satisfaisants en ce qui concerne les transitions entre classes (par exemple au niveau des bords du lac dans la partie en haut à gauche).

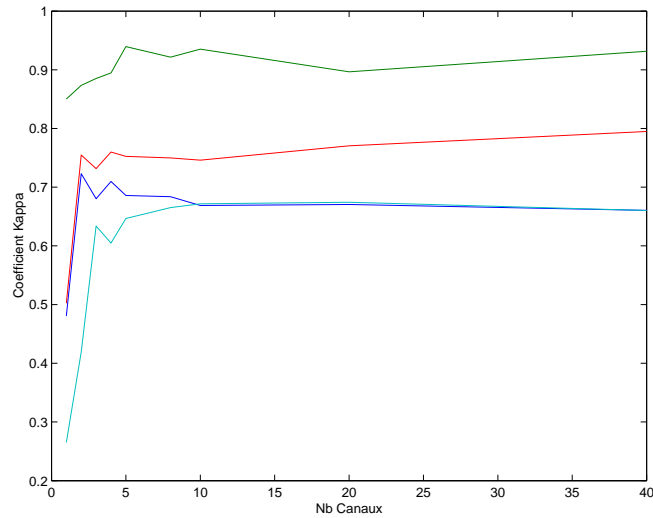


FIG. 2.29 – Classification par différentes méthodes de l'image AVIRIS. En bleu :  $PP - N$ , en rouge :  $PPR - N$ , en vert :  $PP - NN$ , en turquoise :  $ACP - N$ .

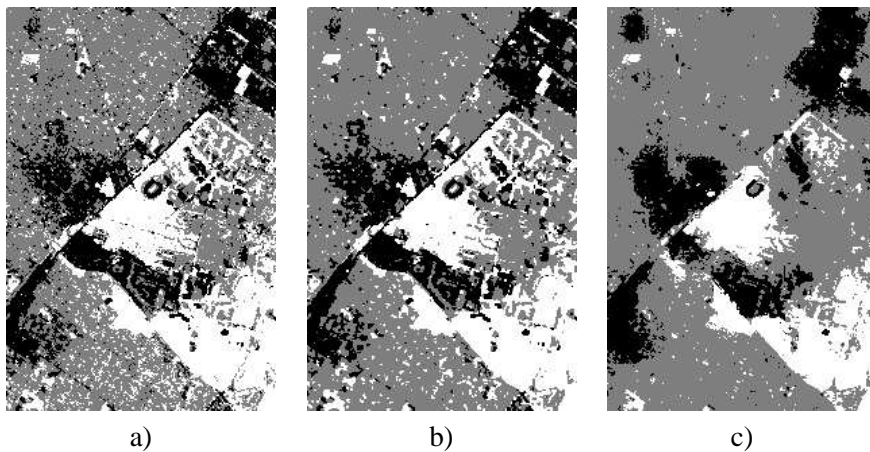


FIG. 2.30 – Classification par 3 méthodes de l'image AVIRIS : a)  $PP - 5$ , b)  $PPR - 5$ , c)  $PP - 5,3$ .

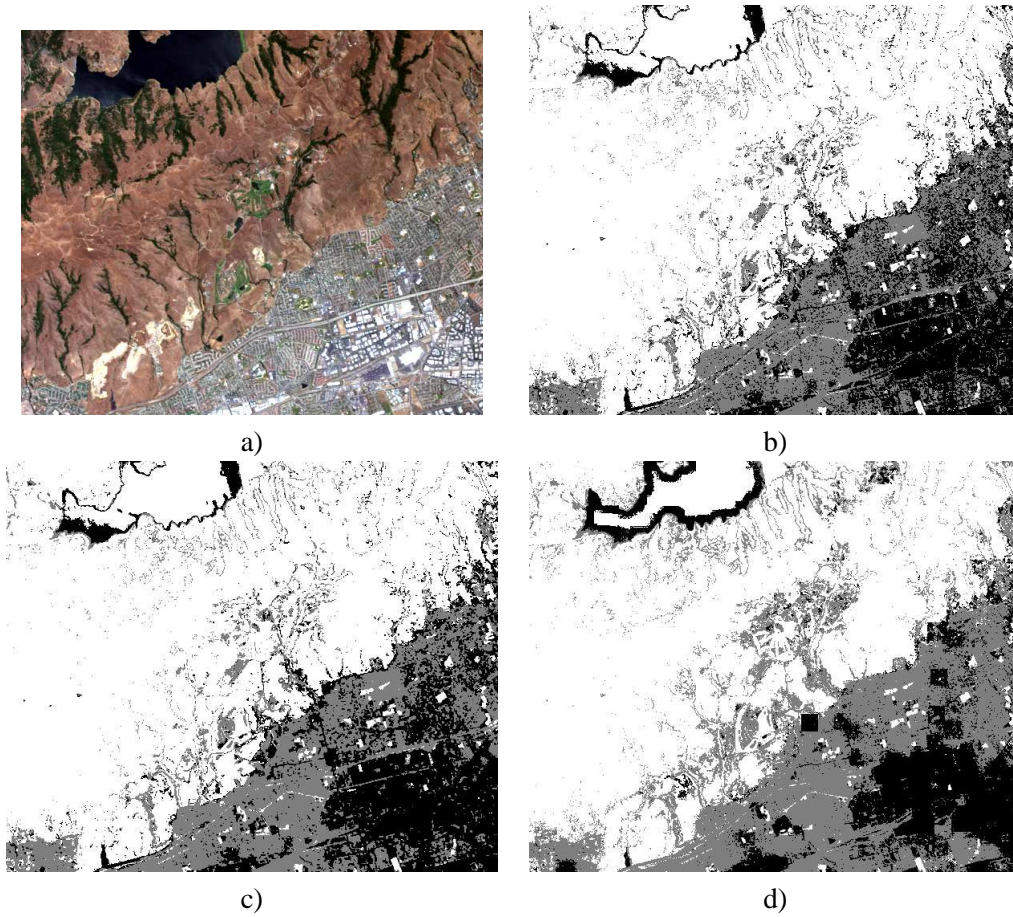


FIG. 2.31 – a) affichage en fausse couleur des canaux  $\{30,20,10\}$  de la scène AVIRIS, b) résultat de la classification par  $PP - 10$ , c)  $PPR - 10$ , d)  $PP - 10,3$ .

### 2.3.2 La segmentation

#### Application à l'imagerie radar : détection des lignes de côte

<sup>12</sup> Cette application concerne l'extraction du trait de côte très utile pour l'étude d'évolutions géographiques ou pour la navigation automatique. Nous considérons des images radar (ERS) qui sont caractérisées par un fort bruit corrélé (bruit de chatoiement). Ce travail montre l'intérêt d'utiliser des interactions autres que celles directement induites des quatre et huit connexités.

Pour extraire la ligne de côte, nous commençons par segmenter l'image à une résolution inférieure. Nous projetons ensuite la segmentation obtenue sur la trame initiale. Des bandes recouvrant les lignes de côtes sont alors obtenues. La ligne de côte est alors extraite par une segmentation binaire de ces bandes. L'approche a été validée sur des images ERS-1 de la Guyanne à une résolution de 12.5 mètres par pixel.

Pour obtenir une extraction fine des lignes côtières, nous avons besoin de la résolution nominale uniquement au voisinage des côtes. De plus, la réduction de la résolution permet de réduire à la fois le bruit et le temps de calcul nécessaire au traitement. Nous moyennons donc les données sur une fenêtre  $4 \times 4$  pour effectuer une première segmentation.

Les approches précédemment proposées pour l'extraction des lignes côtières étaient essentiellement adaptées au cas d'histogrammes bimodaux [LJ90, ZGO94]. Cette hypothèse est néanmoins contestable même à basse résolution. Nous avons donc considéré une segmentation multi-classes pour laquelle nous considérons les quatre classes suivantes : la mer, le continent, les zones sombres correspondant aux réflexions spéculaires des surfaces lisses et les zones claires correspondant aux réflexions diffuses (comme les zones d'érosion côtière par exemple).

Une fonction de coût linéaire par morceaux est définie pour chaque classe. Les paramètres de ces fonctions sont obtenus à partir des moyennes et variances de chacune des classes.

Même à basse résolution, le bruit de chatoiement reste corrélé et induit des segmentations peu satisfaisantes avec l'utilisation d'un modèle de Potts en 4 ou 8 connexités. Nous obtenons en effet, dans notre cas, une sursegmentation contenant un certain nombre de zones érodées d'une cinquantaine de pixels. Pour supprimer ces artefacts, nous utilisons une topologie à trous. Le voisinage correspondant est constitué des pixels associés à la 8-connexité plus huit autres pixels à une distance plus grande (voir figure 2.32(a)). Cette distance doit être plus grande que le diamètre des zones érodées. Un pixel ayant une étiquette différente de ces 8 voisins à distance aura tendance à changer d'étiquette. Dans le cas d'un pixel en bord de côte, la moitié des voisins à distance ont l'étiquette mer et l'autre moitié l'étiquette continent. L'influence des voisins distants est donc nulle et la topologie à trous n'affecte pas la précision des contours.

L'énergie associée à ce système de voisinage s'écrit donc :

$$U(Y) + U(X|Y) = \sum_{c_1=\{i,j\}} \beta_1 \times \delta_{y_i=y_j} + \sum_{c_2=\{i,j\}} \beta_2 \times \delta_{y_i=y_j} + \sum_{c=\{i\}} \sum_l f^l(x_i) \times \delta_{y_i=l}, \quad (2.122)$$

où  $c_1$  représente les cliques de la 8-connexité et  $c_2$  celles contenant des voisins distants, les  $f^l$  étant les fonctions constantes par morceaux définies pour chacune des classes.  $\beta_1$  et  $\beta_2$  sont les paramètres des potentiels de type Potts. L'énergie est minimisée par recuit simulé avec une décroissance géométrique de la température de taux :  $\tau = 0.95$ . Nous obtenons ainsi une segmentation nettoyée des artefacts dus au bruit de chatoiement, comme le montre la figure 2.32(b).

Les résultats obtenus sont comparés à ceux obtenus par un modèle de Potts 8-connexe sur la figure 2.33. La radiométrie des images radar dépend de la réflectivité de la surface imagée. Dès lors, notre

<sup>12</sup> Nous résumons ici des travaux effectués à l'École Nationale Supérieure des Télécommunications en collaboration avec M. Moctezuma. Bibliographie : [20]

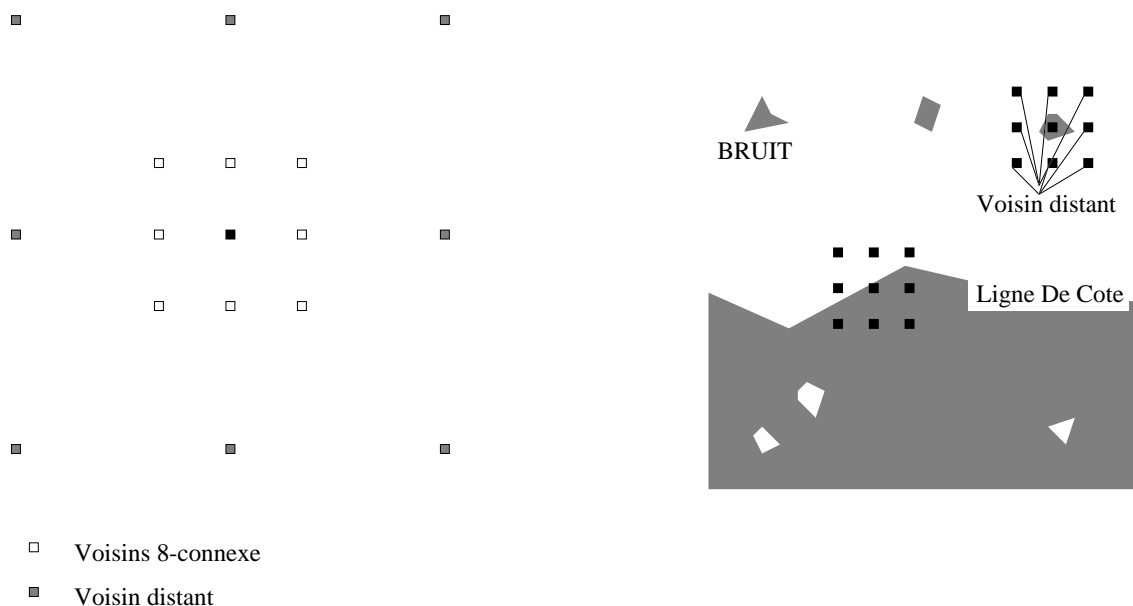


FIG. 2.32 – *La topologie à trous : voisinage (à gauche) et comportement sur les contours (à droite)*

segmentation contient plus de régions que simplement mer et continent. En effet, les réflectivités de la mer, comme du continent, ne sont pas forcément homogènes. Un post-traitement est donc nécessaire pour extraire la ligne de côte parmi les différents contours de l'image segmentée.

Nous projetons les différents contours sur l'image. Nous calculons le gradient orthogonalement à chaque contour. Les segments cotiers sont représentés par les contours dont les gradients sont les plus forts.

Ce résultat, obtenu à une résolution plus basse que celle des données, est ensuite affiné. Pour ce faire, nous projetons la ligne de côte, détectée à basse résolution, sur les données initiales. Nous obtenons ainsi une bande dans laquelle se trouve la ligne de côte à la résolution initiale. Une segmentation est effectuée dans cette bande. Nous avons ici deux classes définies par les bords de la bande. La segmentation est effectuée par un modèle d'Ising. L'attache aux données est définie comme précédemment mais sur les données initiales. Notons que la segmentation à basse résolution nous fournit également des conditions aux bords pour la segmentation finale. Le résultat obtenu est montré sur la figure 2.34.

Nous avons introduit un système de voisinage étendu, appelé topologie à trous, qui permet de s'affranchir des artefacts dus au bruit de chatoiement lors de la segmentation d'images radar. Ce principe a été appliqué à l'extraction des lignes côtières à partir d'images ERS. Nous retrouverons ce système de voisinage pour la segmentation d'image superspectrale.



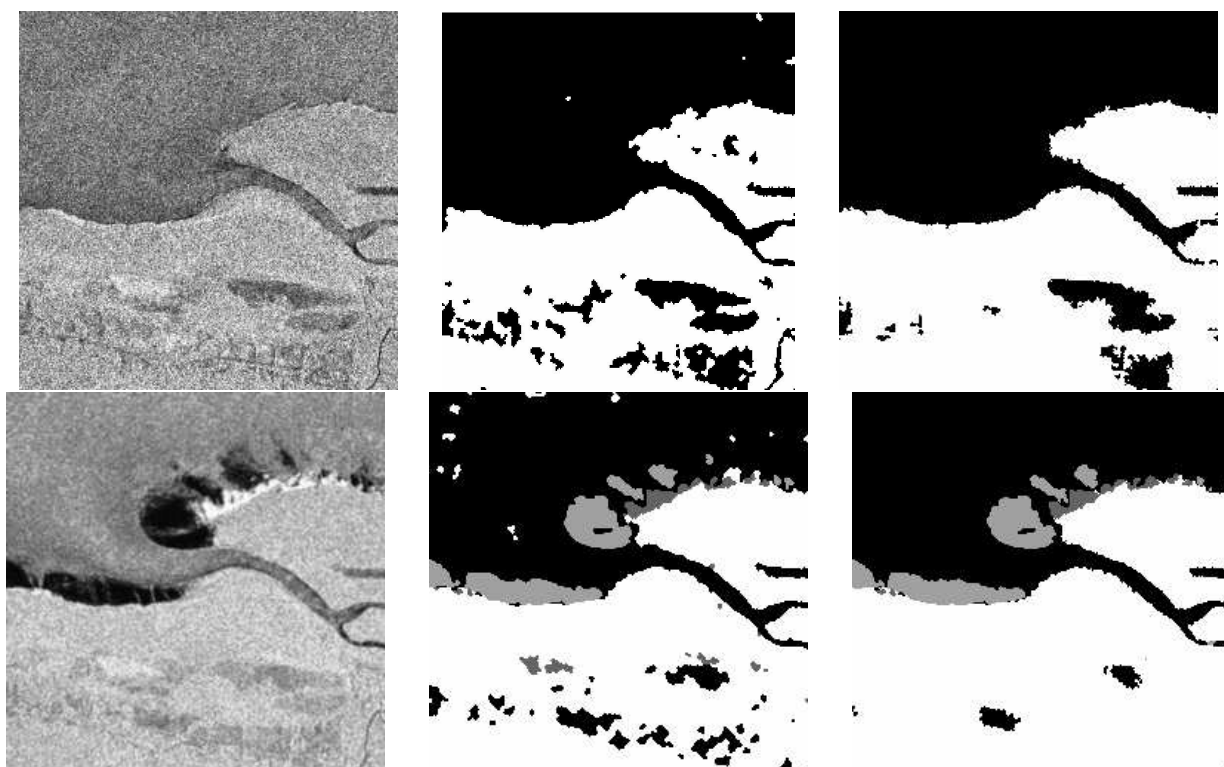
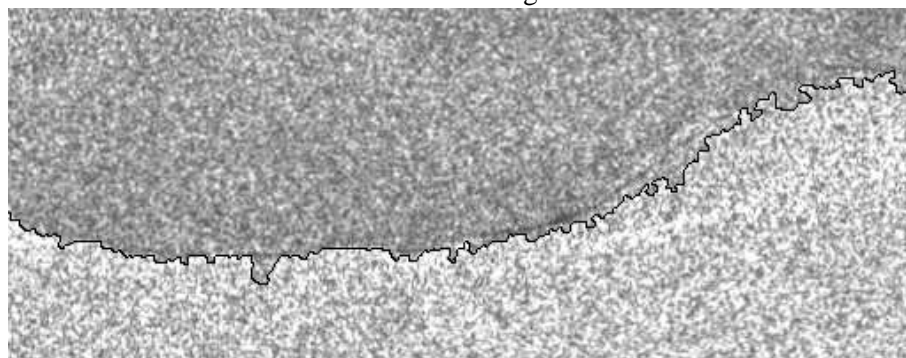


FIG. 2.33 – Images radar ERS de la Guyanne (à gauche), segmentation par modèle de Potts 8-connexe (au milieu) et segmentation avec une topologie à trous (à droite)



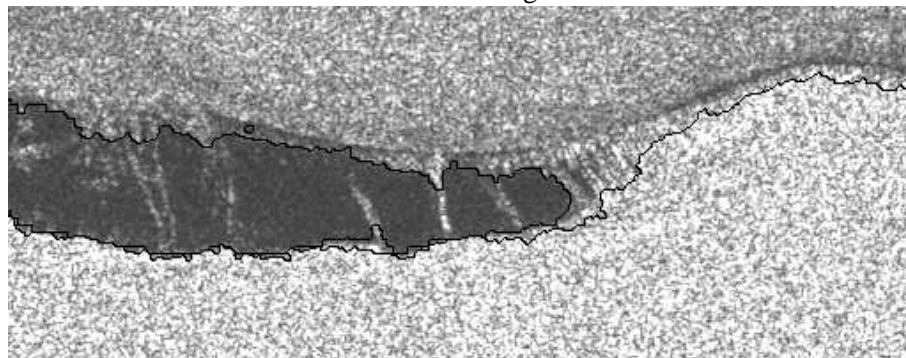
Bande recouvrant la ligne de côte



Ligne de côte extraite



Bande recouvrant la ligne de côte



Ligne de côte extraite

FIG. 2.34 – Lignes de côte extraites à la résolution de l'image initiale.

### Application à l'imagerie hyperspectrale

<sup>13</sup> Nous considérons ici la classification d'images hyperspectrales (i.e. contenant 10 à 15 bandes) avec des modèles vus comme des variations autour du modèle de Potts.

La première image traitée est une image de zone agricole américaine avec une résolution horizontale et verticale de 20 mètres par pixel. Elle comporte 11 bandes, allant du bleu ( $0,44 \mu\text{m}$ ) au proche infra-rouge ( $0,90 \mu\text{m}$ ). Le capteur est un capteur CASI avec 48 bandes de 10 nm, mais l'image est simulée FARMSTAR, qui est un projet de capteur à 11 bandes proposée par Astrium et utilisée pour le "pilotage" agricole.

A gauche de la figure 2.35 se trouve l'image originale, dont trois canaux sur onze ont été sélectionnés pour la coloriser et à droite les régions d'apprentissage.

Les zones oranges sont des champs de blé, avec une distinction entre trois types de blé. A cela s'ajoutent trois types de soja, en vert. La classe route est divisée en trois sous-classes blanches ou grises. Les teintes bleues sont associées à des classes diverses que l'on ne cherche pas à identifier.

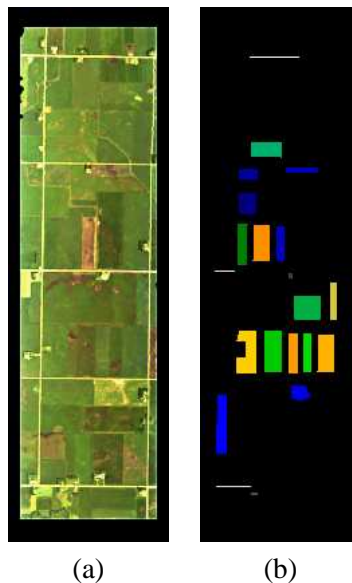


FIG. 2.35 – Classification sur l'image 1 (US) - (a) Image originale - (b) Régions d'apprentissage

La deuxième image représente une zone agricole et forestière proche de Boigneville dans l'Essonne. Elle comporte 10 bandes, du bleu ( $0,44 \mu\text{m}$ ) au proche infra-rouge ( $0,82 \mu\text{m}$ ). Sa résolution est de 10 mètres par pixel. Le capteur est un capteur MIVIS 20 bandes de 20 nm. L'image a été réduite à 10 bandes (voir figure 2.36).

Nous y trouvons diverses cultures : blé, maïs, orge, colza, pois, betteraves. Il existe parfois une distinction entre les cultures d'hiver et celles de printemps. Deux classes ont aussi été réservées pour les terres en jachère et la forêt. Les régions d'apprentissage sont ici très grandes. Même si, en pratique, l'apprentissage n'est réalisé que sur une petite portion de l'image, nous disposons ici d'une grande quantité de données sur le terrain que nous exploiterons lors de l'estimation des erreurs de classification.

**Régularisation : Topologie et géométrie-** Nous étudions l'influence du système de voisinage, ainsi que celle du paramètre de régularisation avec un modèle de Potts. Pour effectuer des tests comparables, nous utiliserons le même algorithme dans toute cette partie : l'algorithme ICM [Bes74].

13. Nous résumons ici des travaux du stage de DEA d'O. Pony financé par Astrium. Bibliographie : [62]

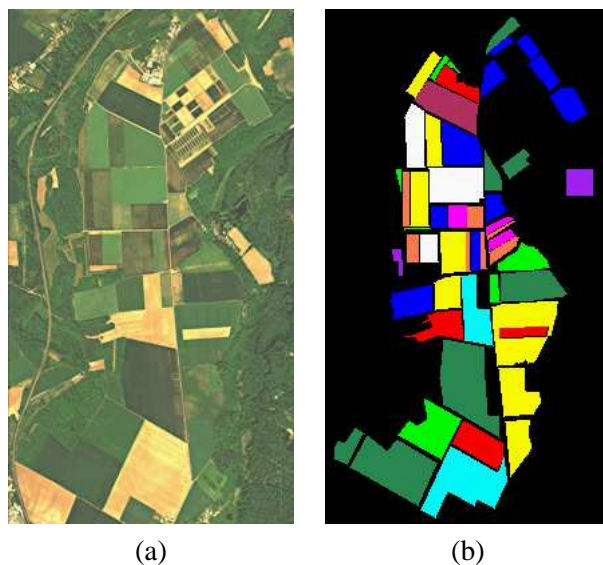


FIG. 2.36 – Classification sur l'image 2 (environs de Boigneville) - (a) Image originale - (b) Régions d'apprentissage

Nous fixons  $\beta$  à 1. Dans le tableau de résultats 2.37, nous recueillons le nombre d'itérations nécessaires pour converger, et le taux d'erreur sur la vérité terrain. Les taux d'erreur sont calculés sur les régions d'apprentissage pour l'image US et sur la vérité terrain (régions plus grandes que celles utilisées pour l'apprentissage) pour l'image de Boigneville.

	Voisinage	Nb iter	Erreur (%)
US	4-connexité	3	0,40
	8-connexité	5	0,35
Boigneville	4-connexité	4	7,00
	8-connexité	8	6,50

FIG. 2.37 – Comparaison des voisinages en 4- et 8-connexité

Nous pourrions nous satisfaire d'une de ces deux topologies. Cependant, en regardant l'image des erreurs commises, nous observons des petites taches de pixels mal classées. Il est possible d'évacuer, en partie, ces structures à l'aide d'une topologie dite à trous.

Nous utilisons maintenant la topologie à trous décrite dans le paragraphe 2.3.2. Une topologie à trous consiste à ajouter au système de voisinage en 4- ou 8-connexité des éléments plus éloignés. Nous choisissons une topologie classique de base, en 4-connexité par exemple, et une longueur de saut, 5 par exemple. Nous ajoutons au voisinage d'un pixel les éléments qui se situent 5 pixels plus loin dans les 4 directions des 4 pixels voisins d'origine (voir figure 2.38).

Ainsi, les pixels dans une tache jugée mal classée auront plus de chance d'être reclassés correctement si nous ajoutons des pixels un peu éloignés à leur voisinage. Les résultats sont présentés dans le tableau 2.39, à comparer avec le tableau 2.37. Pour que les résultats soient comparables, il a fallu fixer  $\beta = 0,5$ , puisque nous avons doublé la taille du voisinage de chaque pixel, donc doublé le nombre de cliques de l'image.

Le tableau 2.40 montre le pourcentage d'erreur pour différentes valeurs du paramètre de régularisation. Dans toute cette partie, nous choisissons une topologie en 8-connexité à trous.

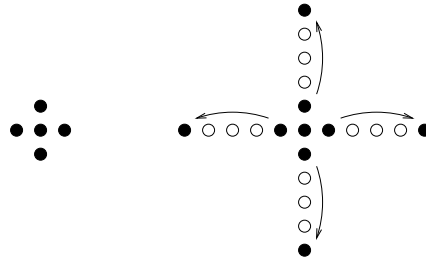


FIG. 2.38 – Topologie en 4-connectivité classique et à trous

	Voisinage	Nb iter	Erreur (%)
US	4 à trous	4	0,45
	8 à trous	5	0,30
Boigneville	4 à trous	6	6,81
	8 à trous	17	5,97

FIG. 2.39 – Comparaison des voisinages dans le cas de la topologie à trous

Dans l'image 1 (US), nous obtenons des résultats presque parfaits pour une valeur de 5.

Dans l'image 2 (Boigneville), les meilleurs résultats sont obtenus pour  $\beta$  entre 5 et 10. Mais en observant les images classifiées pour les valeurs de  $\beta$  entre 3 et 5, nous réalisons que, en dehors des zones d'apprentissage, une régularisation trop forte détruit certaines petites parcelles (voir figure 2.41). Nous préférons donc des valeurs inférieures à 3. Comme nous l'avons déjà signalé, le modèle de Potts atteint ses limites en présence de structures fines. Le chien-modèle est une alternative. Dans le cas présent, une seule classe, la classe route, présente des structures fines. Nous optons donc pour un modèle de Potts avec des paramètres adaptatifs suivant la classe considérée.

	$\beta$	Nb iter	Erreur (%)
US	1	11	0,20
	3	24	0,10
	5	27	0,02
	10	33	0,05
	100	25	0,94
Boigneville	1	19	5,19
	3	23	4,18
	5	40	3,92
	10	30	3,90
	100	37	6,79

FIG. 2.40 – Comparaison pour différentes valeurs de  $\beta$  en 8-connectivité à trous

**Des paramètres de régularisation adaptatifs-** Dans l'image de Boigneville, nous avons rajouté une classe "route" pour tenter de détecter une route qui traverse la forêt. Or, la régularisation telle que nous l'avons décrite, qui s'appuie sur l'observation du voisinage de chaque pixel, risque de détruire la classe "route". En effet, à notre échelle, l'épaisseur de la route est de 1 pixel, et un pixel de route est davantage entouré de pixels de la classe "forêt" que de pixels de la classe "route". Il se peut donc que le terme de régularisation favorise la classe "forêt".

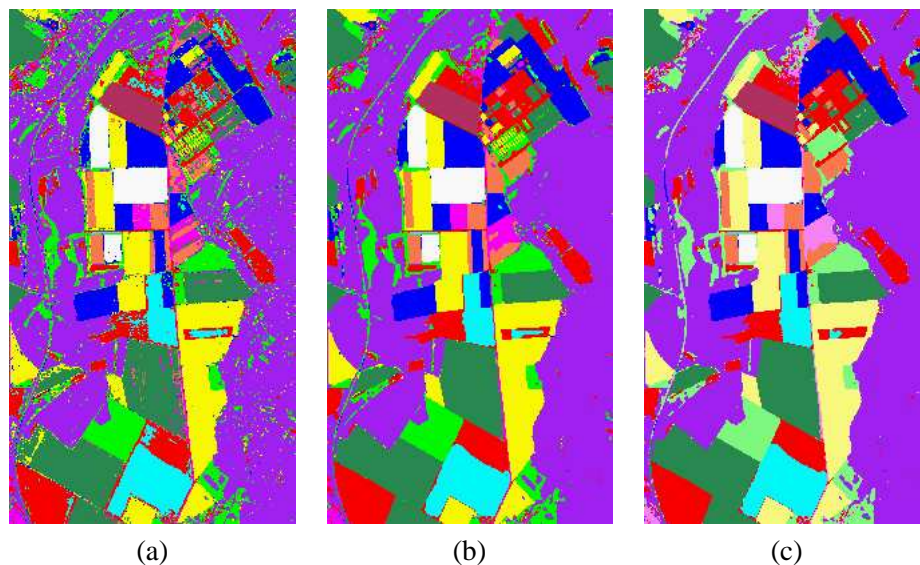


FIG. 2.41 – Image classifiée brute (a), et images régularisées par ICM et modèle de Potts avec une topologie en 8-connexité à trous, pour  $\beta=3$  (b) et  $\beta=5$  (c)

Cependant, dans nos exemples, les structures fines telles que les routes ont une réponse spectrale suffisamment différente de celles des paysages agricoles et forestiers environnants, ce qui permet une bonne détection malgré le problème évoqué précédemment. Par contre, nous avons vu que le taux d'erreur résiduelle sur l'image de Boigneville correspond à la confusion entre le blé et le blé d'hiver (voir la figure 2.42). Il est possible de rendre la classe blé plus "forte" lors de la régularisation. Pour y parvenir, nous modifions le modèle de Potts pour qu'une clique d'ordre 2 de type "blé-blé" ait une énergie plus faible qu'une clique de type "blé d'hiver-blé d'hiver".



FIG. 2.42 – Image de confusion après régularisation

Pour chaque classe  $\lambda$ , nous fixons un paramètre  $\beta_\lambda$  qui vaut 1 par défaut, et qui est d'autant plus

grand que nous souhaitons que la classe soit robuste. Le terme de régularisation est ainsi modifié :

$$\beta \times \sum_{(s,t) \in C_2} \beta_{x_s} \beta_{x_t} \delta_{x_s, x_t}$$

Nous avons appliqué à l'image de Boigneville une régularisation très forte ( $\beta = 20$ ) par recuit simulé pour un voisinage en 8-connexité. Les deux topologies classique et "à trous" ont été testées. Nous avons isolé la classe "route" qui traverse la forêt sur la gauche de l'image, et étudié sa résistance à la régularisation entre le modèle de Potts et le modèle avec paramètres adaptatifs (voir figure 2.43).

La topologie à trous affaiblit davantage ces structures linéiques que la topologie classique, comme nous pouvions nous y attendre. Les paramètres  $\beta_\lambda$  adaptatifs, fixés à 1 pour l'ensemble des classes et à 4 pour la classe "route", améliorent effectivement la détection de la route. Par contre, le paramétrage doit être fait avec précaution, nous voyons pour  $\beta_{route} = 5$  que la classe "route" commence déjà à déborder sur les classes voisines.

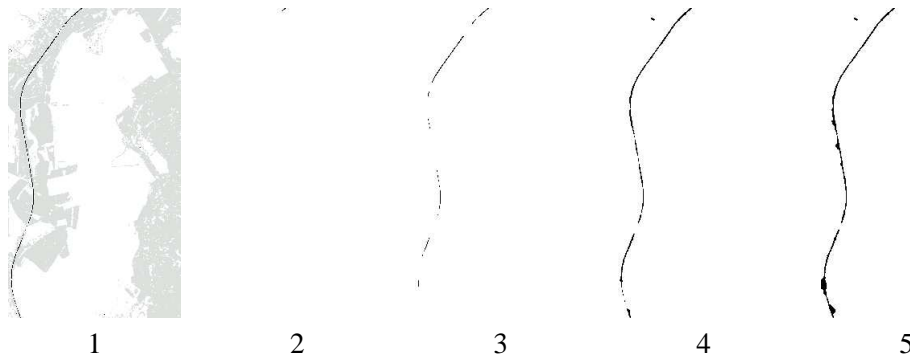


FIG. 2.43 – Résistance de la classe route à la régularisation ( $\beta = 20$ , recuit) : 1. classification sans régularisation ; 2. modèle de Potts classique et topologie à trous ; 3. modèle de Potts classique et topologie classique ; 4.  $\beta_{route} = 4$  et topologie classique ; 5.  $\beta_{route} = 5$  et topologie classique

Revenons au problème de la confusion entre "blé" et "blé d'hiver". Nous avons donné à la classe "blé" plus de force que la classe "blé d'hiver" par un jeu de paramètres. Nous avons aussi avantagé le "maïs" par rapport aux "betteraves", qui avaient tendance à en recouvrir certaines parcelles. Ces démarches sont artificielles, elles ne traduisent aucune réalité géométrique sur les parcelles des classes considérées. Elles ont éventuellement une justification qu'il faut chercher dans la forme des clusters dans l'espace des états. Nous obtenons toutefois des résultats satisfaisants.

Auparavant, nous avions un taux d'erreur d'environ 4% au mieux. Avec un paramétrage adaptatif, il se réduit à 2%. Nous pouvons comparer la nouvelle image des erreurs de classification (figure 2.44) avec celle que nous obtenions pour un modèle de Potts classique (figure 2.42). Nous voyons bien que certains problèmes de confusion de classes ont été résolus.

Même si ce modèle améliore la classification, il introduit autant de nouveaux paramètres que de classes définies à l'apprentissage. L'intérêt de cette méthode serait assuré si nous pouvions fixer ces paramètres de manière automatique. Dans un premier temps, nous proposons une méthode pour fixer ces paramètres de manière grossière.

La classification d'un pixel lors de la régularisation se joue sur la différence entre les termes de régularisation de l'énergie pour deux configurations différentes. Cette grandeur dépend du nombre de pixels d'une certaine classe  $\lambda$  dans le voisinage, multiplié par le paramètre de cette même classe  $\beta_\lambda$ . Nous voudrions donc que, pour toutes les classes, ce produit soit du même ordre de grandeur.

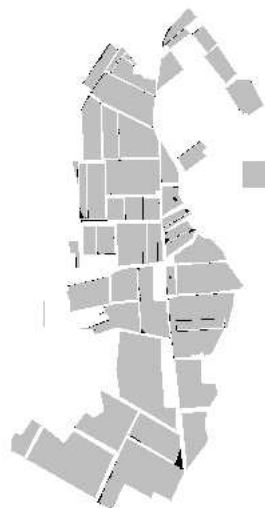


FIG. 2.44 – Image de confusion après régularisation, avec un paramétrage adaptatif

Au début de la phase de régularisation, nous disposons d'une image classifiée par maximum de vraisemblance. L'observation visuelle permet de se faire une idée des structures des classes. Nous percevons distinctement les formes linéiques, telles que les routes, et les grandes parcelles, telles que les cultures. Typiquement, un pixel de route possède deux pixels de la même classe dans son voisinage. Quant à un pixel de culture, il possède habituellement huit pixels de la même classe.

Ainsi, tel que nous l'avons proposé dans un exemple précédent, le paramètre  $\beta_{route}$  devrait être environ 4 fois supérieur au paramètre  $\beta_{champ}$ . Une première méthode pour estimer ces paramètres consiste à étudier sur l'image classifiée brute, pour chaque classe, le nombre moyen, pour les pixels de cette classe, de voisins de même type. Le nombre obtenu est compris entre 0 et 8. 0 correspond à une classe totalement incohérente et éparpillée, et 8, à une classe qui s'étend sur de vastes surfaces uniformes.

Le paramètre  $\beta_\lambda$  est fixé comme l'inverse du nombre obtenu par cette étude de voisinage, à un facteur multiplicatif près, 8 dans nos applications.

Même si cette méthode fonctionne bien en théorie pour les structures linéiques et planaires, nous sommes confrontés, en pratique, à certains problèmes. La classification brute est bruitée, et il se peut que des pixels de deux classes ambiguës se mélangent et faussent l'étude de la structure de la classe. Nous pouvons toujours corriger ce problème en n'estimant les paramètres  $\beta_\lambda$  qu'après quelques étapes de régularisation, qui auront un effet de débruitage sur l'image.

**Un critère de texture-** Nous proposons un modèle isotrope de texture qui se fonde sur le calcul de la variance locale. La figure 2.45 montre une coupe d'une bande de l'image dans le proche infra-rouge. Nous pouvons voir que la variance en niveaux de gris dans la forêt à droite est plus forte que sur les champs du milieu. De même, les pixels au bord des champs ont aussi une forte variance locale et donc répondent fortement, ce qui gênera notre classification.

Nous allons extraire un paramètre qui caractérise le bord d'un champ. Sur la coupe de la figure 2.45, nous voyons que les bords des champs correspondent à des sauts en niveaux de gris, auxquels s'ajoute un signal dont la variance est négligeable par rapport à la hauteur du saut.

Dans une fenêtre  $13 \times 13$  autour du pixel, nous choisissons un pas  $p$ , qui correspond à la taille d'une grille que l'on plaque sur l'image. Nous imposons que  $p$  divise 12, et nous calculons une variation totale au carré  $\Psi$  sur la fenêtre. Si la fenêtre est centrée en  $(x, y)$ , on a :





FIG. 2.45 – Coupe d'une bande du proche infra-rouge de l'image de Boigneville

$$V_p = \sum_{i=-p}^{p-1} \sum_{j=-p}^{p-1} (I(x+p(i+1), y+pj) - I(x+pi, y+pj))^2 + (I(x+pi, y+p(j+1)) - I(x+pi, y+pj))^2$$

La figure 2.46 illustre cette équation.

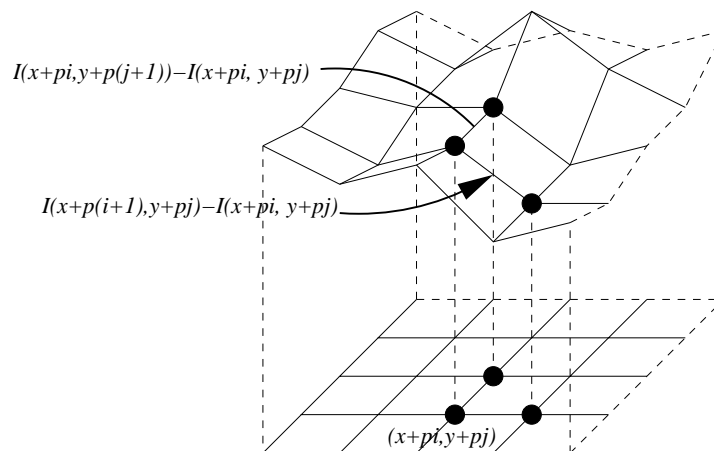


FIG. 2.46 – Illustration du calcul de variation totale sur une fenêtre

On réalise une telle mesure pour toutes les valeurs de  $p$  qui divisent 12 : 1, 2, 3, 4, 6, 12. On estime ensuite un paramètre  $d$ , proche de la notion de dimension de Hausdorff, qui vérifie  $V_p \cdot p^{d-2} = cte$ . On en déduit :  $(d-2) \log(p) + \log(V_p) = cte$ .

On estime donc  $d$  ainsi :

$$d - 2 = \frac{d \log(V_p)}{d \log(p)}$$

Pour une surface plane,  $d = 2$ , et pour une surface bruitée,  $2 < d < 3$ . Un saut tel qu'un bord de champ aura une réponse de type surface plane, alors qu'une micro-texture aura une réponse plus forte, notamment pour les forêts et les champs. En multipliant le paramètre  $(d - 2)$  par la variance locale, on obtient un nouveau paramètre qui sélectionne les zones à forte variance, et évacue les bords des champs.

L'image 2.47 montre le résultat de ce traitement pour une bande dans le proche infra-rouge issue de l'image de Boigneville, après avoir appliqué un filtre médian. Ce filtre s'avère être utile pour rendre l'image plus lisse, et on constate expérimentalement que la classification fonctionne mieux.



FIG. 2.47 – Image de texture d'une bande dans le proche infra-rouge issue de l'image de Boigneville

**Résultats pour un plan d'occupation des sols-** Dans une optique de plan d'occupation des sols, la classification consiste à séparer par exemple la ville, la campagne, la forêt, la mer et la montagne. Nous disposons d'une image qui comporte à la fois de la ville, de la forêt et de la campagne.

C'est une image de la ville de Boigneville, qui prolonge l'image de la campagne que nous avons étudiée jusqu'à présent. Elle comporte 20 bandes, du bleu ( $0,44 \mu m$ ) au proche infra-rouge ( $0,82 \mu m$ ). Sa résolution est de 2,5 mètres par pixel, mais nous avons d'abord simulé une image à 10 mètres par pixel. Le capteur est un capteur MIVIS 20 bandes de 20 nm.

La figure 2.48 représente la scène et les régions d'apprentissage pour la ville, la forêt et la campagne.

A cette échelle, nous considérons la ville comme une microtexture, car nous distinguons à peine les objets. Ceci nous permet d'utiliser les modèles de texture présentés précédemment. Nous pouvons concevoir que l'on ne peut pas segmenter la ville du reste de l'image autrement qu'en extrayant un paramètre de texture. La réponse radiométrique de la ville est totalement chaotique et ne constitue pas un caractère suffisamment discriminant pour bien la séparer des autres classes. Une simple étude spectrale pourrait permettre de mieux séparer la forêt de la campagne mais des erreurs seraient commises.

La figure 2.49 exhibe deux images du critère de texture fondé sur la variance locale appliquée sur deux bandes de l'image : l'une dans le spectre visible, l'autre dans le proche infra-rouge. Ces images montrent que les critères de texture sur une bande du spectre visible permettent de différencier la ville de la forêt et la campagne. Et la texture sur une bande dans le proche infra-rouge permet de discriminer la campagne

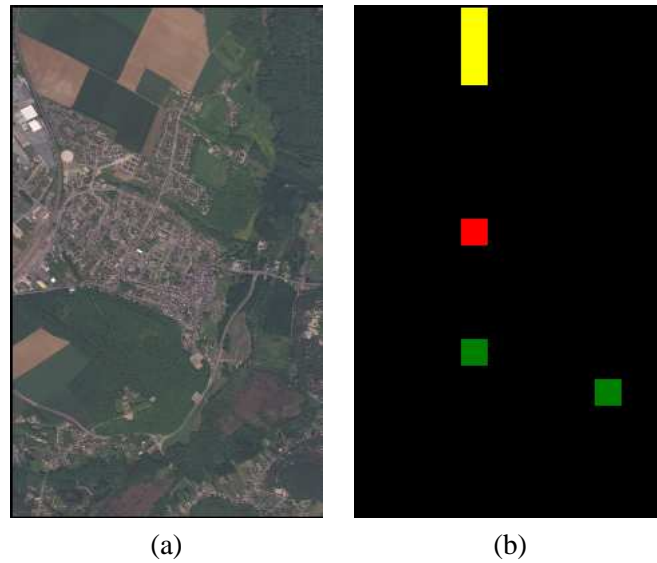


FIG. 2.48 – Image 3 : ville de Boigneville (a), et régions d'apprentissage : ville, campagne, forêt (b)

de la ville et la forêt. A priori, nous avons donc deux critères suffisants pour réaliser une classification de type plan d'occupation des sols.

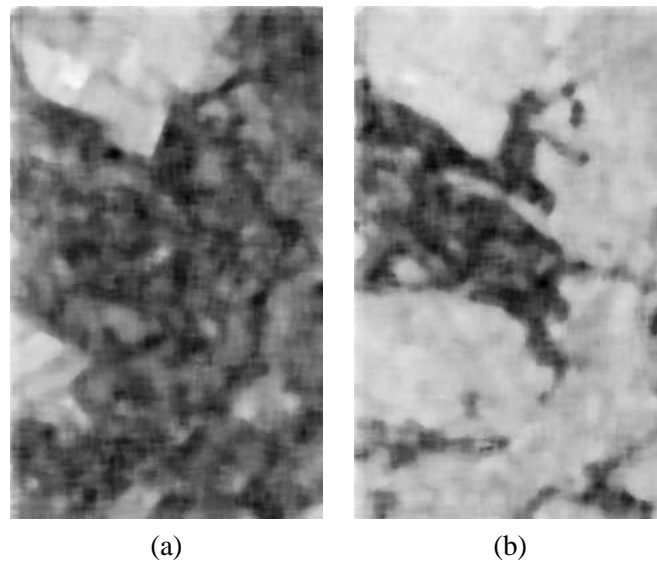


FIG. 2.49 – Images du critère de texture sur deux bandes : (a) proche infra-rouge, (b) bleu

La classification est effectuée en ajoutant aux bandes de l'image nos deux bandes de texture. Nous comparons 4 images :

- a) 20 bandes radiométriques
- b) 20 bandes radiométriques + 2 de texture
- c) 2 bandes de texture
- d) 3 bandes radiométriques + 2 de texture

Les images sont regroupées en figure 2.50.

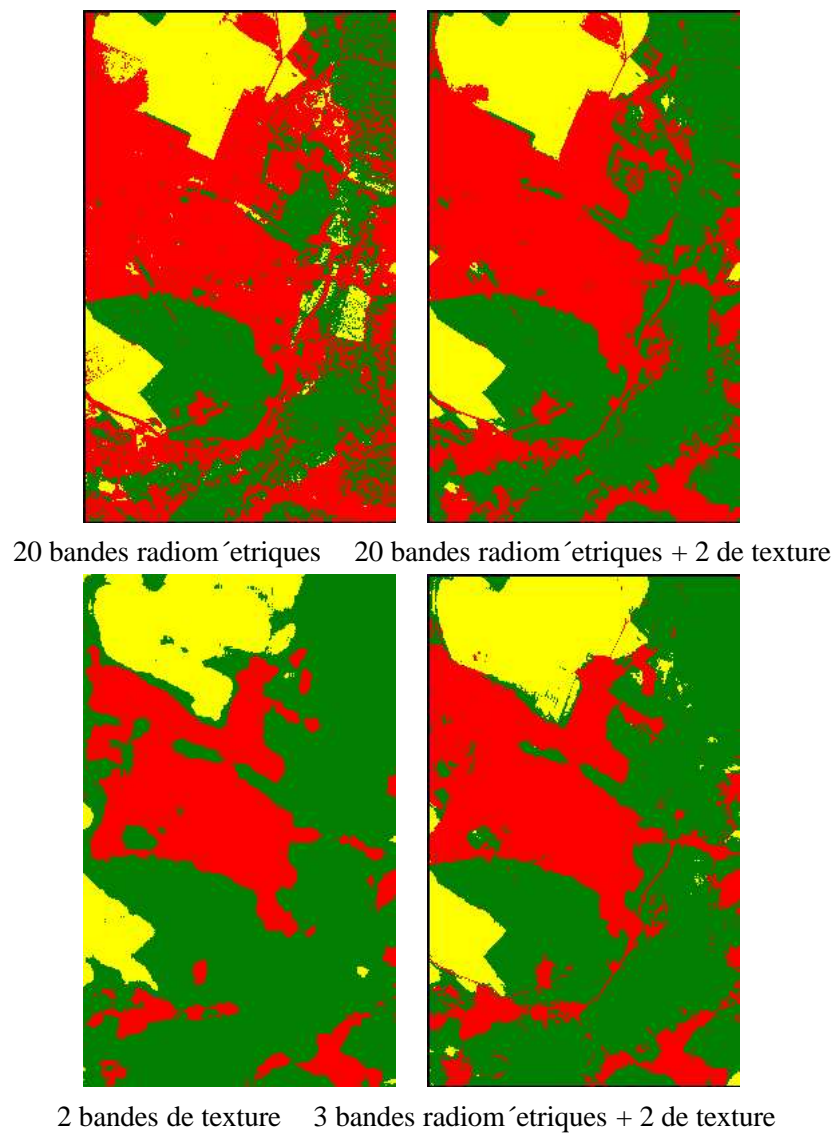


FIG. 2.50 – Classification d'images qui combinent information radiométrique et critères de texture

Les images montrent que la radiométrie seule ou la texture seule ne permettent pas de réaliser une bonne classification. Il faut manifestement mélanger ces deux types d'information. Cependant, cela ne suffit pas, nous constatons qu'entre les deux images mixtes, nous avons un meilleur résultat en retirant quelques bandes radiométriques.

Sur l'image de campagne proche de Boigneville, nous avons réalisé une classification avec des bandes de texture. Nous comparons les résultats dans le tableau 2.51. Remarquons que ce tableau n'indique pas l'omission de la route. Elle est en effet noyée dans une texture de forêt. Par contre, nous avons évoqué le problème des parcelles ambiguës de "blé" et "blé d'hiver". La texture résout ce problème partiellement, les paramètres de régularisation adaptatifs améliorent encore la classification.

	Erreur (%)
classification 10 bandes	7,64
classification 10 bandes +ICM ( $\beta = 2$ )	4,32
classification 10 bandes +recuit simulé ( $\beta = 2$ )	4,00
classification 10 bandes +texture bandes 1 et 10 : variance	6,39
classification 10 bandes +texture bandes 1, 7, 10 : variance	6,27
classification 10 bandes +texture bandes 1, 7, 10 : Gabor	6,49
classification 10 bandes +texture bandes 1, 7, 10 : variance et Gabor	4,77
classification précédente +recuit ( $\beta = 2$ )	2,75
classification précédente +recuit ( $\beta = 2$ ) + $\beta$ adaptatifs	1,42

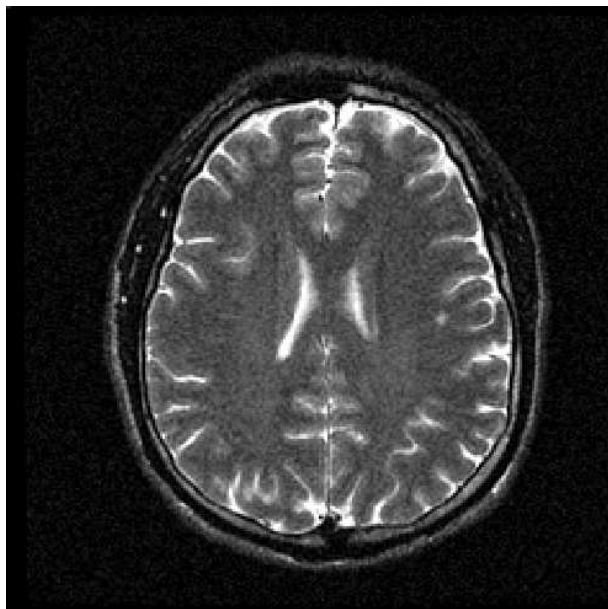
FIG. 2.51 – *Qualité de la classification sur l'image de campagne proche de Boigneville*

### Application à l'Imagerie par Résonance Magnétique (IRM)

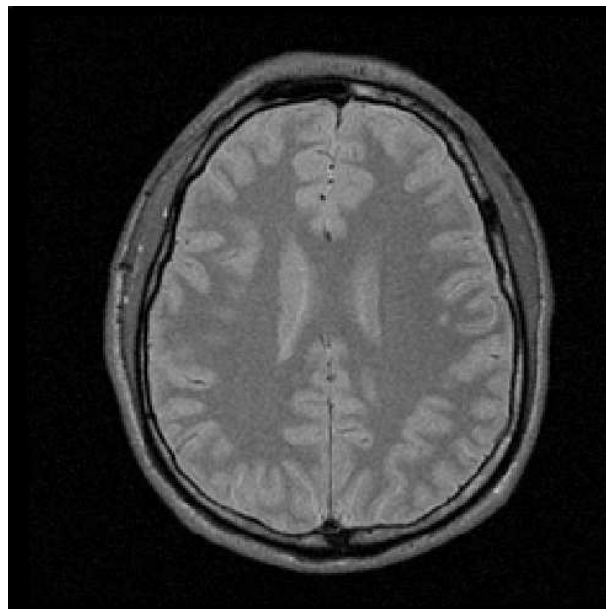
<sup>14</sup> Nous présentons ici une approche automatique de segmentation appliquée aux IRM cérébrales. Nous avons plusieurs échos 3D du même sujet. L'approche consiste à classifier automatiquement chacun des échos, en utilisant l'algorithme HVC présenté au paragraphe 2.2.1. Les différentes classifications sont ensuite fusionnées à partir du principe MDL ("Minimum Description Length"), puis une étape de régularisation utilise une extension 3D du Chien-modèle. Pour finir, nous présentons un second algorithme permettant de corriger les non-stationarités des images. Cet algorithme est fondé sur une extension du processus ligne proposée par Geman et Geman [GG84] suivi d'une estimation des non-stationarités. Les tests sont conduits sur deux images montrées sur les figures 2.52 et 2.53.

**Fusion de plusieurs classifications par MDL-** Nous considérons plusieurs échos d'une même scène préalablement classifiés par l'algorithme HVC décrit au paragraphe 2.2.1. Une classification initiale nous est donnée par les différentes intersections des classes obtenues sur chacun des échos. Cette sur-segmentation est affinée en fusionnant certaines classes par un principe de MDL. Ce critère permet

14. Nous résumons ici des travaux effectués à l'Université Catholique de Louvain (KUL). Bibliographie : [74]

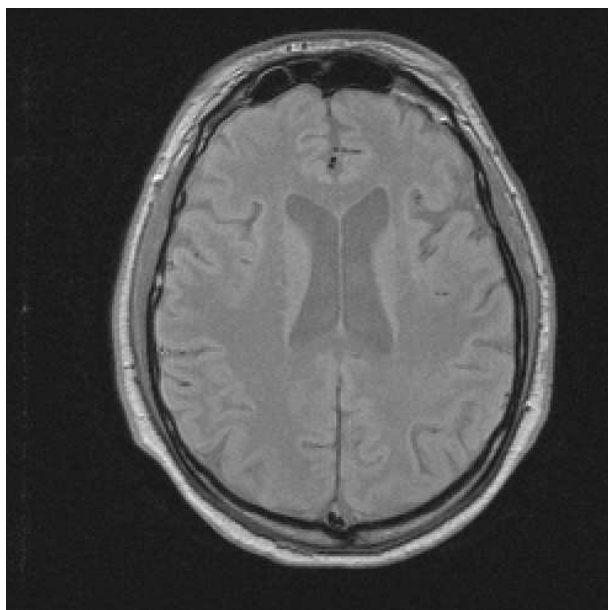


Echo 1

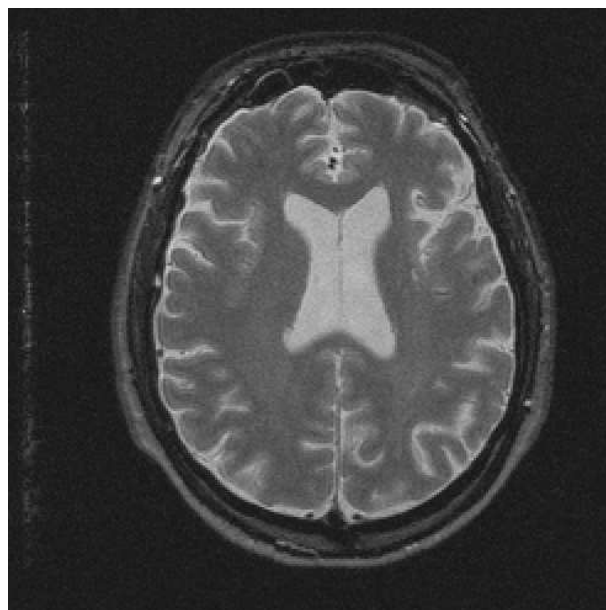


Echo 2

FIG. 2.52 – Coupe 2D d'une IRM de bonne qualité



Echo 1



Echo 2

FIG. 2.53 – Coupe 2D d'une IRM de qualité moyenne

de trouver un compromis entre la complexité du modèle et sa précision pour représenter les données. La précision du modèle est donnée par la log-vraisemblance et sa complexité par le nombre de ses paramètres.

Considérons  $n$  échos ou capteurs notés  $X_{sp} = (x_s^{cap})$ . Une classe  $c_k$  est définie par la probabilité a priori  $P(c_k)$  (proportionnelle au nombre de pixels de cette classe dans le cas présent) et une Gaussienne multivariée  $p_k$  caractérisée par un vecteur de moyenne et une matrice de covariance. La vraisemblance s'écrit alors :

$$L = \prod_{s \in S} \sum_{k=1}^K P(c_k) p_k(x_s^1, \dots, x_s^{cap}). \quad (2.123)$$

Le critère MDL est alors défini par [LMH94]:

$$MDL(K) = -\ln(L) + \lambda J \ln(\text{card}(S)), \quad (2.124)$$

où  $J = 10K - 1$  est le nombre de paramètres libres.

Considérons une classification à  $c$  classes et examinons la fusion de deux classes. Le second terme de l'équation (2.124) décroît car  $J$  décroît mais le premier terme représentant la vraisemblance croît. C'est donc un compromis que nous recherchons.

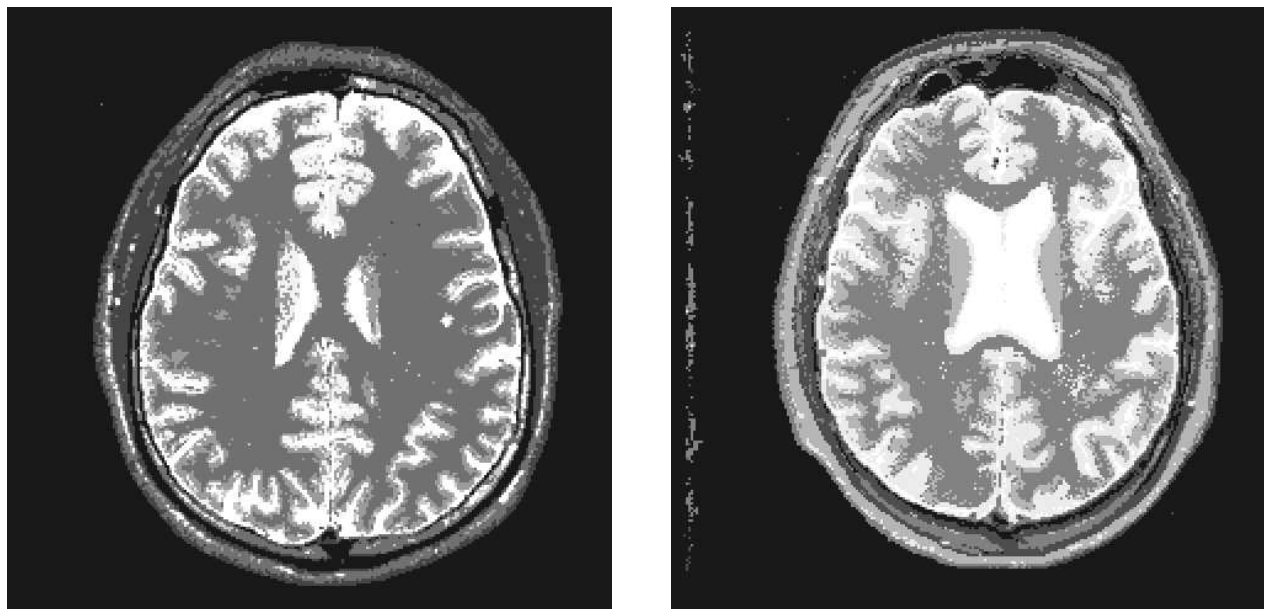
L'initialisation considérée est donc constituée de l'intersection des différentes classes obtenues sur chacun des échos. Nous considérons la fusion des classes  $c_i$  et  $c_j$ . Cette fusion est acceptée si elle permet de faire décroître le critère MDL. La classe candidate  $\epsilon$  est celle ayant la probabilité a priori  $P(\epsilon)$  la plus faible (le plus faible nombre de pixels).  $c_j$  est alors la classe la plus proche de  $c_i$  au sens de la distance de Mahalanobis. Si la fusion est rejetée, la classe  $\epsilon$  est inhibée et la classe de plus faible probabilité a priori parmi les classes non-inhibées est sélectionnée. Si la fusion est acceptée, les classes inhibées sont ré-activées. Le processus s'arrête lorsque toutes les classes sont inhibées.

**Résultats sur des IRM-** Nous considérons d'abord les deux échos de la figure 2.52. La figure 2.54.a montre les résultats obtenus à partir de la fusion MDL sur les deux classifications par HVC. Nous obtenons 4 classes sur le premier écho et 6 classes sur le second. La fusion aboutit à une classification à 6 classes. Ces classes correspondent à l'os, la peau, la matière blanche, la matière grise et le LCR (Liquide Céphalo-Rachidien). La sixième classe est un mélange matière blanche-matière grise qui peut être interprétée comme du volume partiel. Cette classification automatique est donc cohérente avec l'interprétation sémantique que l'on peut faire de l'image et fait ressortir l'information présente dans les deux échos.

La deuxième image (figure 2.53) est plus bruitée que la première. Un des échos a un faible contraste et produit une classification de mauvaise qualité. Néanmoins, la fusion montrée sur la figure 2.54.b, n'est pas gênée par ce capteur "polluant". La variance du noyau de convolution utilisée dans l'algorithme HVC est ici égale à 20 (10 dans le cas précédent) en raison du bruit plus important de l'image. Les nombres de classes obtenues sont de 6 et 7 sur les deux capteurs alors que la fusion produit 6 classes comme dans le cas précédent.

Les fusions obtenues sont satisfaisantes et correspondent à ce qu'un opérateur aurait pu faire à partir des classifications initiales. Il reste néanmoins un paramètre ( $\lambda$ ) à régler, qui permet de pondérer la vraisemblance et la complexité du modèle dans la définition du critère MDL. Notons que la valeur utilisée ( $\lambda = 400$ ) a été la même dans les deux cas, mais ne correspond pas à celle déduite de la théorie de l'information [LJC92].

**Extension du chien-modèle à un modèle 3D-** Nous abordons maintenant la régularisation des classifications obtenues précédemment. Rappelons qu'il s'agit de volume et non d'images 2D. En outre, ces images contiennent des structures fines (interface matière grise-matière blanche) qu'il convient de préserver. Nous généralisons donc le chien-modèle au cas 3D. Une généralisation naturelle consisterait



a) A partir de l'image de bonne qualité

b) A partir de l'image de qualité moyenne

FIG. 2.54 – Classification par fusion MDL

à considérer des cliques de tailles  $3 \times 3 \times 3$ , ce qui induit un voisinage de 124 voxels. Un tel modèle nécessite un temps de calcul considérable. En outre, il nous faudrait définir les différents objets par des volumes, des surfaces, des lignes ou du bruit, en considérant les différentes symétries. Nous avons opté pour une approche plus réaliste en définissant des modèles 2D couplés. Sur chacun des plans principaux  $X = Constante$ ,  $Y = Constante$  ou  $Z = Constante$ , nous considérons un chien-modèle 2D. Un voxel donnée a donc 60 voisins (cf figure 2.55), et le temps de calcul est multiplié par trois relativement à un modèle 2D. La pénalisation induite sur les différents objets est décrite sur la figure 2.56.

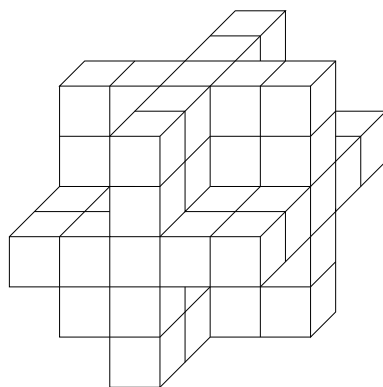


FIG. 2.55 – Voisinage induit par le chien-modèle 3D

**Résultat sur des IRM-** Nous comparons ici les segmentations obtenues par le modèle de Potts et par le chien-modèle. Pour éviter de détruire les structures fines par le modèle de Potts, nous minimisons volontairement l'effet du modèle a priori. Lorsque le bruit est faible, nous pouvons effectivement diminuer le poids du modèle a priori. Dans ce cas, les différences entre le chien-modèle et le modèle de Potts sont minimales (voir la figure 2.57). Sur l'image de bonne qualité, nous pouvons uniquement remarquer que le LCR est mieux connecté avec le chien-modèle. Lorsque le bruit est plus important, nous devons accroître



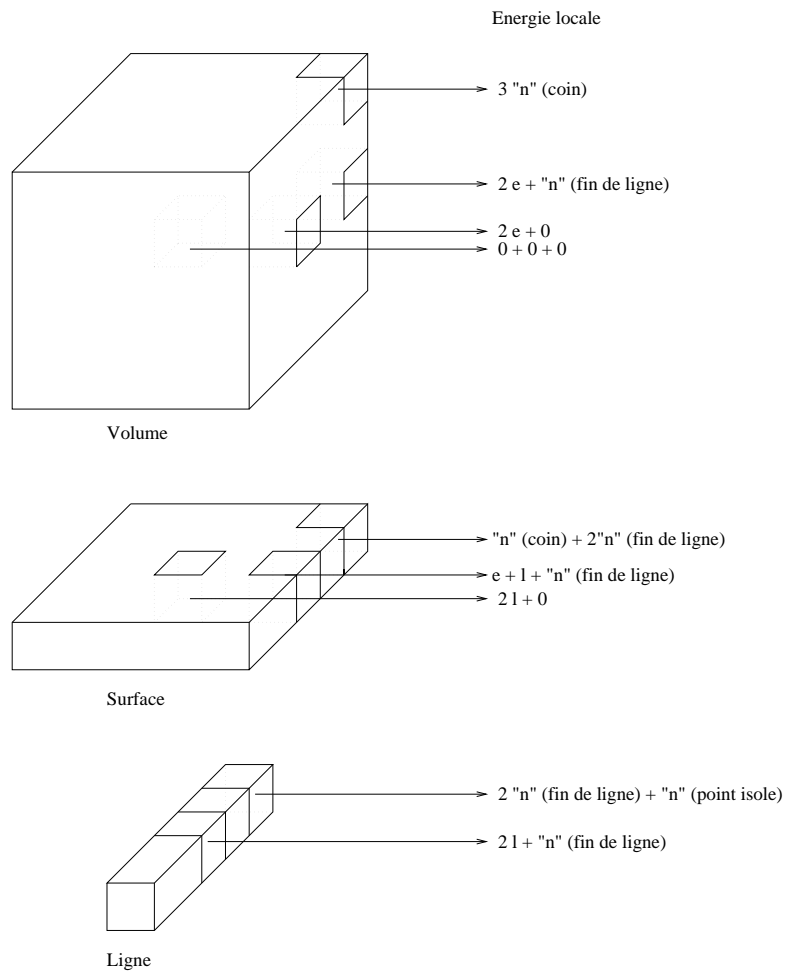


FIG. 2.56 – Pénalisation des objets par le chien-modèle 3D

l'importance du modèle a priori pour obtenir un effet de régularisation. Dans ce cas, les propriétés de préservation des structures fines du chien-modèle sont mises en valeur (voir la figure 2.58). Certaines parties du LCR sont perdues avec le modèle de Potts sur l'image de qualité moyenne alors qu'elles sont préservées avec le chien-modèle.

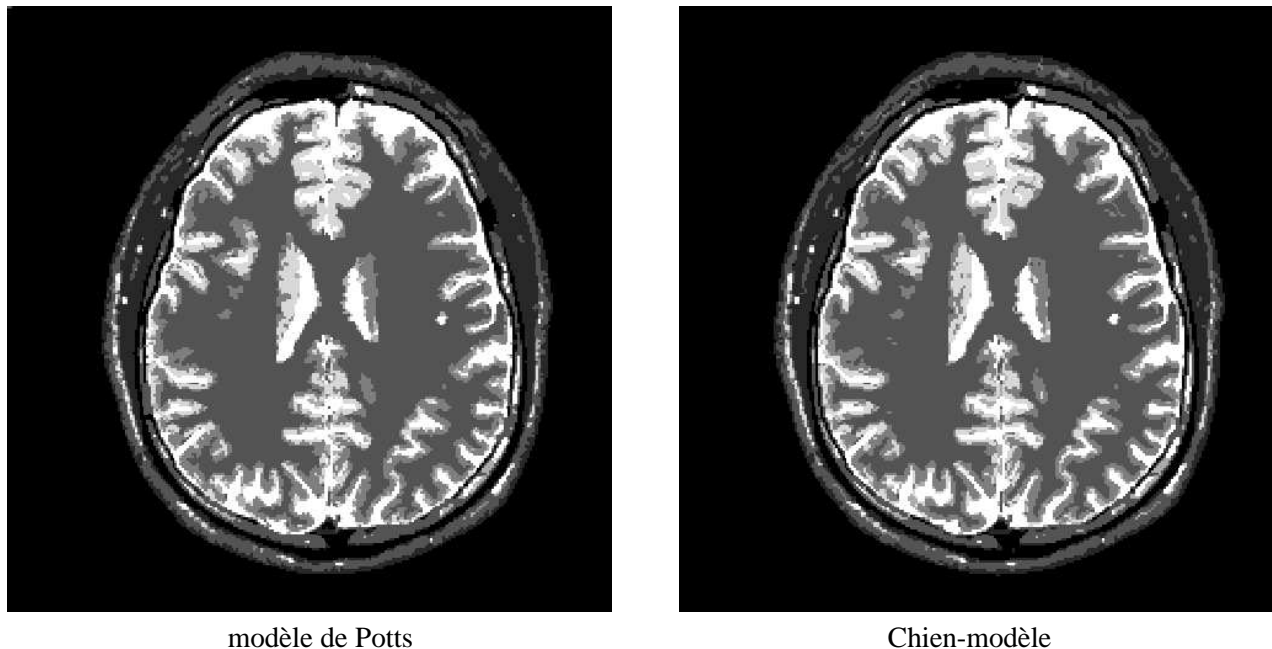


FIG. 2.57 – Segmentation de l'image de bonne qualité

**Corrections des non-stationarités** Les IRM présentent souvent de bons rapports signal à bruit. En revanche, la qualité des segmentations souffre parfois de la non-stationarité du signal, qui n'est pas pris en compte lors de la définition des classes. Nous proposons ici une approche pour estimer, puis corriger, ces non-stationarités. La problématique concernant l'estimation de ces non-stationarités consiste à faire la distinction entre les gradients dus aux interfaces entre les différentes classes et ceux dus aux non-stationarités. Pour aborder ce problème, nous proposons d'effectuer une première segmentation. Néanmoins, pour limiter l'influence des non-stationarités, nous nous appuyons sur les contours pour effectuer cette segmentation. Cette approche, moins robuste que celles fondées sur les niveaux de gris dans un cadre stationnaire, est nécessaire ici. Toutefois, cette première segmentation nous servira à estimer un modèle des non-stationarités. Il est donc suffisant d'extraire des composantes connexes incluses dans une seule classe sans chercher à segmenter toute la scène.

En chaque voxel, nous définissons une variable aléatoire dans l'espace d'état formé des différentes orientations de la normale d'un contour et d'un état représentant l'absence de contour. Nous considérons la 26 connectivité. Une configuration du champ de Markov ainsi défini est notée  $\mathcal{C}$ . Et l'état du voxel  $s$  est donné par  $\mathcal{C}_s$ . L'espace d'état associé est noté  $D$ . Si un voxel n'appartient pas à un contour,  $\mathcal{C}_s$  prend la valeur 13. Les directions sont codées entre 0 et 12. La valeur d'un voxel de direction  $d$  est donnée par  $\mathcal{C}_s$  ou  $14 + d$  suivant l'orientation de la normale (voir la figure 2.59). Les cliques considérées sont les paires de voxels adjacents. Nous définissons trois types de potentiels définis sur des variables binaires (voir la figure 2.60). Le premier terme régularise la surface dans les directions perpendiculaires à la normale. Soit deux voxels adjacents  $s$  et  $t$  induisant une direction  $d$ . S'ils possèdent la même étiquette et si cette étiquette correspond à une direction orthogonale à la direction  $d$ , cette configuration est favorisée. Ces potentiels sont appelés *terme d'extension de la surface*. Ils s'écrivent comme suit :

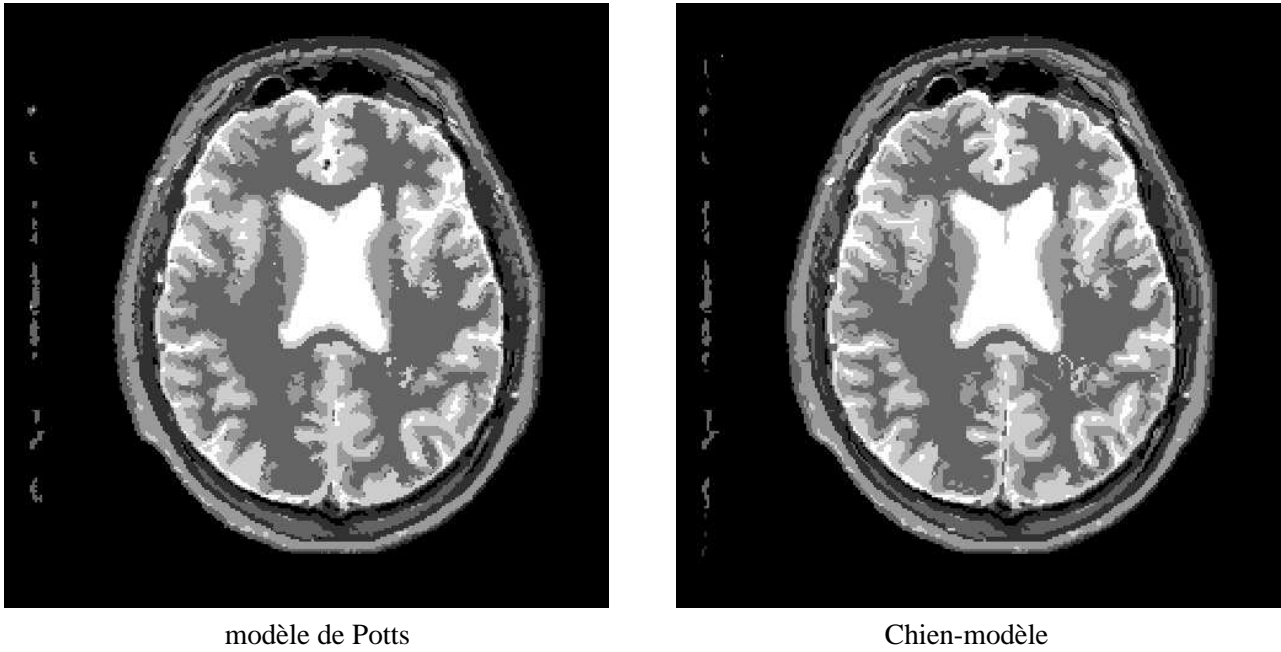


FIG. 2.58 – Segmentation de l'image de qualité moyenne

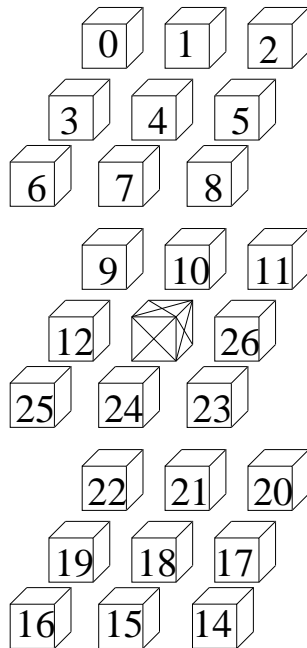


FIG. 2.59 – Les étiquettes d'orientation du processus de surface

$$V_{c=\{s,t\}} = \beta_1 \sum_{d \in D, d \neq 13} \delta_{e_s=d, e_t=d, e_s \vec{e}_t \perp \vec{d}}. \quad (2.125)$$

Le second terme est appelé *terme de cohérence*. A chaque voxel correspondant à un contour, doit correspondre un second voxel, voisin du premier dans la direction considérée, dont l'étiquette correspond à la même direction mais à l'orientation opposée :

$$V_{c=\{s,t\}} = \beta_2 \sum_{d, d \in \{0, \dots, 12\}} \delta_{|e_s - e_t| = 14, e_s \vec{e}_t \parallel \vec{d}}. \quad (2.126)$$

Le troisième terme régularise les voxels n'appartenant à aucun contour. Nous l'appelons simplement *terme de régularisation*. Deux voxels adjacents au sens de la 26 connexité dont l'étiquette est 13 sont favorisés. Ce dernier terme s'écrit :

$$V_{c=\{s,t\}} = \beta_3 \delta_{e_s=13, e_t=13}. \quad (2.127)$$

Pour définir le terme d'attache aux données, considérons pour chaque direction  $d$  entre 0 et 12, le plan

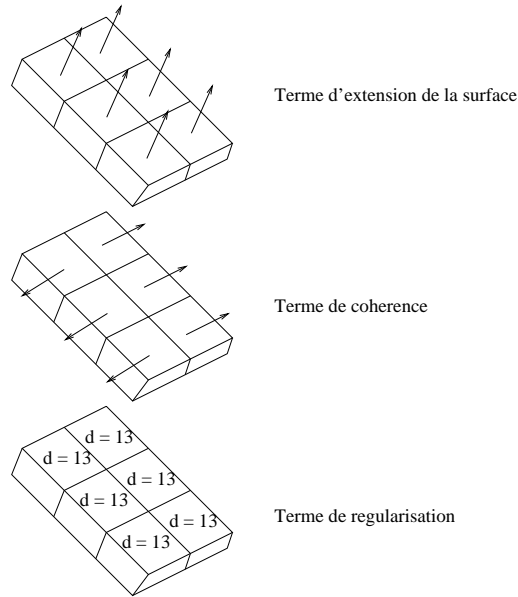


FIG. 2.60 – Configurations favorisées

perpendiculaire contenant le voxel  $s$  (voir la figure 2.61). Nous définissons alors  $\mu_1(d, s)$  (resp.  $\mu_2(d, s)$ ) comme la moyenne des voxels du voisinage de  $s$  se trouvant au-dessus (resp. au dessous) de ce plan. Soit  $x_s$  le niveau de gris de  $s$ . Si  $(x_s - \mu_1(d, s))^2$  est plus petit que  $(x_s - \mu_2(d, s))^2$ , alors nous définissons  $F_s(d) = (\mu_1(d, s) - \mu_2(d, s))^2$  and  $F_s(d + 14) = 0$ . Dans le cas contraire, nous définissons,  $F_s(d) = 0$  et  $F_s(d + 14) = (\mu_1(d, s) - \mu_2(d, s))^2$ . Le terme d'attache aux données est donc donné par :

$$V_{c=\{s\}}(d) = \frac{F_s(d)}{\sup_{u \neq 13} F_s(u)}, d \neq 13 \quad (2.128)$$

$$V_{c=\{s\}}(13) = \frac{\alpha^2}{1 + \sup_{u \neq 13} F_s(u)} \quad (2.129)$$

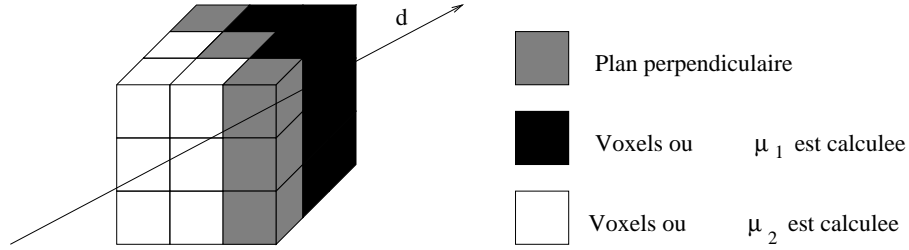


FIG. 2.61 – Calcul de l'attache aux données

**Modèle de non-stationarité et estimation des paramètres associés-** Considérons une composante connexe détectée par la segmentation précédente et notée  $CC_c$ . Ces non-stationarités affectant les IRM sont principalement multiplicatives, nous considérons donc le logarithme des données, soit  $X_s = \log(x_s)$ . Nous choisissons un modèle polynomial des non-stationarités [LJ91, MBP95], de sorte que  $X_s$  s'écrit :

$$X_s = y_c + p(s) + \mathcal{N}(0, \sigma_c^2), \quad (2.130)$$

où  $y_c$  est une constante caractéristique de la classe considérée et  $\mathcal{N}(0, \sigma_c^2)$  est une loi normale centrée.  $p(s)$  est un polynôme modélisant les non-stationarités et qui s'écrit :

$$\forall s = (i, j, k) \in CC_c, p(s) = \sum_{(p, q, r)} a_{p, q, r} i^p j^q k^r. \quad (2.131)$$

Si nous supposons que les probabilités conditionnelles  $P(X_s/s \in C)$  sont indépendantes, nous pouvons écrire la vraisemblance  $L$  comme suit :

$$L = P(CC_c) = \prod_{s \in CC_c} P(X_s/s \in CC_c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \prod_{s \in CC_c} \exp - \left( \frac{(X_s - y_c - p(s))^2}{2\sigma_c^2} \right). \quad (2.132)$$

La log-vraisemblance s'écrit alors :

$$\text{Log}L = -\frac{1}{2\sigma_c^2} \sum_{s=(i, j, k) \in CC_c} (X_s - y_c - \sum_{(p, q, r)} a_{p, q, r} i^p j^q k^r)^2 - \log \sigma_c - \frac{1}{2} \log 2\pi. \quad (2.133)$$

Les estimateurs sont obtenus par maximisation de la log-vraisemblance. En annulant les dérivées partielles, nous obtenons :

$$\frac{\partial \text{Log}L}{\partial a_{p, q, r}} = 2i^p j^q k^r \sum_{s=(i, j, k) \in CC_c} (X_s - y_c - \sum_{(p, q, r)} a_{p, q, r} i^p j^q k^r), \quad (2.134)$$

$$\frac{\partial \text{Log}L}{\partial y_c} = 2 \sum_{s=(i, j, k) \in CC_c} (X_s - y_c - \sum_{(p, q, r)} a_{p, q, r} i^p j^q k^r). \quad (2.135)$$

Considérons un polynôme à  $n$  paramètres et une segmentation contenant  $m$  composantes connexes. Le modèle a alors  $n+m$  inconnues (les paramètres du polynôme et les paramètres  $y_c$ ) et nous avons  $m(n+1)$  équations, qui dépendent linéairement des paramètres  $a_{p, q, r}$ . Le système linéaire s'écrit alors :

$$XA = B, \quad (2.136)$$

où  $X$  est une matrice  $(m(n+1), n+m)$ ,  $A$  est un vecteur  $(n+m)$  et  $B$  un vecteur  $(m(n+1))$ .

Une décomposition en valeurs singulières nous donne :

$$X = UDV^t, \quad (2.137)$$

où  $D$  est une matrice diagonale contenant les valeurs propres du système et  $U$  (resp.  $V$ ) contient les vecteurs propres à gauche (resp. droite). La matrice pseudo-inverse est alors donnée par :

$$X^{-1} = VD^{-1}U^t. \quad (2.138)$$

Nous pouvons alors obtenir un estimateur du vecteur de paramètres  $A$  comme suit :

$$\hat{A} = VD^{-1}U^tB. \quad (2.139)$$

L'estimation est effectuée au sens des moindres carrés.

**Résultats-** L'approche a été testée uniquement sur des données de synthèse. Nous avons utilisé un fantôme 3D sur lequel des non-stationarités multiplicatives polynomiales ont été simulées. Les résultats obtenus sont montrés sur la figure 2.3.2. La figure 2.3.2.b montre la classification obtenue par l'algorithme HVC. Ces résultats sont médiocres car l'hypothèse sous-jacente de stationarité n'est pas respectée. La segmentation fondée sur les contours, bien que moins robuste, donnent de meilleurs résultats dans ce cas, comme le montre la figure 2.3.2.c. Les zones en noir définissent des composantes connexes de la segmentation alors que les voxels en gris représentent les bords. Nous obtenons une sur-segmentation, ce qui suffit pour l'étape d'estimation des non-stationarités. Les données corrigées des non-stationarités sont données sur la figure 2.3.2.d. La correction est satisfaisante, excepté dans la partie basse du cerveau (voir la coupe 52) et de la peau. Dans ces parties, l'hypothèse polynomiale n'est pas vérifiée et le facteur correctif est trop fort. Pour finir, la figure 2.3.2.e montre la classification obtenue après correction des non-stationarités. L'amélioration est évidente.

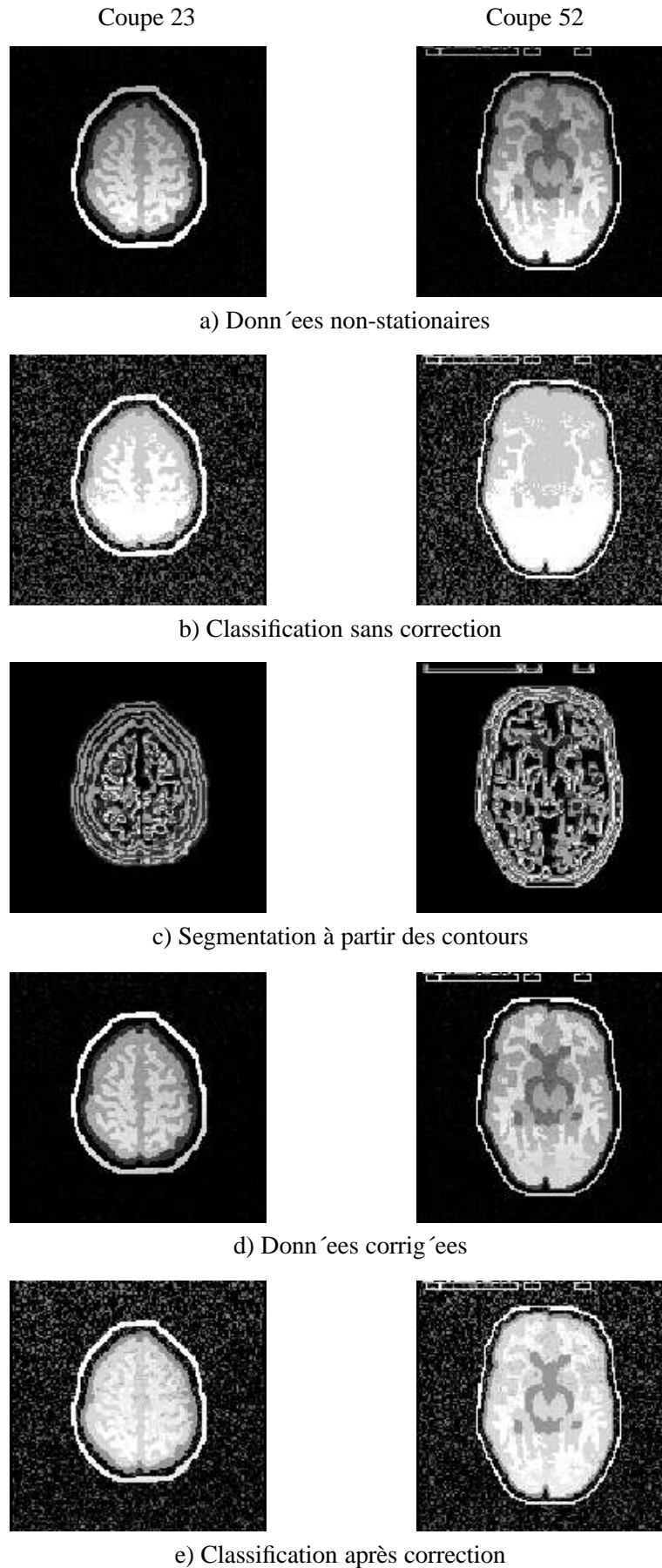


FIG. 2.62 – Correction des non-stationarités pour la segmentation

### 2.3.3 La restauration

#### Restauration de signaux d'IRM fonctionnelle (IRMf)

<sup>15</sup> **Introduction :** Les données de l'IRMf sont une série temporelle de volumes 3D. On se restreindra ici à l'étude de coupes 2D. Les différentes images correspondent à la même coupe cérébrale prise à des instants différents. Le protocole consiste en l'alternance de périodes comprenant  $A$  acquisitions sans stimulus et de périodes comprenant  $B$  acquisitions avec stimulus. Le but de l'analyse d'IRMf est de comparer les distributions des niveaux de gris d'un voxel donnée pour détecter les voxels pour lesquels la distribution durant les périodes de stimulus diffère de façon significative de celle durant les périodes sans stimulus. Ces voxels sont appelés voxels activés. Ils correspondent aux zones corticales activées par le stimulus.

L'approche la plus couramment utilisée pour analyser les IRMf se décompose en trois étapes [WEMN92, FJT94]: (i) restauration du signal par un filtrage gaussien, (ii) calcul d'une carte statistique et (iii) seuillage. Considérons une série temporelle d'IRMf notée  $F = (f(l,t))$  avec  $l \in L$  les coordonnées spatiales et  $t \in T$  la coordonnée temporelle. Soit  $T_A$  le sous-ensemble de  $T$  correspondant à la condition  $A$  et  $T_B$  celui correspondant à la condition  $B$ . Une carte statistique paramétrique (SPM) est une image pour laquelle la valeur en un voxel, appelée "z-score", est donnée par un test statistique comparant les distributions sur  $T_A$  et sur  $T_B$  :

$$\forall l \in L, SPM(l) = test(d_A(l), d_B(l)), \quad (2.140)$$

où :

$$d_A(l) = \{f(l,t), t \in T_A\} \text{ and } d_B(l) = \{f(l,t), t \in T_B\} \quad (2.141)$$

Les tests les plus couramment utilisés en pratique sont le test paramétrique de Student (t-test) et le test non-paramétrique de Kolmogorov-Smirnov. Les zones jugées activées de manière significative sont obtenues par seuillage de la SPM [FWF<sup>+</sup>94, WF95]. Le seuil est établi en testant l'hypothèse nulle. Une p-valeur qui prévient contre les fausses alarmes est fixée (en principe 0.05) et le seuil est calculé pour limiter la probabilité de fausse alarme à  $p$ . Un filtre spatial gaussien est souvent utilisé dans les études d'IRMf pour réduire le bruit avant de calculer la SPM et ainsi réduire les fausses alarmes. Le lissage introduit est alors pris en compte pour le calcul de la p-valeur [FHP<sup>+</sup>95, FCF<sup>+</sup>95]. Néanmoins, le signal lui-même est affecté par ce filtre passe-bas, qui introduit un effet de flou et un déplacement possible des zones activées. En outre, les signaux de faible amplitude sont détruits. Ainsi, la détection est améliorée, du fait de la réduction du bruit, mais les caractéristiques du signal et la précision de la localisation des zones détectées sont partiellement perdues. Pour résoudre ce problème, nous avons proposé de restaurer les données plutôt que de les filtrer. Le processus de restauration réduit le bruit tout en préservant la forme du signal.

Nous avons développé un modèle spatio-temporel à trois dimensions : deux dimensions spatiales correspondant à une coupe et une dimension temporelle. Ce modèle est d'abord validé par une mesure quantitative de l'effet de restauration sur des signaux synthétiques (signal carré, sinusoïdal et prototype de réponse hémodynamique). Nous appliquons ensuite ce modèle de restauration à des images d'IRMf obtenues par un imageur echo-planaire (EPI). Les résultats sont comparés avec ceux obtenus en utilisant un filtrage gaussien comme pré-traitement ou sans pré-traitement.

**Un modèle markovien spatio-temporel :** Rappelons la définition du  $\Phi$ -modèle qui va nous permettre de définir notre modèle *a priori*  $P(Y)$ . Le but de la restauration est d'appliquer des contraintes d'homogénéité et de douceur sur la solution. Les filtres passe-bas lissent les données sans prendre en compte le signal sous-jacent. Ceci résulte en la perte de l'information présente dans les hautes

---

<sup>15</sup>. Nous résumons ici des travaux effectués à l'Institut Max Planck de Leipzig en collaboration avec F. Kruggel. Bibliographie : [17, 16, 47, 46, 48]



fréquences. Nous développons ici un modèle markovien préservant les structures fines et les contours. Nous considérons un  $\Phi$ -modèle pour préserver les transitions entre les zones activées et non-activées dans le domaine spatial et entre les périodes d'activation et de repos dans le domaine temporel.

La formulation générale [GR92] pour les  $\Phi$ -fonctions est la suivante :

$$\Phi(u) = \frac{-\beta}{1 + (|u|/\delta)^p}. \quad (2.142)$$

Une  $\Phi$ -fonction dépend de trois paramètres :  $\beta$ ,  $\delta$  et  $p$ . Le paramètre  $\beta$  est un facteur d'échelle suivant l'axe des ordonnées. Il représente la force de l'interaction. Les deux autres paramètres définissent la forme de la  $\Phi$ -fonction. La  $\Phi$ -fonction a un minimum égal à  $-\beta$  en 0 et l'axe des  $X$  pour asymptote. Si  $u$  est un gradient, la valeur de  $\beta$  correspond au coût d'un contour. Le paramètre  $\delta$  permet de définir la différence entre deux voxels à partir de laquelle nous avons une forte probabilité d'être en présence d'un contour ou d'une transition. Le paramètre  $p$  définit la forme du potentiel autour de la valeur de transition ainsi que la forme du potentiel autour de 0 donc du lissage. Après une phase de calibration, nous avons fixé  $p$  à 2.

Nous considérons une 6-connexité composée de quatre voisins spatiaux et deux voisins temporels (voir figure 2.63). Sur les cliques de deux sites, nous définissons une  $\Phi$ -fonction comme potentiel.

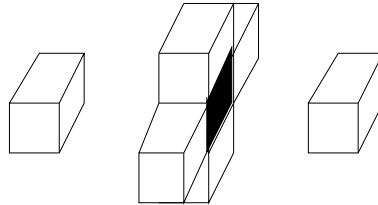


FIG. 2.63 – *Voisinage spatio-temporel : quatre voisins spatiaux et deux voisins temporels.*

Dans le domaine spatial, nous pouvons prévoir deux types de transition : (i) les transitions entre zones activées et zones inactivées et (ii) les transitions entre les différentes structures anatomiques (matière blanche et matière grise notamment). Pour être aveugle aux transitions anatomiques nous normalisons les données en leur soustrayant la sortie d'un filtre moyenneur dans le domaine temporel. La série temporelle associée à chaque voxel est alors de moyenne nulle. Ainsi, nous pouvons appliquer la même régularisation dans le domaine spatial et le domaine temporel. Comme nous avons deux fois plus de voisins spatiaux que de voisins temporels, nous considérons des interactions plus fortes pour les potentiels temporels :  $\beta_{temp} = 2\beta_{spat} = \beta$ . Les hauteurs des transitions sont les mêmes dans les domaines spatiaux et temporels car elles représentent la différence entre activation et non-activation. Nous prenons donc :  $\delta_{temp} = \delta_{spat} = \delta$ . Pour certains protocoles d'acquisition, les voxels ne sont pas isotropes, les tailles suivant les lignes (axe I) et les colonnes (axe J) diffèrent. Pour compenser cette anisotropie, nous réduisons les interactions spatiales le long de l'axe I par le rapport d'anisotropie des pixels (cette normalisation n'étant bien entendu qu'une approximation). Le modèle proposé s'écrit donc :

$$P(Y) = \frac{1}{Z} \exp - \left[ \sum_t \sum_{(i,j)} V_T(y_{i,j}(t), y_{i,j}(t+1)) + \sum_{(i,j)} \sum_t V_I(y_{i,j}(t), y_{i+1,j}(t)) \right. \\ \left. + \sum_{(i,j)} \sum_t V_J(y_{i,j}(t), y_{i,j+1}(t)) \right], \quad (2.143)$$

où

$$V_T(y_{i,j}(t), y_{i,j}(t+1)) = \frac{-2\beta}{1 + (y_{i,j}(t) - y_{i,j}(t+1))^2 / \delta^2}, \quad (2.144)$$

$$V_I(y_{i,j}(t), y_{i+1,j}(t)) = \frac{-A\beta}{1 + (y_{i,j}(t) - y_{i+1,j}(t))^2 / \delta^2}, \quad (2.145)$$

$$V_J(y_{i,j}(t), y_{i,j+1}(t)) = \frac{-\beta}{1 + (y_{i,j}(t) - y_{i,j+1}(t))^2 / \delta^2}, \quad (2.146)$$

A représente le rapport d'anisotropie des voxels dans le domaine spatial. Le modèle *a priori* dépend donc de deux paramètres :  $\beta$  et  $\delta$ .

Pour compléter le modèle, il nous reste à définir la vraisemblance. Nous faisons l'hypothèse classique d'indépendance conditionnelle :

$$P(X|Y) = \prod_{(i,j),t} p(x_{(i,j)}(t)|y_{(i,j)}(t)). \quad (2.147)$$

En l'absence d'information précise sur la nature du bruit dans les données d'IRMf, une première approche pourrait être de considérer une vraisemblance gaussienne. Néanmoins, nous préférons un modèle à queue lourde pour prendre en compte le bruit correspondant aux artefacts physiologiques et ceux dus au scanner. Pour équilibrer l'influence de la vraisemblance par rapport au modèle *a priori*, nous prenons une  $\Phi$ -fonction :

$$p(x_{(i,j)}(t)|y_{(i,j)}(t)) = \frac{1}{Z} \exp -V_L(y_{(i,j)}(t)), \quad (2.148)$$

où

$$V_L(y_{(i,j)}(t)) = \frac{-\beta_L}{1 + (y_{(i,j)}(t) - x_{(i,j)}(t))^2 / \delta^2}. \quad (2.149)$$

L'énergie globale est définie à une constante additive près, nous pouvons donc, sans perte de généralité, imposer  $\beta_L = 1$ . Le modèle ne dépend alors que de deux paramètres :  $\beta$  and  $\delta$ .

**Validation sur des données synthétiques-** L'avantage principal de la méthode proposée est de préserver la forme du signal durant l'opération de filtrage du bruit. Nous devons aussi vérifier que les connaissances *a priori* du modèle markovien n'induisent pas d'artefacts sur le signal. Pour ce faire, nous considérons différents modèles synthétiques d'ondes et étudions leur restauration.

Nous modulons trois fonctions test (une onde sinusoïdale, une onde carrée et un prototype de réponse hémodynamique) sur un extrait d'une expérience d'IRMf pour lequel aucune activation significative n'a été détectée par les méthodes standard. Ces données sont corrigées des fluctuations basse fréquence. Nous comparons la détection de l'activation après un filtrage spatial gaussien ("gauss"), une restauration par le modèle proposé ("mrf") et sans prétraitement ("native"). Nous calculons le "z-score" du pic d'activation et le taux de recouvrement pour les trois fonctions tests. Le recouvrement du signal  $c$  pour la fonction test  $m(t)$  à partir de la série temporelle pré-traitée  $\psi$  est donnée par :

$$c = 1 - \frac{\sum_t^N (y_s(t) - (am(t) + b))^2}{\sum_t^N y_s(t)^2}, \quad (2.150)$$

où  $c$  est maximisée suivant  $a$  et  $b$ . Ainsi,  $c = 1$  signifie un recouvrement parfait alors que  $c = 0$  reflète une perte totale du signal. Les résultats sont présentés dans les tableaux 2.17 et 2.18. Pour le filtrage gaussien nous avons utilisé un sigma de 0.8 et pour la restauration par le modèle markovien  $\beta = 0.6$  et  $\delta = 60$ .

**Validation sur une expérience fonctionnelle-** Cet algorithme est évalué à partir d'une expérience faite à l'Institut Max Planck pour les neurosciences de Leipzig sur un scanner Bruker 3.0 Tesla MED-SPEC System. Pendant une période de 18 secondes un son standard ( $f = 600\text{Hz}$ ), un son déviant

Fonction d'onde	native	gauss	mrf
sinus	0.456	0.581	0.872
hemo	0.456	0.579	0.804
carrée	0.458	0.556	0.631

TAB. 2.17 – *Taux de recouvrement des trois fonctions tests après correction des fluctuations des basses fréquences seules (native), suivie d'un filtrage gaussien (gauss) et suivie d'une restauration par champ de Markov (mrf).*

Fonction d'onde	native	gauss	mrf
sinus	8.0	10.6	14.3
hemo	12.9	14.7	19.1
carrée	12.8	14.0	18.2

TAB. 2.18 – *“Z-scores” des trois fonctions tests après correction des fluctuations des basses fréquences seules (native), suivie d'un filtrage gaussien (gauss) et suivie d'une restauration par champ de Markov (mrf).*

( $f = 660\text{Hz}$ ) ou un nouveau son unique (accident de voiture, aboiement de chien) est présente aux deux oreilles du sujet. Le séquençage est aléatoire. On demande aux sujets de compter le nombre de sons déviants. Un total de 215 périodes est enregistrée. Les données brutes sont prétraitées pour supprimer les coupes contenant des artefacts et corriger les mouvements du sujet. Les voxels activés sont détectés par un coefficient de corrélation avec une fonction carrée standard translatée du délai caractéristique de 6s entre le stimulus et la réponse de l'IRMf. Ces coefficients de corrélation sont convertis en z-scores, seuillés ( $z \geq 2.75$ ), et superposés aux images anatomiques  $T1$ .

Nous comparons les résultats avant et après restauration ainsi qu'avec un filtrage gaussien. Nous comparons aussi la forme du signal en moyennant les séries temporelles des voxels d'une même zone activée. Lorsque l'on compare différents pré-traitements pour l'IRMf il faut garder à l'esprit qu'il n'y a pas de “vérité terrain” avec laquelle la validité du résultat ou l'amélioration due à un traitement puisse être mesurée. Néanmoins, une évaluation des données brutes et des arguments neuro-anatomiques nous permettent de distinguer entre les zones de réelle activation corticale, les activations dans les compartiments veineux, les fausses alarmes dues au bruit, les artefacts, etc. Les zones réelles d'activation sont supposées : (i) suivre étroitement la bande corticale, (ii) être situées dans des zones liées à la tâche effectuée par le sujet, et (iii) avoir une certaine forme (i.e. force de l'activation versus étendue de l'activation).

La figure 2.64 montre la carte des z seuillée ( $t = 2.75$ ) obtenue respectivement sur les données brutes, après un filtrage gaussien ( $\sigma^2 = 0.8$ ) et après la restauration ( $\delta = 4, \beta = 0.4$ ). Comme escompté, des zones bilatérales d'activation très significative sont trouvées sur le sillon temporal supérieur et, dans une moindre mesure, dans le thalamus de chaque côté. En comparaison avec l'analyse sur les données brutes (ligne du haut), le filtrage gaussien (ligne du milieu) apporte un réhaussement général du signal mais aussi un flou spatial considérable. En revanche, la schéma de restauration proposée (ligne du bas), tout en réhaussant encore plus le signal permet de conserver les détails anatomiques.

Sur la figure 2.65 nous avons tracé l'évolution temporelle du signal IRMf sur une période (les trois courbes correspondent aux données brutes, après filtrage gaussien et après restauration). Ces signaux sont moyennés sur la région activée du cortex auditif droit (c.a.d. l'aire du côté gauche des images de la troisième colonne de la figure 2.64). La restauration apporte une meilleure interprétation de l'activation grâce à la régularisation dans le domaine temporel. Le bruit est supprimé mais les pentes d'activation et de désactivation ne sont pas floues. Un filtre gaussien temporel aurait également supprimé le bruit mais au prix d'une déformation de ces pentes.

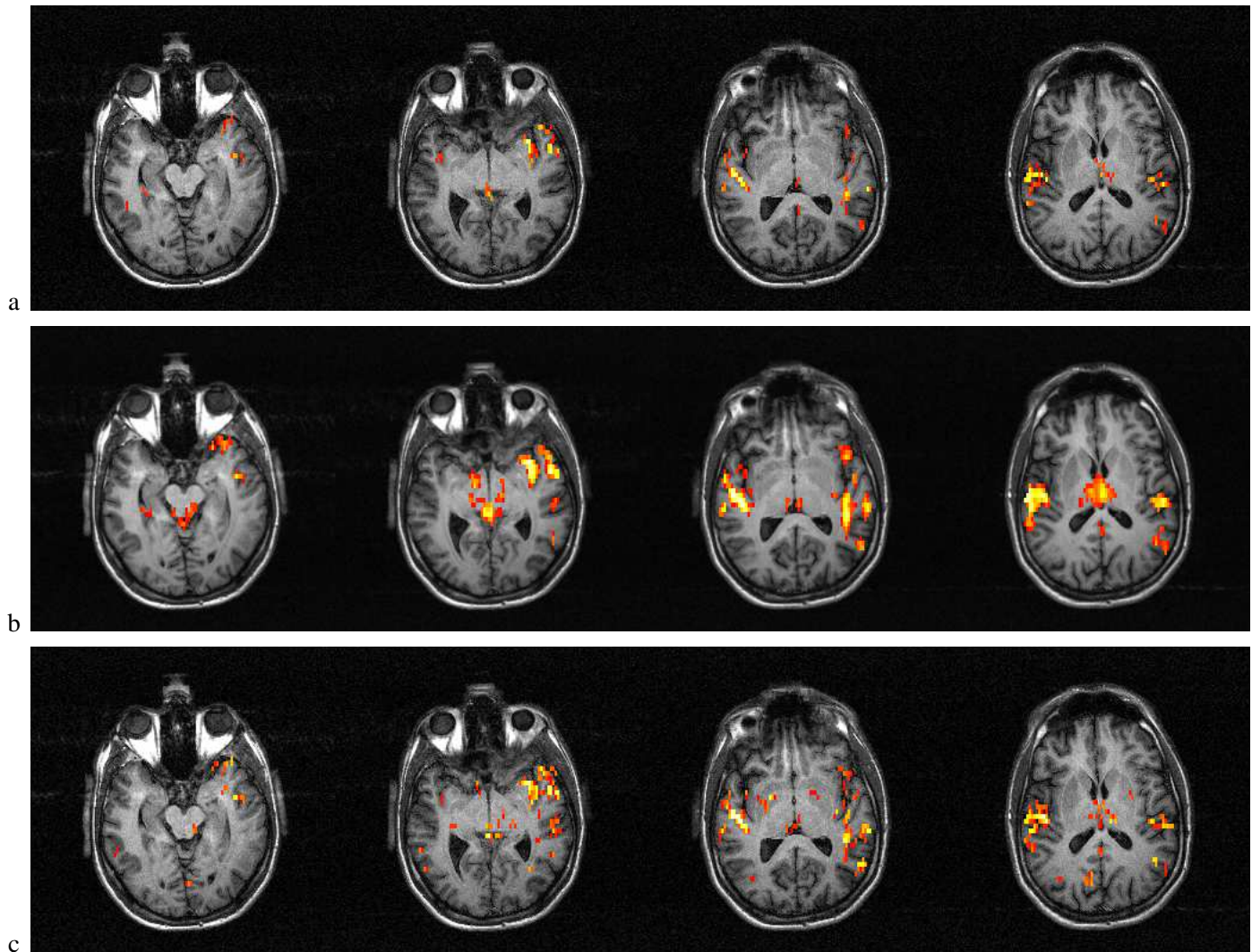


FIG. 2.64 – Evaluation statistique d'une expérience d'IRMf: coefficient de corrélation seuillé à 2.75 sur les données originales (a), les données filtrées par un noyau spatial gaussien  $\sigma = 0.8$  (b), les données restaurées (c).

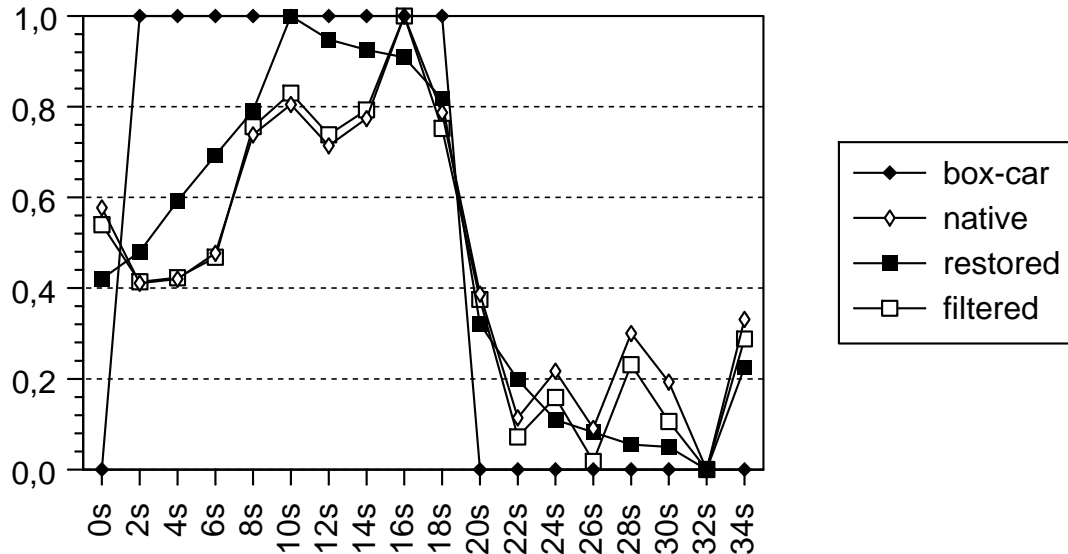


FIG. 2.65 – Evolution temporelle du signal dans le cortex auditif droit : onde carrée (translatée), données brutes, données filtrées par un filtre spatial gaussien et données restaurées par le modèle proposé.

La contrepartie du modèle markovien est l'accroissement du temps de calcul. L'optimum de l'énergie est obtenu par recuit simulé. Dans le cas présentée ici, c.a.d. pour 4 coupes de  $128 \times 64$  voxels et 912 échantillons temporels, le temps de calcul est de 2h sur une station standard. Des algorithmes sous-optimaux, mais plus rapides, peuvent être une alternative si le temps de calcul devient rédhibitoire. L'avantage du modèle markovien est la distinction introduite entre le bruit et le signal, même haute fréquence, par le modèle *a priori*. Cela permet une meilleure définition des zones activées et une meilleure caractérisation du signal dans ces zones. De plus, aucun modèle de la réponse hémodynamique n'est nécessaire.

### Restauration d'images angiographiques (MRA)

<sup>16</sup> L'interprétation des images angiographiques nécessite une délimitation précise des vaisseaux sanguins. Sur les données réelles, le contraste est trop faible pour faire une bonne interprétation, c'est pour cette raison qu'un produit iodé est injecté au patient avec l'acquisition de l'image. Pour réduire le volume du produit injecté et améliorer l'interprétation, il est nécessaire de restaurer les images. Nous abordons ici ce problème de restauration 3D en prenant soin de préserver les contours des vaisseaux qui sont des structures fines. Pour ce faire, nous proposons d'utiliser un champ markovien 3D à deux composantes, l'une étant le niveau de gris et la seconde une étiquette de contours permettant d'inhiber les interactions entre voxels voisins n'appartenant pas au même objet. Ce modèle peut donc être vu comme une généralisation du processus de lignes en 2D [GG84]. La différence majeure est qu'ici, le processus de contours n'est pas défini sur la trame duale mais sur la trame de l'image.

**Modélisation-** Le processus de surface, défini sur un espace d'état à 27 étiquettes (l'étiquette 13 représente l'absence de contour), est le même que celui décrit dans le paragraphe 2.3.2.

Une configuration du processus de restauration est notée  $\Sigma_V = \{\sigma_v\}$ . Pour réduire le temps de calcul, l'optimisation est effectuée par un algorithme ICM, itérativement appliquée sur le processus de surface et sur le processus de restauration. Nous pouvons calculer les différentes probabilités conditionnelles

16. Nous résumons ici des travaux effectués à l'Université Catholique de Louvain (KUL).

locales correspondant au processus de surface car nous avons un espace d'état réduit à 27 éléments. En revanche, l'espace d'état du processus de restauration contient 256 éléments. Pour avoir accès rapidement, c'est à dire analytiquement, au minimum de l'énergie conditionnelle locale, nous adoptons des potentiels gaussiens pour ce processus. Ces interactions gaussiennes sont activées ou non suivant la valeur du processus de surface (voir la figure 2.66). Le modèle contient donc deux termes. Le premier concerne les voxels à l'intérieur des objets ( $d = 13$ ). Le second concerne les voxels de surface. Soit un tel voxel  $v$  dont l'étiquette de surface est  $d_v$ . Nous considérons alors une interaction entre le voxel  $v$  et son voisin  $w$  si et seulement si  $v\vec{w} \perp \vec{d}_v$  et  $d_v = d_w$ . Ainsi, un voxel de surface est en interaction avec un voxel appartenant au même objet. Le terme de régularisation s'écrit donc ainsi :

$$V_{c=\{v,w\}} = (\sigma_v - \sigma_w)^2 \left[ \delta_{d_v=d_w, \vec{d}_v \perp v\vec{w}} + \delta_{d_v=13, d_w=13} \right]. \quad (2.151)$$

Nous définissons maintenant le terme d'attache aux données. Soit  $f$  les données (initiales ou après un

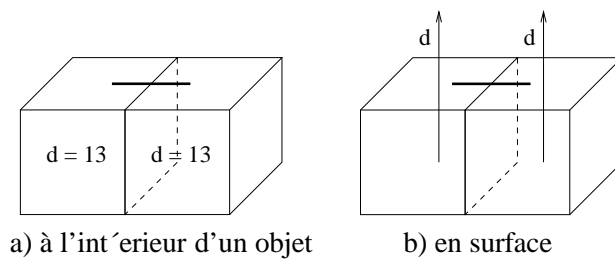


FIG. 2.66 – **Processus de restauration : interactions activées.**

pré-traitement). Nous considérons un potentiel gaussien d'ordre 1 :

$$V_c(\sigma_v) = \lambda(\sigma_v - f(\sigma_v^0))^2, \quad (2.152)$$

où  $\sigma_v$  est l'état du processus de restauration et  $f$  le niveau de gris correspondant sur les données.

**Validation sur des données de synthèse-** Nous considérons un objet formé de trois cylindres (voir la figure 2.67). La base des cylindres est de  $3 \times 3$  voxels. Un bruit gaussien est ajoutée. La moyenne du fond (resp. de l'objet) est de 40 (resp. 100) alors que l'écart type du bruit est de 30 (voir la figure 2.68). La restauration obtenue sans le processus de surface est montrée sur la figure 2.69. Un effet de flou est produit par les interactions gaussiennes entre le fond et l'objet. Sur les coupes extrêmes (coupes 1 et 5 de la figure 2.69), on reconnaît l'empreinte de l'objet. Cet effet indésirable disparaît avec le processus de surface, ce qui nous permet une plus forte régularisation interne aux objets.

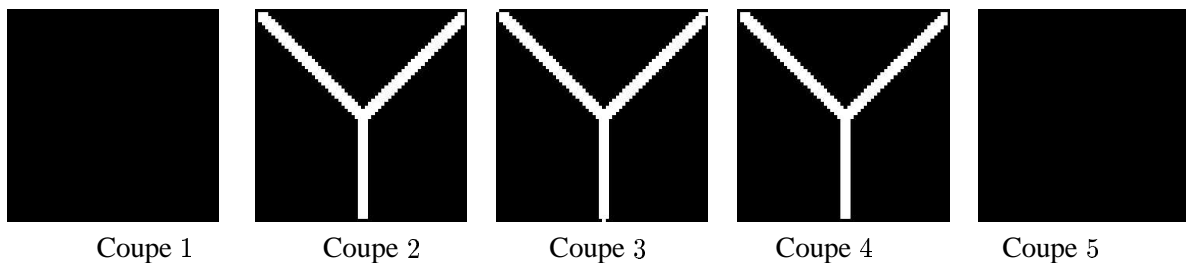
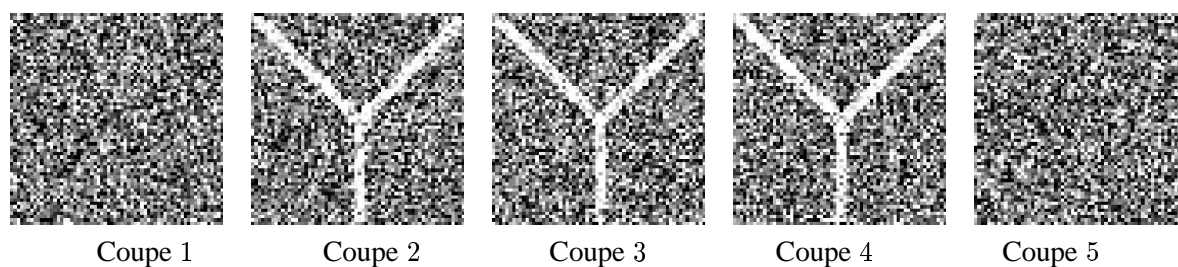
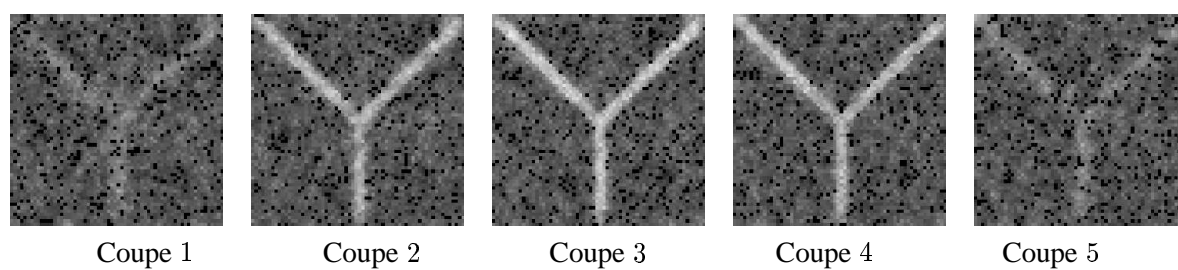
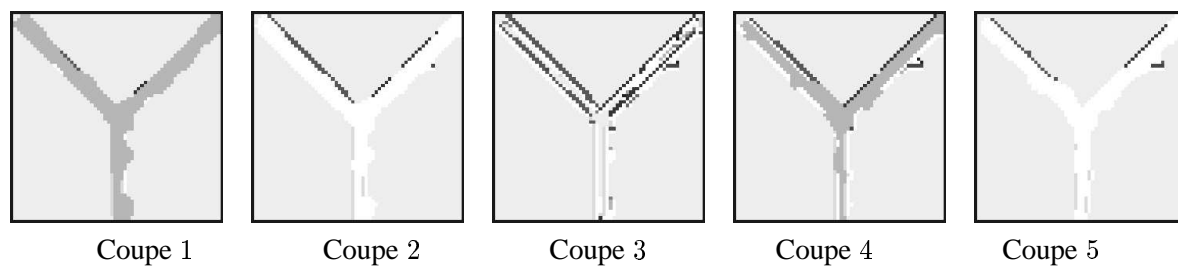
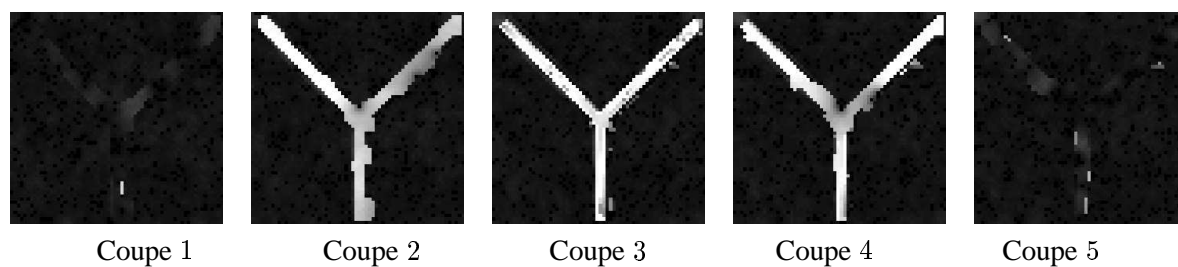


FIG. 2.67 – **Données de synthèse**

**Application à l'angiographie-** De façon générale, les vaisseaux sanguins ont une radiométrie plus forte que le fond. Néanmoins, la radiométrie n'est pas stationnaire. Par exemple, les yeux ou une éventuelle tumeur accroissent la moyenne locale du signal. Pour éviter une sur-détection dans ces zones nous appliquons un pré-traitement visant à réduire ces non-stationarités. Ce pré-traitement est résumé sur la

FIG. 2.68 – **Données bruitées**FIG. 2.69 – **Données restaurées sans processus de surface**FIG. 2.70 – **Résultat du processus de surface**FIG. 2.71 – **Données restaurées avec le processus de surface**

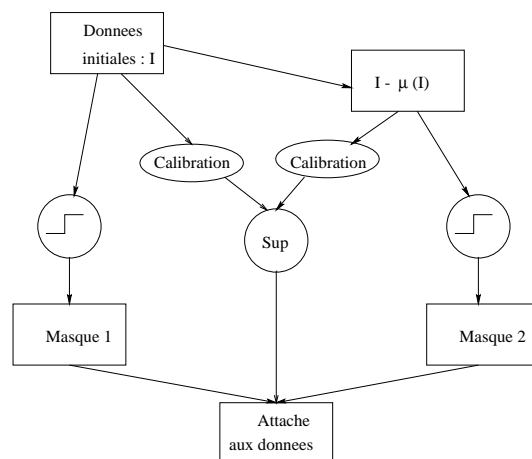


FIG. 2.72 – Synoptique du pré-traitement

figure 2.72. Nous définissons un second volume de données en soustrayant la moyenne locale à l'image en chaque voxel. Un vaisseau sanguin est alors caractérisé par une forte réponse dans l'image initiale et dans ce second volume. Pour utiliser ces deux informations, nous calibrons les données en imposant aux deux volumes une même moyenne et une même variance. Le terme d'attache aux données s'applique alors sur le maximum des deux volumes obtenus.

Pour réduire encore le temps de calcul, nous seuillons l'image du maximum. Les valeurs les plus faibles, pour lesquelles nous savons que les voxels correspondant n'appartiennent à aucun vaisseau, sont mise à zéro. Les voxels correspondant sont masqués et ne sont pas visités lors de l'ICM. Nous testons l'algorithme sur deux jeux de données correspondant à deux résolutions différentes. La résolution la plus fine entraîne un bruit plus conséquent. Néanmoins, la phase de calibration des données nous permet de conserver les mêmes valeurs des paramètres pour les deux expériences. La figure 2.73 montre quelques coupes du premier volume. Les vaisseaux parallèles au plan de coupe sont visibles sur les données originales, ce qui n'est pas le cas des autres en raison du bruit. La restauration permet une meilleure interprétation des données (voir la figure 2.74). De plus, le processus de surface permet d'éviter les effets de flou (voir la figure 2.75). La figure 2.76 montre les projections du maximum d'intensité (chaque pixel correspond au maximum d'intensité suivant l'axe perpendiculaire) des images originale et restaurée.

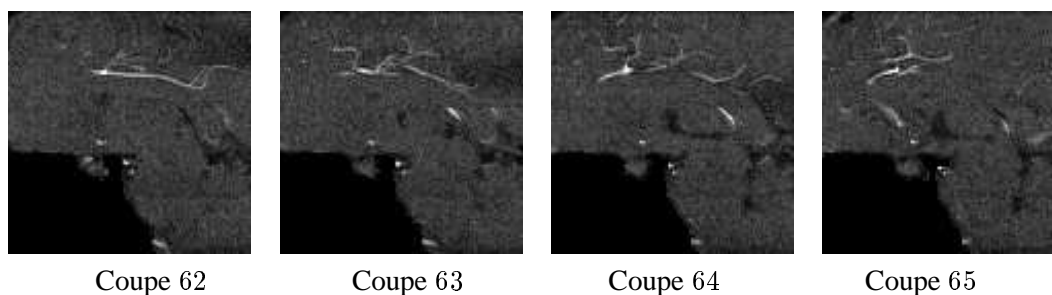


FIG. 2.73 – Volume angiographique initial numéro 1

Le second volume test a une résolution plus fine que le précédent. Néanmoins, les résultats obtenus sont similaires comme le montre la figure 2.78 et ceci en utilisant des valeurs identiques pour les paramètres, excepté pour le paramètre  $\alpha$  de l'équation (2.129).



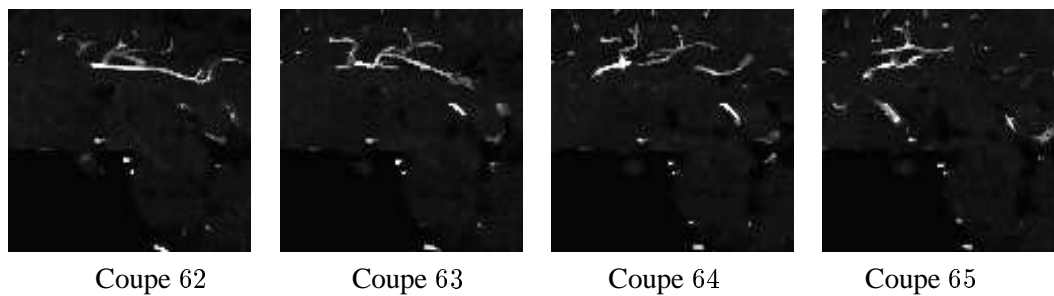


FIG. 2.74 – Volume angiographique restauré numéro 1

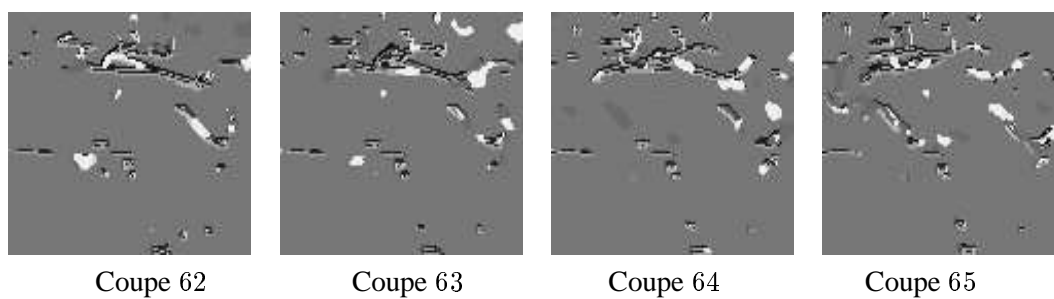


FIG. 2.75 – Processus de surface numéro 1

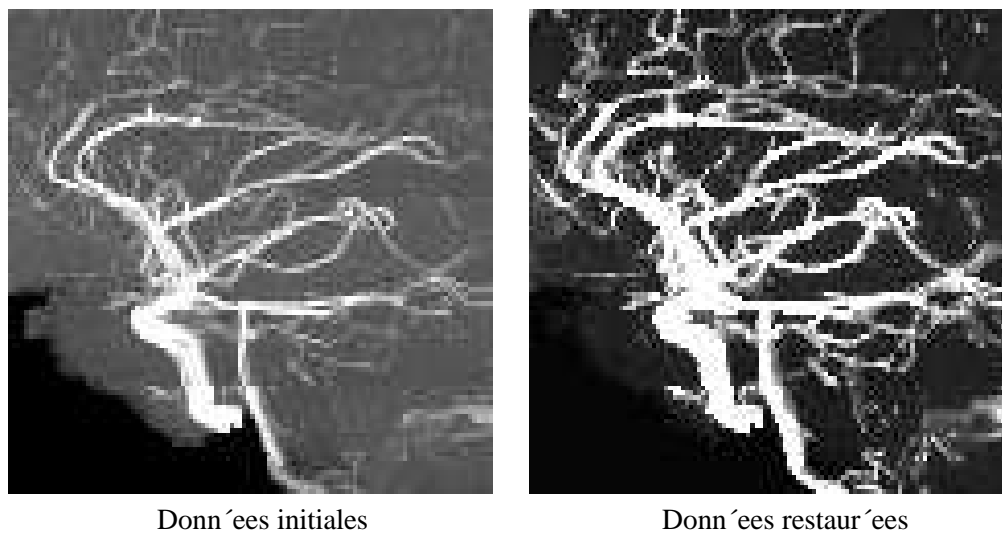


FIG. 2.76 – Projection par maximum d'intensité numéro 1

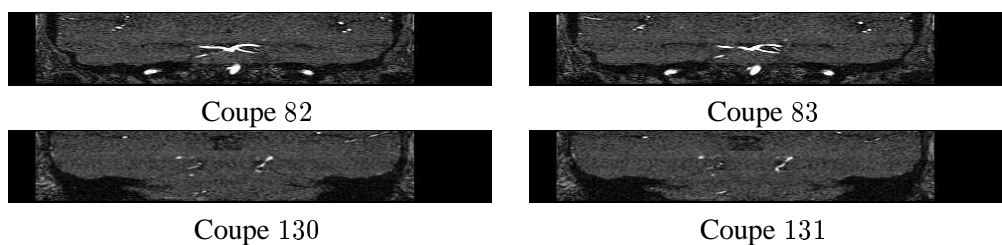


FIG. 2.77 – Volume initial numéro 2



FIG. 2.78 – Volume restauré numéro 2

### L'approche MPI

<sup>17</sup> Nous considérons ici conjointement les problèmes de restauration et de segmentation d'image. L'image restaurée est donnée par une convolution locale de la carte de segmentation par des formes élémentaires (paraboloïdes tronqués), appelées pixons, dont les valeurs des paramètres en chaque point sont données par la carte des pixons. Cette approche s'inspire de travaux effectués dans le cadre de la déconvolution d'images en astronomie [PP93b]. La scène est décrite par trois cartes : la carte de segmentation, la carte de restauration et la carte des pixons. La taille des pixons représente l'échelle locale de la scène sous-jacente.

Notons les données par  $X$  et respectivement les carte de restauration, de segmentation et des pixons par  $R$ ,  $S$  et  $K$ . Nous considérons le problème de la maximisation de la probabilité a posteriori  $P(R, S, K | X)$ . En appliquant la règle de Bayes, nous obtenons :

$$P(R, S, K | X) \propto P(X | R, S, K) P(R, S, K) \quad (2.153)$$

$$\propto P(X | R, S, K) P(R | S, K) P(S, K). \quad (2.154)$$

où  $P(X | R, S, K)$  correspond à l'attache aux données et  $P(S, K)$  est l'a priori sur les cartes de segmentation et des pixons. Nous ne supposons pas le nombre de classes de l'image segmentée connu. Nous introduisons un a priori d'information, c'est-à-dire un a priori entropique pour extraire un nombre réduit de niveaux de gris (les étiquettes) permettant de décrire la scène (voir le paragraphe 2.2.1).

La segmentation permet d'extraire les objets mais en donne une description grossière. Pour affiner cette description, nous proposons de restaurer l'image avec une convolution locale de l'image segmentée par un noyau variant spatialement. Ce noyau, appelé pixon, définit l'échelle locale (ou la résolution locale) des objets. Le concept de pixon a été introduit, sous une forme différente, par Piña et Puetter [PP93a] pour restaurer des images d'astronomie.

Nous considérons une information contextuelle à la fois sur l'image segmentée et sur la carte des pixons. L'image segmentée est modélisée par un modèle de Potts. Sur la carte des pixons, des structures fines, représentant les contours doivent apparaître. Nous considérons alors le chien-modèle comme a priori sur cette carte. Ceci définit une description bas niveau de l'image que nous nommons modèle MPI ("*Markov Pixon Information model*").

**Le I-modèle :** le modèle est exprimé sous la forme d'une énergie que nous allons minimiser. Cette énergie se décompose en plusieurs termes. Le premier modèle, le I-modèle, est composé des deux termes suivants :

Un a priori d'information :

$$E_I = -\alpha_I \sum_{i=1}^n \frac{N_i}{N} \ln \frac{N_i}{N}. \quad (2.155)$$

avec  $N$  le nombre de pixels et  $N_i$  le nombre de pixels de la classe  $i$ , et un terme d'attache aux données :

$$E_D = \sum_{l \in L} \frac{(x(l) - s(l))^2}{\sigma^2} \quad (2.156)$$

où  $\sigma$  est l'écart type des données et  $s$  la moyenne de la classe associée à  $l$ . A ce niveau, nous avons le modèle de classification décrit dans le paragraphe 2.2.1.

**Le PI-modèle :** L'information, dans le I-modèle, est considérée au niveau du pixel. Or, la plupart des images n'ont pas une quantité d'information uniforme spatialement. Des zones comme le fond

---

<sup>17</sup>. Nous résumons ici un travail effectué à l'Institut Max Planck de Leipzig en collaboration avec F. Kruggel. Bibliographie : [15]

de l'image, contiennent peu d'information, alors que les détails des objets nécessitent une plus haute résolution pour être décrits. L'ajout des pixons dans la description de l'image permet d'obtenir une description multi-échelle. L'image restaurée est donc définie comme la convolution locale de l'image segmentée par les pixons :

$$R(r(l), l \in L) = (K \otimes S)(r(l), l \in L), \quad (2.157)$$

ce qui s'écrit :

$$\forall l \in L, (K \otimes S)(r_l) = \int_{V_l} K_l(l, l') s(l') dl', \quad (2.158)$$

où  $K_l(l, l') = K_l(l' - l)$  est le pixon en  $l$  et  $V_l$  le support de  $K_l$ .

Différents choix sont bien entendu possibles pour la forme des pixons. Nous considérons des pixons radialement symétriques [MHK<sup>+</sup>96] définis par des paraboloïdes tronqués :

$$K_l(l' - l) = K(l' - l, d_l) = \begin{cases} \frac{1}{C_l} \left(1 - \frac{\|l' - l\|^2}{d_l^2}\right) & \text{if } \|l' - l\| \leq d_l \\ 0 & \text{if } \|l' - l\| > d_l \end{cases} \quad (2.159)$$

où  $C_l$  est un facteur de normalisation tel que  $\|K_l\| = 1$ .

Un pixon est donc complètement défini par sa taille  $d$ . En pratique, comme nous traitons de données discrètes, les différentes tailles admissibles appartiennent à l'ensemble  $\{0, 1, \dots, d_{max}\}$ .

Dans le PI-modèle, le terme d'attache aux données est remplacé par :

$$E_{DPI} = \sum_{l \in L} \frac{\left\| x_l - \int_{V_l} K_l(l, l') s(l') dl' \right\|^2}{2\sigma^2} \quad (2.160)$$

L'a priori d'information sur la carte de segmentation est, par ailleurs, conservée. L'optimisation s'effectue alternativement sur  $K$  et  $S$ . L'image restaurée  $R$  est ensuite obtenue par l'équation 2.157.

**Le MPI-modèle-** Pour compléter le modèle, nous introduisons maintenant des contraintes de régularisation sur les cartes de segmentation et de pixons. Pour régulariser la segmentation, nous considérons un modèle de Potts. L'a priori sur la carte de segmentation, incluant le terme d'information ayant pour but de réduire le nombre de classes de la description, s'écrit comme suit :

$$P(S) = \frac{1}{Z} \exp - \left( -\alpha_I \sum_{i=1}^n \frac{N_i}{N} \ln \frac{N_i}{N} + \beta \sum_{c=\{l,m\} \in \mathcal{C}} \delta_{s(l) \neq s(m)} \right), \quad (2.161)$$

où  $Z$  est la fonction de partition.

La carte des pixons apporte une description multi-échelle de la scène qui correspond à la résolution locale de l'image. Cette carte doit vérifier une certaine contrainte d'homogénéité mais également représenter les structures fines tels que les contours. L'a priori choisi pour modéliser cette carte est donc le chien-modèle.

L'énergie globale du MPI-modèle s'écrit donc comme suit :

$$E_{MPI} = E_{GOF}(X, K, S) + E_I(S) + E_M^{(1)}(S) + E_M^{(2)}(K), \quad (2.162)$$

où :

$$E_{GOF}(X, K, S) = \frac{1}{2\sigma^2} \sum_{l \in L} \left\| x_l - \int_{V_l} K_l(l, m) s(m) dm \right\|^2 \quad (2.163)$$

$$E_I(S) = -\alpha_I \sum_{i=1}^n \frac{N_i}{N} \ln \frac{N_i}{N} \quad (2.164)$$

$$E_M^{(1)}(S) = \beta \sum_{c=\{l, m\} \in \mathcal{C}} \delta_{s(l) \neq s(m)} \quad (2.165)$$

$$E_M^{(2)}(K) = \sum_{c \in \mathcal{C}_K} \sum_l V_c^{(b)}(i_l) \quad (2.166)$$

Le maximum a posteriori correspond au maximum de la probabilité conditionnelle  $P(K, S|X)$ . Nous avons donc une optimisation sur deux images. Ceci est obtenu par un recuit simulé sur l'espace produit  $\mathcal{K} \times \mathcal{S}$ .

Nous avons constaté que les dynamiques associées aux a priori markoviens ont une vitesse de convergence plus rapide que celle associée à l'a priori d'information. Durant l'optimisation, les termes de régularisation ont tendance à former de petits "clusters" de plusieurs pixels qui sont difficilement supprimés avec une dynamique pixélique. Le terme d'information ne peut donc pas jouer son rôle. Pour pallier ce problème, nous effectuons en premier lieu un certain nombre d'itérations en n'utilisant que le PI-modèle. Puis, nous incluons les a priori markoviens lorsque le terme d'information a suffisamment réduit l'espace des états de la carte de segmentation.

**Analyse d'une carte statistique-** L'application qui a motivé ce travail est l'analyse des cartes de paramètres statistiques pour les études en IRM fonctionnelle.

Nous avons appliqué l'approche MPI pour analyser une SPM. En premier lieu, l'image restaurée permet de supprimer les fausses alarmes et d'accroître le rapport signal à bruit. De plus, l'image segmentée donne une description des régions contenant de l'information et de celles correspondant au fond. Ainsi, nous évitons le seuillage dans l'analyse de la SPM et analysons l'information de la SPM sur les modes de l'image segmentée qui sont spatialement délimités par la carte de segmentation.

Nous comparons, dans un premier temps, les résultats obtenus avec respectivement les modèles I, PI et MPI. La figure 2.79 montre une SPM obtenue à partir d'une expérience traitant du langage ainsi que les images restaurées, les cartes représentant la taille des pixons et les segmentations. Notons que pour le I-modèle, la carte des pixons n'est pas définie, de sorte que restauration et segmentation sont identiques. La description obtenue par le I-modèle se réduit à une simple classification de l'image effectuée par l'algorithme présentée dans le paragraphe 2.2.1. Le PI-modèle permet de reconstruire l'image en fonction de la classification. Le MPI-modèle permet de régulariser les images obtenues, ce qui permet de mieux distinguer le bruit de l'information et résulte en une meilleure reconstruction de l'image.

L'approche a également été validée sur des IRMf échoplanaires (EPI), de meilleure résolution temporelle. La figure 2.80.a montre la carte d'un t-test obtenue pour différentes coupes, le résultat d'un seuillage superposé aux coupes anatomiques est montré sur la figure 2.80.b, l'image restaurée par l'approche MPI sur la figure 2.80.c et le seuillage de cette restauration sur la figure 2.80.d. Pour finir, la superposition de ce résultat sur les coupes anatomiques est montrée sur la figure 2.80.e. L'expérience consistait ici à demander aux sujets un jugement grammatical à partir de phrases présentées auditivement. On trouve des zones activées dans l'hémisphère droit (partie gauche de l'image) correspondant au cortex auditif primaire (et sur certaines veines). En particulier, l'activation thalamique de l'hémisphère gauche est clairement écrite sur la figure 2.80.e alors qu'elle n'est pas apparente sur les images simplement seuillées (figure 2.80.b). A posteriori, l'activation thalamique pourrait être détectée sur les images non restaurées mais serait difficilement jugée significative. Ce résultat a été confirmé sur douze sujets

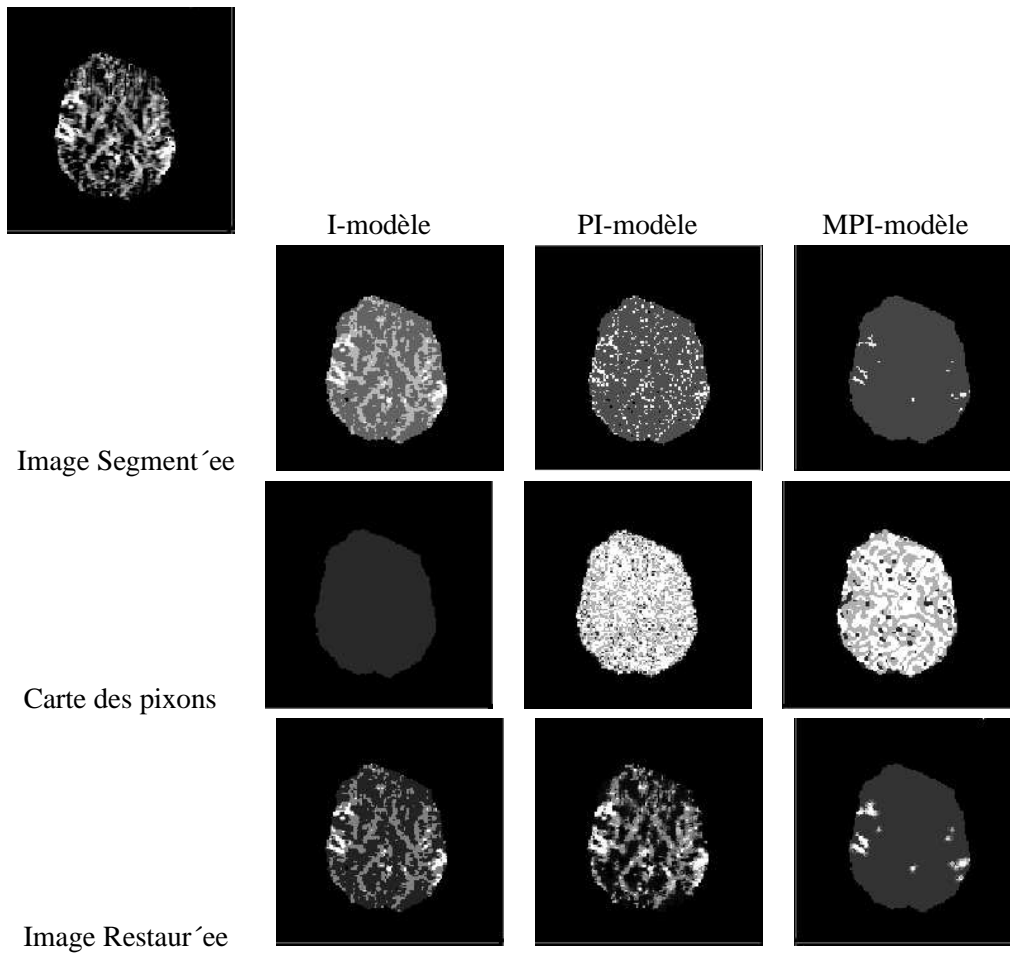


FIG. 2.79 – Carte statistique (en haut à gauche) et cartes de segmentation, de pixon et de restauration pour les modèles I,PI et MPI.

différents. L'approche MPI permet donc d'extraire plus d'information de la carte statistique.

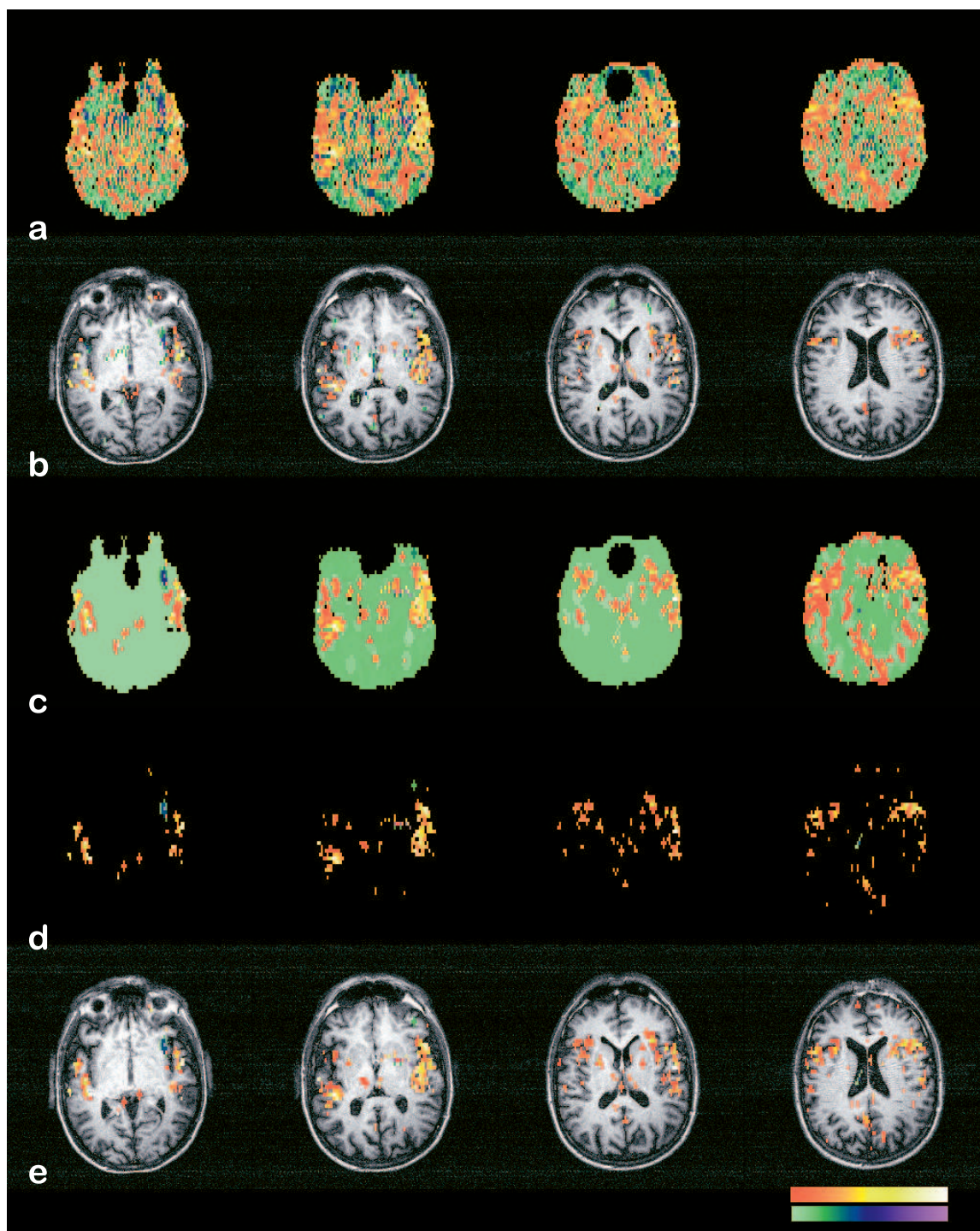


FIG. 2.80 – restauration par le MPI-modèle sur une séquence EPI (carte statistique (t-test) originale (a), carte statistique seuillée ( $th = 1.0$ ) (b), carte statistique restaurée (c) et carte statistique restaurée puis seuillée ( $th = 1.0$ ) (d et e). L'échelle rouge-jaune (resp. vert-bleu) correspond aux valeurs positives (resp. négatives).



### 2.3.4 Champs de Markov sur graphes

#### Mise en correspondance de réseaux routiers

<sup>18</sup> Nous abordons ici le problème de la mise en correspondance de deux réseaux linéiques. Ces réseaux sont supposés représenter une même réalité mais ils sont issus de deux sources de données exogènes (carte et image SPOT dans notre cas). Nous supposons que les caractéristiques des réseaux sont données par les différents points de branchement (angles entre les branches d'un même point et distance entre deux points de branchements) et les points de changement d'orientation. Le réseau est donc ici symbolisé par un ensemble de segments. Nous adoptons une représentation par graphe pour laquelle chaque nœud représente un segment et chaque arête symbolise la connexité entre deux segments. L'exogénité des données nous a amené à utiliser le segment comme unité de représentation car le tracé linéique peut ne pas être fiable. La mise en correspondance des deux réseaux est donc effectuée par la mise en correspondance des deux graphes.

La mise en correspondance de graphes a été abordée sous plusieurs angles suivant les contraintes du problème traité. Étant données deux ensembles  $C_1$  et  $C_2$  de primitives, la mise en correspondance consiste à déterminer la partie de  $C_1 \times C_2$  qui satisfait au mieux un certain nombre de critères. La plupart des problèmes abordés en traitement d'image permettent des restrictions pour la recherche du meilleur ensemble d'appariements (un appariement étant un couple  $(c_1, c_2)$ , avec  $c_1 \in C_1$  et  $c_2 \in C_2$ ). La principale contrainte rencontrée est l'unicité. En effet, une primitive ne peut être appariée qu'une seule fois. La mise en correspondance revient alors à trouver un isomorphisme entre les deux ensembles de primitives. Cependant, dans notre cas, nous considérons la possibilité de données incomplètes et/ou erronées dans les réseaux.

De façon générale, l'ensemble des appariements retenus doit vérifier deux propriétés. Premièrement, deux primitives appariées doivent disposer de caractéristiques similaires (ressemblance mesurée par une distance entre les primitives). Ensuite, l'ensemble des appariements doit être globalement cohérent (cohérence des angles et des longueurs dans notre cas).

Nous avons orienté notre modèle de mise en correspondance sur l'invariance de quantités géométriques caractéristiques d'un réseau routier.

Pour traiter le cas des données manquantes et la dissymétrie du problème, nous nous sommes orientés vers la définition d'une fonctionnelle définissant l'étiquetage d'un graphe (graphe de départ) sur un second (graphe d'arrivée). Cette fonctionnelle intègre des contraintes *a priori* sur le résultat et un terme d'attache aux données pour fonder l'étiquetage sur la cohérence des deux réseaux. Le formalisme markovien nous permet une souplesse de modélisation et l'utilisation d'un algorithme de recuit simulé pour minimiser la fonctionnelle.

L'approche adoptée est donc une approche bayésienne fondée sur les champs de Markov. Les variables aléatoires sont constituées des nœuds du graphe de départ et l'espace des états des nœuds du graphe d'arrivée.

#### Mise en correspondance de réseaux linéiques : un modèle markovien invariant par déplacements-

Nous considérons deux réseaux linéiques définis par un ensemble de segments. Ces réseaux sont représentés par des graphes, à chaque segment est associé un nœud dont les attributs sont la longueur et l'orientation du segment. Un arc est présent entre deux nœuds lorsque les segments correspondants sont connexes. La mise en correspondance est effectuée en considérant les nœuds du graphe de départ  $D = \{d\}$  comme des variables aléatoires prenant leurs valeurs dans l'ensemble des étiquettes définies par les nœuds du graphe d'arrivée  $A = \{g\}$ . Une étiquette nulle, connectée avec aucune autre, est ajoutée au graphe d'arrivée pour permettre le non appariement des nœuds du graphe de départ (données manquantes dans

<sup>18</sup> Nous résumons ici le travail de DEA de C. Hivernat financé par Acatel CRC en collaboration avec S. Randriamasy. Bibliographie : [14, 44, 43]

le graphe d'arrivée) [WH97]. Notons que les données manquantes du graphe de départ sont représentées par les étiquettes non affectées du graphe d'arrivée.

Nous considérons un champ markovien sur le graphe constitué par l'ensemble de départ. Nous pouvons donc écrire la distribution sous la forme d'une distribution de Gibbs :

$$P(d_i, i = 1, \dots, n_d) = \frac{1}{Z} \exp -U(d_i, i = 1, \dots, n_d) = \frac{1}{Z} \exp - \sum_{c \in \mathcal{C}} V_c(d_i, i \in c) \quad (2.167)$$

où  $Z$  est la fonction de partition,  $U$  la fonction d'énergie et  $V_c$  sont les fonctions potentiel.  $\mathcal{C}$  est l'ensemble des cliques  $c$ .

Pour définir les potentiels, nous considérons une notion de voisinage sur les deux graphes :

**Définition 1.** Deux nœuds sont voisins si la distance entre les deux extrémités les plus proches des segments associés est inférieure à un seuil  $dV$  (en pratique  $dV=6$  pixels pour notre application).

Le choix de  $dV \neq 1$  permet de s'affranchir de certaines discontinuités issues de l'extraction des lignes et de mieux définir la structure géométrique du réseau.

On note  $a_d = a_{d'}$  si les étiquettes  $q_d$  et  $a_{d'}$  des nœuds du graphe de départ sont égales,  $a_d \simeq a_{d'}$  si elles sont voisines au sens de la définition 1 et  $q_d \neq a_{d'}$  si elles ne sont pas voisines. Dans ces trois cas, on considère que  $a_d$  et  $a_{d'}$  sont différents de l'étiquette  $\emptyset$ .

Les potentiels sont définis comme suit :

$$V_c^{(1)}(a_d, a_{d'}) = \begin{cases} \alpha_1 \times g(\theta(d, d') - \pi) & \text{si } a_s = a_{d'} \\ \alpha_2 \times g(|\theta(d, d') - \theta_0(a_d, a_{d'})|) & \text{si } a_d \simeq a_{d'} \\ \alpha_3 & \text{si } a_d \neq a_{d'} \\ \alpha_4 & \text{si } a_d = \emptyset \text{ ou } a_{d'} = \emptyset \end{cases}$$

où  $\theta(d, d')$  est l'angle entre les deux segments  $d$  et  $d'$  du graphe de départ et  $\theta(a_d, a_{d'})$  est l'angle entre les deux segments  $a_d$  et  $a_{d'}$  du graphe d'arrivée. Ces définitions d'angle ne sont valables qu'entre des primitives voisines au sens de la définition 1. Les angles sont considérées entre les segments orientés en considérant leur point commun comme origine.

La fonction  $g$  (cf. Figure 2.81), permettant d'introduire la contrainte d'angle dans les potentiels, doit vérifier les propriétés suivantes :

- définie sur  $[-2\pi, 2\pi]$ ,
- convexe,
- décroissante sur  $[-2\pi, 0]$  et croissante sur  $[0, 2\pi]$  car on veut favoriser les angles tels que  $\theta(d, d) = \pi$  si les étiquettes sont identiques et  $\theta(d, d) = \theta_0(a_d, a_{d'})$  pour des étiquettes voisines.

La rapidité de la décroissance va permettre de fixer le seuil de tolérance entre les angles. En effet, une décroissance lente vers 0 favorise un plus large intervalle, alors qu'une décroissance rapide permet de mieux focaliser sur l'angle nul et donc d'être plus sélectif dans le choix des appariements. En revanche, prendre une fonction trop sélective revient à éliminer toute variation entre les deux réseaux et donc ne permet pas de tenir compte de la variabilité des données de type exogène.

Il faut remarquer que les angles associés aux primitives ne sont fiables que si la longueur de ces primitives est suffisante. Nous avons choisi de ne prendre en compte les angles que si la longueur des primitives est supérieure à un seuil, en l'occurrence  $dV-2$ . Dans le cas où l'angle n'est pas valide, les

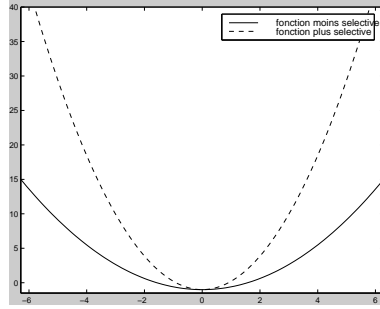


FIG. 2.81 – Fonctions  $g$  plus ou moins sélectives ( $g(x) = 4 \times \frac{x^2}{\pi^2} - 1$  et  $g(x) = 12 \times \frac{x^2}{\pi^2} - 1$ )

potentiels sont :

$$V_c^{(2)}(a_d, a_{d'}) = \begin{cases} -\alpha_1 & \text{si } a_d = a_{d'} \text{ et si } d \text{ et } d' \text{ ne sont pas s'éparées par un point de bifurcation du réseau} \\ \alpha_2 & \text{si } a_d \simeq a_{d'} \text{ et si } \theta(d, d') \text{ est non valide} \\ \alpha_3 & \text{si } a_d \simeq a_{d'} \text{ et si } \theta_0(a_d, a_{d'}) \text{ est non valide} \\ 0 & \text{sinon} \end{cases}$$

On complète ce modèle avec un potentiel défini sur un voisinage d'ordre 2. Il a pour objectif d'éviter des configurations entraînant une rupture de continuité pour une étiquette. Soient  $d, d', d''$  tels que  $d \simeq d'$  et  $d' \simeq d''$  :

$$V_c^{(3)}(a_d, a_{d'}, a_{d''}) = \begin{cases} \beta & \text{si } a_{d'} = a_{d''} \text{ et } a_d \neq a_{d'} \\ 0 & \text{sinon} \end{cases}$$

Pour compléter le modèle et le rendre plus robuste, on doit tenir compte de la longueur des segments apparus. En effet, la longueur totale prise par une étiquette sur le graphe de départ doit être proche de sa longueur sur le graphe d'arrivée. Nous supposons donc ici que les deux réseaux sont à la même échelle. Le potentiel tenant compte de cette contrainte n'est pas markovien car il dépend de la configuration de toutes les primitives et pas seulement de la configuration sur un voisinage, en revanche nous conservons le formalisme des champs de Gibbs.

Notons  $l(d)$  la longueur de la primitive  $d$  du graphe de départ.  $l(a)$  est la longueur du segment du graphe d'arrivée correspondant à l'étiquette  $a$ . L'énergie issue de ce terme est :

$$V^{(4)} = \sum_{a(\text{étiquettes})} \left[ \sum_{d(\text{sites})} l(d) \delta_{a_d=a} - l_a \right]^2$$

L'énergie totale s'écrit donc :

$$U = \sum_{(d, d') : d \simeq d', \max(l(d), l(d')) > dV - 2} V_c^{(1)} + \sum_{(d, d') : d \simeq d', \max(l(d), l(d')) \leq dV - 2} V_c^{(2)} + \sum_{(d, d', d'') : d \simeq d', d' \simeq d''} V_c^{(3)} + V^{(4)}. \quad (2.168)$$

Cette énergie est minimisée par un recuit simulé. Le choix des paramètres  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$  et  $\beta$  a été fait empiriquement et est discuté par la suite.

**Application aux réseaux routiers d'un couple image SPOT/carte :** Dans ce modèle de mise en correspondance de linéiques, le choix des primitives utilisées a été guidé par la nature des sources de données. La carte étant généralisée [Mus98], il est naturel de choisir des segments pour la représenter. En revanche, le réseau routier extrait de l'image est constitué de pixels. Un des objectifs de ce travail

est d'améliorer la précision des données cartographiques. Pour conserver la précision de l'image, nous évitons de polygonaliser le réseau extrait. Pour obtenir un étiquetage robuste, il est préférable de mettre en correspondance des entités de même nature, c'est-à-dire 1D dans le cas présent. Nous effectuons donc un premier étiquetage des pixels dont le but est de segmenter le réseau extrait de l'image en chaînes de pixels, qui seront à leur tour étiquetées par le modèle que nous venons de décrire. Le but de cette première étape est d'effectuer un découpage en chaînes du réseau de l'image qui soit cohérent avec les données cartographiques (cf figure 2.3.4). Comme précédemment un nœud nul est ajoutée au réseau cartographique pour modéliser les données manquantes.

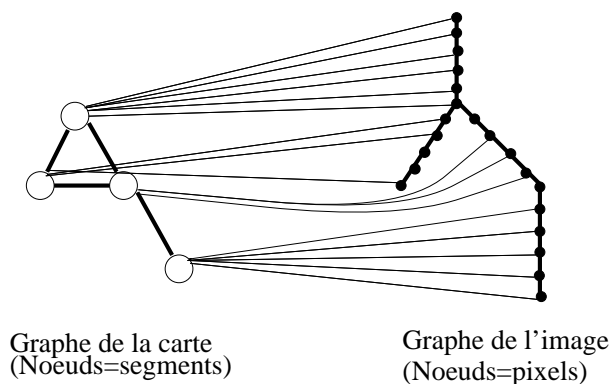


FIG. 2.82 – Graphes des deux sources et mise en correspondance pixels-segments

Un potentiel de type attache aux données directement liée aux données du problème est défini. Il est en quelque sorte une mesure de cohérence d'un appariement. Il est défini par la distance entre les deux nœuds appariés, soit la distance entre un point et un segment. Ce potentiel est une constante  $C$  pour toute primitive appariée à l'étiquette nulle  $\emptyset$  ce qui a pour effet de borner le terme d'attache aux données. En effet, tout pixel se trouvant à une distance  $d > C$  de tous les segments aura tendance à être apparié à l'étiquette nulle.

Des potentiels d'interaction permettent d'introduire des contraintes et de maintenir l'homogénéité des appariements. On les définit sur les cliques formées de deux nœuds voisins (pixels connexes) par la fonction suivante :

$$V_c(a_s, a_{s'}) = \begin{cases} 0 & \text{si } a_s = a_{s'} \\ \alpha_1 & \text{si } a_s \simeq a_{s'} \\ \alpha_2 & \text{sinon} \end{cases}$$

où  $a_s$  (resp.  $a_{s'}$ ) représente l'étiquette au site  $s$  (resp.  $s'$ ).

Les termes  $\alpha_1$  et  $\alpha_2$  sont déterminés en fonction du degré de pénalité que l'on veut imposer aux configurations correspondantes. Leurs valeurs, telles que  $0 < \alpha_1 < \alpha_2$ , sont données manuellement. L'énergie, somme de tous les potentiels, est minimisée par un algorithme de recuit simulé.

Cette première étape nous permet de segmenter le réseau routier issu de l'image tout en tenant compte des données cartographiques, ce qui représente un avantage majeur par rapport à une polygonalisation aveugle du réseau. Un mauvais recalage initial pourra induire des erreurs d'étiquetage mais nous ne gardons que la segmentation du réseau lors de cette étape, l'étiquetage proprement dit étant effectué à partir des chaînes.

Le résultat de la mise en correspondance par les méthodes détaillées précédemment présente des cas pathologiques dus soit aux données (mauvaise détection des routes dans l'image ou base de données

cartographique incomplète), soit à la mise en correspondance (imprécision des données cartographiques due à la généralisation, mauvais recalage,...). Nous proposons de détecter ces cas automatiquement et de les interpréter. Les appariements validés vont nous permettre d'estimer une matrice de recalage et d'itérer ainsi le processus.

A ce stade, une mise en correspondance a été effectuée. Elle a permis d'associer des primitives de l'image à celles de la carte. Pour utiliser ces résultats, il faut les transférer sur des entités significatives du réseau routier. Sur un tel réseau, les points importants sont les carrefours qui déterminent les extrémités des tronçons de route. Il est donc important d'utiliser les résultats précédents afin de déterminer les appariements des tronçons (définis par plusieurs segments/chaînes) et de les valider. Les appariements obtenus ne sont pas tous biunivoques, c'est-à-dire qu'à chaque élément d'un ensemble ne correspond pas toujours un seul élément de l'autre ensemble.

Soit  $N_I$  et  $N_C$  le nombre de tronçons contenus respectivement dans l'image  $\mathcal{J}$  et dans la carte  $\mathcal{C}$ . Les résultats de la mise en correspondance peuvent être représentés par un ensemble :

$$\mathcal{M}(\mathcal{J}, \mathcal{C}) = \{M(i, c), i \in [0, N_I] \text{ et } c \in [0, N_C]\}$$

Les cas types pouvant se présenter sont :

- M(0,1) : tronçon de la carte non apparié,
- M(1,0) : tronçon de l'image non apparié,
- M(1,1) : un tronçon de  $\mathcal{J}$  apparié à un tronçon de  $\mathcal{C}$ , (ce cas remplit une condition nécessaire à la validation de l'appariement mais il n'est pas suffisant)
- M(1,c) : tronçon de  $\mathcal{J}$  apparié à plusieurs tronçons de  $\mathcal{C}$ ,
- M(i,1) : tronçon de  $\mathcal{C}$  apparié à plusieurs tronçons de  $\mathcal{J}$ ,
- M(i,c) : i tronçons de  $\mathcal{J}$  appariés à c tronçons de  $\mathcal{C}$ ,

Toutes les configurations différentes de M(1,1) sont considérées comme pathologiques. Les ambiguïtés doivent être levées avant de valider les appariements biunivoques M(1,1). Nous supprimons, en premier lieu, les ambiguïtés engendrées par les cas M(1,c), M(i,1) et M(i,c). Puis, nous étudions la validité des cas biunivoques M(1,1) pour traiter, enfin, les cas de non appariement M(0,1) et M(1,0).

Un couple de primitives,  $(S_i, S_c)$   $S_i \in \mathcal{J}$  et  $S_c \in \mathcal{C}$ , peut être décrit par des attributs  $A(S_i, S_c)$  qui vont permettre une mesure de la cohérence de l'appariement. Grâce à l'étude de ces attributs, la levée des ambiguïtés pourra être effectuée ainsi que la validation des appariements biunivoques.

- **Distance entre deux tronçons** : Soit  $N(i)$  le nombre de points du tronçon  $S_i$ . Soit  $d(p_n^i, S_c)$ , la distance mesurée entre un point  $p_n^i$ , appartenant à  $S_i$ , et le tronçon cartographique  $S_c$ , c'est-à-dire le minimum des distances entre  $p_n^i$  et les différents segments qui composent le tronçon. La distance entre le tronçon de l'image  $S_i$  et le tronçon cartographique  $S_c$  est donnée par :

$$D(S_i, S_c) = \frac{1}{N(i)} \sum_{n=1}^{N(i)} P(n) \times d(p_n^i, S_c)$$

Le terme  $P(n)$  permet de pondérer les distances de manière à favoriser les extrémités du tronçon, et autoriser une plus grande déviation pour les points intérieurs au tronçon. En effet, les carrefours de la carte sont censés être définis avec plus de précision que la forme des routes elle-même. Le choix de ce terme est déterminé par le poids relatif que l'on veut donner aux points intérieurs par rapport aux extrémités. On peut prendre :

$$P(n) = \frac{\text{MAX}(n, N(i) - n) - N(i)/2}{N(i)}$$

- **Longueur relative** : La longueur relative entre la longueur du tronçon de l'image,  $l(S_i)$ , et celle du tronçon de la carte,  $l(S_c)$ , est aussi un critère de validité de l'appariement. Il est moins fiable que le critère de distance car il dépend fortement de la qualité de la généralisation des données cartographiques. Pour faciliter la comparaison entre appariements, il est préférable de prendre un rapport toujours inférieur à 1 :

$$R_l(S_i, S_c) = \frac{MIN(l(S_i), l(S_c))}{MAX(l(S_i), l(S_c))}$$

- **Longueur relative appariée** : La longueur relative appariée représente le taux de points appariés au tronçon cartographique considéré par rapport à la longueur totale du tronçon de l'image :

$$R_{M_c}(S_i) = \frac{l_{M_c}(S_i)}{l(S_i)}$$

$l_{M_c}(S_i)$  représente la longueur du tronçon apparié, c'est-à-dire le nombre de points portant l'étiquette correspondante au tronçon cartographique considéré.

- **Critère de comparaison des appariements** : Utiliser les attributs séparément peut conduire à favoriser des appariements erronés. Il faut donc plutôt utiliser un critère de comparaison composé des différents attributs :

$$C(S_i, S_c) = D(S_i, S_c) \times (1 - R_l(S_i, S_c) \times R_{M_c}(S_i))$$

L'appariement  $(S_i, S_c)$  sera d'autant meilleur que  $C(S_i, S_c)$  est proche de 1.

Les ambiguïtés correspondent aux cas où les tronçons de l'image sont multi-étiquetés, ou aux cas où les tronçons de la carte sont appariés plusieurs fois. Dans chaque cas, on ne conservera que l'appariement le plus probant afin d'obtenir un résultat ne comportant que des appariements biunivoques.

**Cas M(1,c)** : Cette configuration correspond au cas où un tronçon  $t_i$  de  $\mathcal{J}$  est apparié à plusieurs tronçons cartographiques. Pour les différentes hypothèses d'appariement correspondant aux tronçons cartographiques mis en correspondance avec  $t_i$ , nous calculons la valeur du critère donné au paragraphe 2.3.4. L'appariement retenu est celui qui minimise le critère.

**Cas M(i,1)** : Après avoir résolu le problème des tronçons de  $\mathcal{J}$  multi-étiquetés, on peut aboutir à une mise en correspondance où les tronçons de la carte sont appariés à plusieurs tronçons de l'image. Le critère de comparaison permet de ne conserver que le tronçon  $\mathcal{J}$  le plus approprié.

**Cas M(i,c)** : Pour traiter ce cas, nous décomposons la configuration en  $i$  appariements de type M(1,c) que l'on traite comme précédemment. Les cas de type M(i,1) qui peuvent en résulter sont ensuite également traités comme précédemment.

Nous nous sommes ramenés à des appariements biunivoques. Ces appariements seront validés s'ils ont des valeurs suffisantes sur les attributs,  $A(S_i, S_c)$ .

Il faut donc au préalable déterminer un seuil de valeurs acceptables pour  $A(S_i, S_c)$ . Pour chaque appariement non valide, la connaissance d'un (ou des) attribut(s) non valide(s) est utile à l'interprétation des résultats. L'utilisateur est maître des seuils qui seront définis en fonction des expériences. Un deuxième jeu de seuils pourrait être introduit afin de trier les appariements en trois classes : appariements valides, appariements non fiables et appariements faux.

Pour nos tests, la qualification des résultats de la mise en correspondance a été effectuée grâce aux seuils suivants :

- seuil de distance entre paires : 50

- seuil pour le rapport entre les longueurs : 0.7
- seuil pour le rapport relatif des longueurs apparées : 0.9

L'amélioration du recalage permet d'obtenir un meilleur appariement pixel-segment lors de la première étape. La définition des sites du second modèle est alors plus cohérente avec la structure du réseau cartographique et les performances de la mise en correspondance finale s'en trouvent nettement améliorées. Nous effectuons le recalage par une transformation appliquée aux coordonnées des primitives de la carte. Les coordonnées sont mises sous la forme d'un vecteur  $z = (x, y, 1)^T$  et la transformation est définie par une matrice  $3 \times 3$  en géométrie affine. Une transformation affine est définie par six paramètres indépendants. Elle est composée d'une translation et d'une homothétie. Elle permet donc la correction du facteur d'échelle ainsi que la correction des positions (rotation et translation). Dans ce modèle de recalage, on ne tient pas compte des modifications entraînées par le relief. En coordonnées homogènes, la matrice de recalage s'écrit donc :

$$\Phi = \begin{pmatrix} \Phi_{1,1} & \Phi_{1,2} & \Phi_{1,3} \\ \Phi_{2,1} & \Phi_{2,2} & \Phi_{2,3} \\ 0 & 0 & 1 \end{pmatrix}$$

Les coordonnées résultant de cette transformation sont données par  $z' = \Phi z$ .

Les appariements jugés valides par les critères de qualification servent de points d'appui pour estimer la matrice de recalage  $\Phi$ . Nous minimisons le terme de distance moyenne entre les appariements valides qui sont au nombre de  $N_{val}$ . Ce terme est défini par :

$$\overline{D}_{val}((J, \mathcal{C}), \Phi) = \frac{1}{N_{val}} \sum_{(S_i, S_c) \text{ valides}} D(S_i, S_c)$$

Cette fonction dépend de la matrice de recalage qui intervient dans le calcul de la distance  $D(S_i, S_c)$  entre le tronçon de route de l'image satellitaire et le tronçon cartographique qui lui est apparié. La matrice de recalage estimée est celle qui vérifie :

$$arg\{\min(\overline{D}_{val}((J, \mathcal{C}), \Phi))\}$$

En fait, la matrice à déterminer peut être considérée comme une inconnue dans un espace à 6 dimensions. La minimisation est effectuée avec la méthode de Powell qui permet de traiter un problème multidimensionnel. Le point de départ est la matrice de recalage approchée fournie avec les données cartographiques.

La mise en correspondance, fondée sur les angles, perd de sa fiabilité lorsque les segments sont trop courts. Pour éviter ce cas lors de la première étape d'étiquetage des pixels "routes" de l'image, il convient d'avoir un recalage approximatif entre les deux jeux de données.

Les appariements valides permettent de déterminer un meilleur recalage. Après avoir amélioré l'équation entre les repères des deux jeux de données, il est possible de refaire une étape de mise en correspondance afin d'améliorer les résultats.

**Résultats-** Sur la figure 2.83, nous avons simulé les données cartographiques en polygonalisant fortement le réseau extrait de l'image, puis en le translatant de 10 pixels. Sur la figure 2.83.b se trouve le découpage en chaînes du réseau extrait de l'image, après l'étiquetage des pixels. La première mise en correspondance est montrée sur la figure 2.83.c. Un certain nombre d'erreurs apparaît, notamment dans le quadrant nord-est qui correspond à l'endroit le plus dense du réseau. Ces erreurs sont automatiquement détectées lors de la phase de qualification des résultats comme le montre la figure 2.83.d. Lors de cette première étape de qualification, uniquement utilisée pour le calcul de la matrice de recalage, nous sommes sévères sur le critère de validation, ce qui explique qu'un certain nombre d'appariements corrects ne sont pas validés. Les appariements validés permettent de recalibrer les deux réseaux. Le résultat

de la mise en correspondance après recalage est présentée sur la figure 2.83.e. Seule une chaîne de deux pixels est mal appariée. Pour cette chaîne, très courte, la valeur de l'angle est peu fiable. Cette erreur est bien détectée lors de la qualification (voir la figure 2.83.f). Une seconde chaîne, reflétant un manque de précision des données cartographique, est détectée.

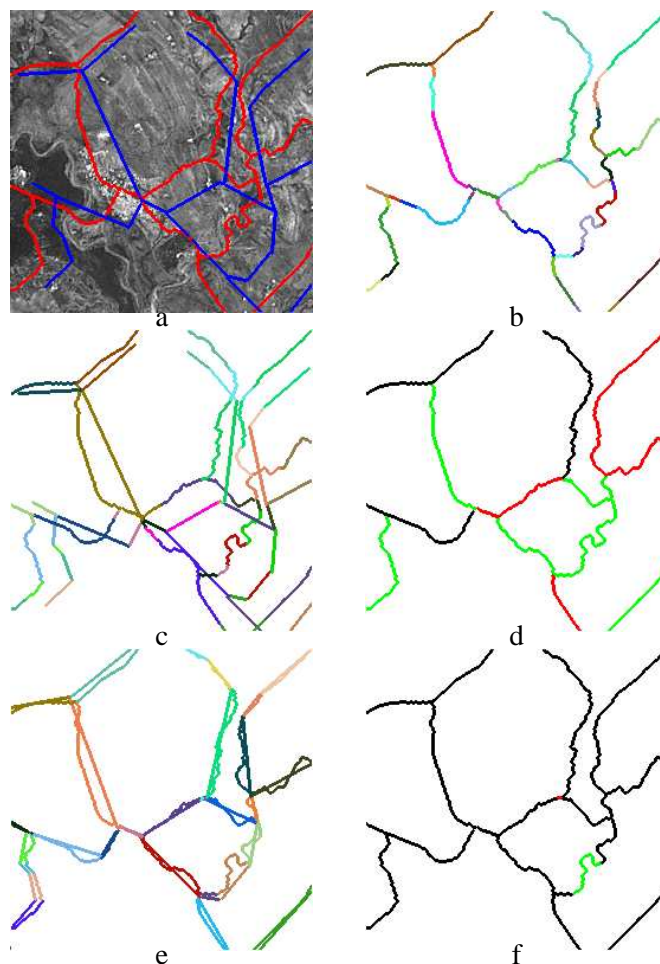


FIG. 2.83 – Résultat pour un réseau cartographique simulé- a: données de l'image (en rouge) et cartographiques (en bleu), b: graphe de départ (après étiquetage des pixels), c: première mise en correspondance, d: qualification des résultats (seules les chaînes en noir sont jugées bien appariées), e: les deux réseaux après recalage des données cartographiques sur l'image, f: résultat de la qualification (la chaîne en rouge est non appariée, la chaîne en vert est un appariement non validé)

Un cas réel est traité sur la figure 2.84. Les réseaux cartographique et extrait de l'image SPOT sont respectivement en vert et en bleu. Le réseau cartographique une fois recalé est présenté rouge et jaune. Les tronçons bien appariés sont en bleu foncés et en rouge. Un segment dans la zone urbaine au centre de l'image est présent dans la carte mais n'a pas trouvé de correspondant sur l'image (en jaune sur la figure 2.84). Ceci s'explique par la résolution de l'image SPOT (10m) qui rend les algorithmes d'extraction de routes peu fiables dans les zones urbaines. Dans le quadrant sud-est les routes extraites de l'image, absentes des données cartographiques, ont bien été appariées à l'étiquette nulle (en bleu ciel sur la figure 2.84). Notons toutefois qu'une mise à jour automatique n'est pas encore faisable car le tracé de la route devient éroné dans la zone urbaine.





FIG. 2.84 – Résultat sur données réelles : le réseau extrait de l'image est en bleu, le réseau cartographique initial est en vert, le réseau cartographique après recalage est en rouge, les routes de l'image non appariées sont en bleu ciel et la route de la carte non appariée en jaune.

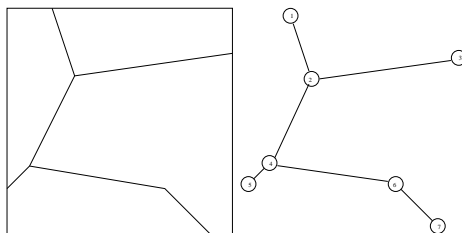


FIG. 2.85 – Un réseau cartographique et le graphe correspondant

### Recalage non rigide d'un réseau routier

<sup>19</sup> Le but de ce travail est de projeter, par un processus de recalage et de distorsion (ou recalage non rigide), un réseau routier extrait d'une base de données cartographique sur une image satellitaire. Un prérequis est donc le recalage du réseau cartographique avec l'image, car la distorsion du réseau pour coller à l'image ne peut être précise sans une pré-localisation correcte. Ce problème est classiquement traité par une mise en correspondance des structures linéiques extraites de l'image avec les items cartographiques (voir le paragraphe précédent). En modélisant les items linéiques comme les noeuds d'un graphe dont les arêtes reflètent la connexité du réseau, nous considérons le problème comme une mise en correspondance de graphes.

**Modélisation des données-** Nous avons deux types de données : une image et un réseau vectorisé cartographique. Il nous faut clairement définir les variables aléatoires considérées (sites) et leurs valeurs (états). L'image satellite (SPOT Panchromatique) est considérée comme référence pour le recalage. Le réseau cartographique est modélisé par un graphe non-orienté  $G = \{X, U, A\}$ ,  $X$  étant l'ensemble des noeuds,  $U$  l'ensemble des arêtes et  $A$  l'ensemble des attributs associés aux noeuds. Les noeuds sont des points caractéristiques du réseau (jonctions, points de forte courbure) et les arêtes définissent les routes qui les joignent. Le seul attribut d'un noeud est sa position dans l'image. Un modèle markovien est construit sur ces noeuds, l'état d'un noeud étant sa position. Les routes sur l'image entre les noeuds sont calculées par un processus déterministe. Le graphe est construit en utilisant l'information de forme (nombre de noeuds et liens entre eux) et la position des données cartographiques. Un recalage local est effectué en ajustant la position des noeuds et en distordant les liens par un algorithme de détection de routes déterministe. Ce modèle permet de prendre en compte le phénomène de généralisation qui fait que le tracé cartographique entre les noeuds est approximé.

La figure 2.85 donne un exemple de la correspondance d'un réseau routier cartographique et de son modèle de graphe (les noeuds sont représentés par des cercles).

Nous considérons un graphe  $X$  de structure identique à celui défini par le réseau cartographique. La position des noeuds représente les variables aléatoires. En d'autres termes, le réseau recherché  $x$  est obtenu en déplaçant les noeuds des données initiales  $m$ .

Le champ  $X$  suit une loi a priori  $P(X = x)$  (notée  $P(x)$  par la suite). Les données sont modélisées comme suit :

- les données cartographiques  $M$  sont un graphe similaire à  $X$ , i.e., un graphe ayant la même "structure" (même nombre de noeuds et d'arêtes), seuls les attributs des noeuds diffèrent,
- l'image  $I$  est une image SPOT, naturellement représentée par un tableau de pixels.

Le champ  $X$  est vu comme une déformation du champ  $M$ , ces deux champs sont composées de variables aléatoires interagissant entre elles. L'image  $I$  paramétrise le modèle via les probabilités définies.

<sup>19</sup>. Nous résumons ici le travail de DEA de G. Rellier. Bibliographie : [65, 66]

Cela signifie que le modèle a priori, le modèle a posteriori et la vraisemblance d'épendent du paramètre  $I$  et sont notées respectivement  $P(x)$ ,  $P_I(x|m)$  et  $P_I(m|x)$  (ou  $P(x)$ ,  $P(x|m)$  et  $P(m|x)$  pour simplifier).

En utilisant le théorème d'Hammersley-Clifford nous pouvons écrire :

$$P(m|x) = \frac{\exp(-U_m(x))}{Z_m} \text{ et } P(x) = \frac{\exp(-U_{ap}(x))}{Z_{ap}} \text{ avec}$$

$$U_m(x) = \sum_{c \in C} U_m^c(x)$$

et

$$U_{ap}(x) = \sum_{c \in C} U_{ap}^c(x)$$

où  $C$  représente les cliques.

Nous considérons le critère bayésien du maximum a posteriori (MAP) :

$$\hat{x} = \arg \min_x (U_m(x) + U_{ap}(x))$$

Le terme d'épendant de la carte,  $U_m(x)$ , reflète la compatibilité des graphes  $x$  et  $m$ . Brièvement, une configuration de  $x$  a une énergie d'autant plus faible que les tronçons de  $x$  sont proches de leur correspondant dans  $m$ , et que les connections entre tronçons sont similaires. Cela conduit à deux types de contraintes :

- la première contrainte opère sur une paire de noeuds reliés par un tronçon de route. Elle d'épend de la distance entre les deux noeuds, qui doit être proche de celle entre les noeuds correspondant de la carte,
- La seconde contrainte opère sur un triplet de noeuds successifs. Elle d'épend de l'angle formée par ces trois noeuds, qui doit être proche de l'angle correspondant dans les données cartographiques.

En réalité, la transformation idéale de la carte vers le réseau obtenue  $x$  est une isométrie et, plus elle s'en écarte, plus elle est pénalisée. Ceci se fonde sur l'hypothèse d'une position relative précise des noeuds des données cartographiques. Pour traduire ces deux contraintes, nous définissons une énergie d'épendant de la carte comme suit. Nous définissons une distance associée aux graphes :

**Définition 1.** *Un chemin entre  $x_i$  et  $x_j$  est une ensemble de noeuds ( $x_i = x_0, x_1, \dots, x_N = x_j$ ) pour lequel deux noeuds consécutifs  $x_k$  et  $x_{k+1}$  sont joints par une arête.*

**Définition 2 (Distance de graphes).** *La distance entre deux noeuds d'un graphe  $X$ ,  $x_i$  et  $x_j$ , notée  $d(x_i, x_j)$ , est égale au nombre minimal d'arêtes des chemins ( $x_i, x_j$ ) (égal au nombre de noeuds moins un).*

Le terme de l'énergie d'épendant de la carte est donné par :

$$U_m(x) = \sum_{d(x_i, x_j)=1} U_m^1(x_i, x_j) + \sum_{\substack{d(x_i, x_j)=1 \\ d(x_j, x_k)=1}} U_m^2(x_i, x_j, x_k)$$

C'est un terme markovien relativement au système de voisinage suivant :

$$x_j \in V_{x_i} \text{ si et seulement si } \begin{cases} x_i \neq x_j \\ d(x_i, x_j) \leq 2 \end{cases}$$

Les potentiels  $U_m^1(x_i, x_j)$  et  $U_m^2(x_i, x_j, x_k)$  représentent respectivement les deux contraintes mentionnées plus haut.



FIG. 2.86 – A droite, la route joignant deux extrémités obtenue par ELIESER ; A gauche, l'image SPOT correspondante.

$$U_m^1(x_i, x_j) = f_{ij}(\text{dist}(x_i, x_j))$$

où  $\text{dist}(a, b)$  est la distance euclidienne entre les noeuds  $a$  et  $b$ , et :

$$U_m^2(x_i, x_j, x_k) = g_{i,j,k}(\alpha_{i,j,k})$$

où  $\alpha_{i,j,k}$  est l'angle formé par les deux segments  $[x_i, x_j]$  et  $[x_j, x_k]$ .

La fonction  $f_{ij}$ , dont l'argument est la distance euclidienne entre deux noeuds, est une fonction convexe prenant son minimum lorsque  $\text{dist}(x_i, x_j) = \text{dist}(x_i^m, x_j^m)$  (où  $\text{dist}(x_i^m, x_j^m)$  est la distance euclidienne des noeuds correspondant dans les données cartographiques).

La fonction  $g_{i,j,k}$  a pour argument la valeur de l'angle formé par le triplet considéré. Elle est définie sur l'intervalle  $[0, 2\pi]$ , et atteint son minimum pour  $\alpha_{i,j,k} = \alpha_{i,j,k}^m$  (où  $\alpha_{i,j,k}^m$  est l'angle correspondant sur la carte), et croît avec la différence entre  $\alpha_{i,j,k}$  et  $\alpha_{i,j,k}^m$ .

Pour l'expérimentation, nous avons considéré  $f_{ij}$  et  $g_{i,j,k}$  comme suit :

$$f_{ij}(d) = (d - \text{dist}(x_i^m, x_j^m))^2$$

$$g_{ijk}(\alpha) = ((\alpha - \alpha_{i,j,k}^m) [2\pi])^2$$

Les configurations optimales relativement à ce premier terme sont donc celles obtenues par une transformation rigide des données cartographiques. L'énergie croît en fonction de l'importance des déformations locales.

L'énergie  $U_{ap}(x)$  prend en compte l'information donnée par l'image  $I$ . Elle est composée de deux termes. Les arêtes du graphe représentent les routes entre les noeuds, elles sont constituées de segments dans les données cartographiques. Nous devons les déformer pour coller à l'image. Nous utilisons un algorithme fondé sur la programmation dynamique pour estimer le parcours de la route sur l'image entre deux noeuds. Ce chemin ne dépend donc que de la position des noeuds sur l'image et des niveaux de gris. Pour le calculer nous minimisons une fonction de coût dépendant du contraste entre le chemin et le fond de l'image, de l'homogénéité des pixels appartenant à la route et de la courbure locale. La minimisation est effectuée par le programme ELIESER [MZ96]. Le chemin ainsi obtenu de manière déterministe est utilisé pour définir l'énergie a priori du modèle.

L'énergie a priori dépend de l'adéquation du réseau ainsi calculée, à partir de la configuration  $x$  du graphe, avec un réseau routier réel. Cette adéquation est mesurée par un terme géométrique et par un second terme quantifiant la cohérence avec les données image. Nous avons l'expression suivante :

$$U_{ap}(x) = U_{ap,1}(x) + U_{ap,2}(x)$$



FIG. 2.87 – Deux configurations possibles pour deux paires de noeuds : influence de la position des noeuds sur les chemins calculés (en noir) qui peuvent se superposer si les noeuds sont trop proches (à droite)

Le premier terme, exprimant les contraintes géométriques, évite que deux routes joignant deux paires distinctes de noeuds se superposent, même partiellement. Ce terme de “non-chevauchement” est nécessaire car certaines routes très prononcées, peuvent attirer plusieurs sections routières lors de l’optimisation par programmation dynamique. L’énergie associée est égale au nombre de pixels partagés par deux sections routières différentes.

L’expression de  $U_{ap,1}(x)$  est donnée par :

$$U_{ap,1}(x) = \sum_{s_1, s_2 \in S} U_{ap,1}^{s_1, s_2}(x) = \sum_{\substack{d(x_i, x_j) = 1 \\ d(x_k, x_l) = 1}} U_{ap,1}(x_i, x_j, x_k, x_l)$$

où  $s_1 = (x_i, x_j)$  est la section routière joignant  $x_i$  et  $x_j$ ,  $s_2 = (x_k, x_l)$  la section routière joignant  $x_k$  et  $x_l$ , le potentiel étant :

$$U_{ap,1}^{s_i, s_j}(x) = n_{ij}$$

où  $n_{ij}$  est le nombre de pixels commun à  $s_i$  et  $s_j$ .

La figure 2.87 montre un exemple où le premier terme va pénaliser le fait que deux routes se superposent (b) alors qu’il favorisera la configuration (a).

La seconde partie de l’énergie a priori  $U_{ap,2}^z(x)$  est directement liée à l’image. Elle donne une mesure, pour chaque chemin entre deux noeuds estimée par ELIESER, de ressemblance entre les niveaux de gris du chemin et de son voisinage et les caractéristiques d’une route. En considérant un système de voisinage défini par les paires de noeuds joints par une arête, nous avons :

$$U_{ap,2}(x) = \sum_{d_{ij}=1} U_{ap,2}(x_i, x_j) = \sum_{s \in S} U_{ap,2}^s(x)$$

où  $S$  est l’ensemble des sections routières  $s$  et  $d_{ij}$  la distance entre les noeuds  $x_i$  et  $x_j$ .

Chaque terme  $U_{ap,2}^s(x)$  exprime la probabilité pour la section considérée  $s$  d’être située sur la position courante sachant les données image. Une route est définie comme une structure linéaire ayant un fort contraste par rapport au fond. Nous définissons trois zones : une zone “route”, une zone “fond” et une zone intermédiaire (permettant de traiter différentes épaisseurs de route), comme montré sur la figure 2.88. Nous calculons la moyenne des zones “route” et “fond”. Les pixels de la zone intermédiaire sont classés comme “route” ou “fond” en accord avec leur écart aux moyennes “route” et “fond”. Un test d’hypothèse est calculé entre les zones obtenues. Nous utilisons un test de Student :

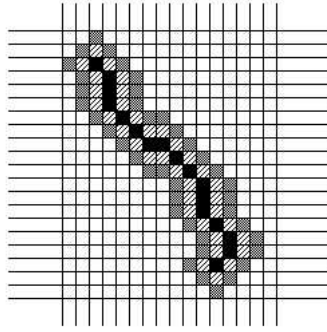


FIG. 2.88 – Les trois zones définies pour le test : la route (en noir), le fond (en gris) et la zone intermédiaire (hachurée).

$$t\_test(D_1, D_2) = \frac{\mu_1 - \mu_2}{\sqrt{\left(\frac{N_1\sigma_1^2 + N_2\sigma_2^2}{N_1 + N_2 - 2}\right)\left(\frac{1}{N_1} + \frac{1}{N_2}\right)}}$$

Une fonction décroissante sur la valeur du test donne le potentiel dépendant de l'image :

$$U_{ap,2}^s(x) = \Psi(t\_test(D_1, D_2))$$

où  $D_1$  et  $D_2$  sont respectivement les distributions de pixels du “fond” et de la “route” et  $\Psi(x) = \lambda/x$ . L'utilisation de ce test est plus robuste que la fonction de coût utilisée par ELIESER car cette dernière dépend fortement du nombre de pixels de la route.

Finalement, l'expression de l'énergie a priori est donnée par :

$$U_{ap}(x) = \sum_{s_i, s_j \in S} U_{ap,1}^{s_i, s_j}(x) + \sum_{s \in S} U_{ap,2}^s(x)$$

où  $S$  est l'ensemble des sections routières, c'est à dire l'ensemble des arêtes du graphe.

**Résultats-** L'optimisation du modèle se fait par un recuit simulé. Pour chaque noeud nous proposons une nouvelle localisation par une dynamique de Metropolis. Le schéma de décroissance de la température est donné par  $T^{(n+1)} = 0.97 * T^{(n)}$ , où  $n$  est le numéro de l'itération.

La figure 2.89 montre le résultat pour une carte initialement translatée de 20 pixels par rapport aux données cartographiques réelles. Le résultat est celui escompté malgré un décalage initial important.

La figure 2.90 montre le résultat avec une configuration initiale ayant subi une rotation de 0.2 radian par rapport aux données initiales. Les noeuds ont été localisés à la bonne place. De petites erreurs apparaissent dans la partie supérieure droite en raison d'un faible rapport signal à bruit.

Pour finir, nous introduisons un bruit sur les distances entre noeuds et sur les angles en translatant aléatoirement chaque noeud du graphe initial dans une fenêtre de 20 pixels par 20 centrée en chacun des noeuds. Nous pouvons apprécier la qualité du résultat dans ce cas, où la solution n'est pas déduite d'une transformation rigide du graphe.

Ce deuxième exemple montre l'utilisation des champs de Markov et du contexte bayésien pour la résolution de problèmes sur graphes. Cela montre notamment que l'approche markovienne ne se réduit pas aux modélisations pixeliques et permet de prendre en compte des entités géométriquement plus informatives, comme les segments dans cette application. La restriction de cette modélisation réside dans le caractère figé du graphe puisque le nombre de noeuds et d'arêtes ne peuvent pas évoluer au cours

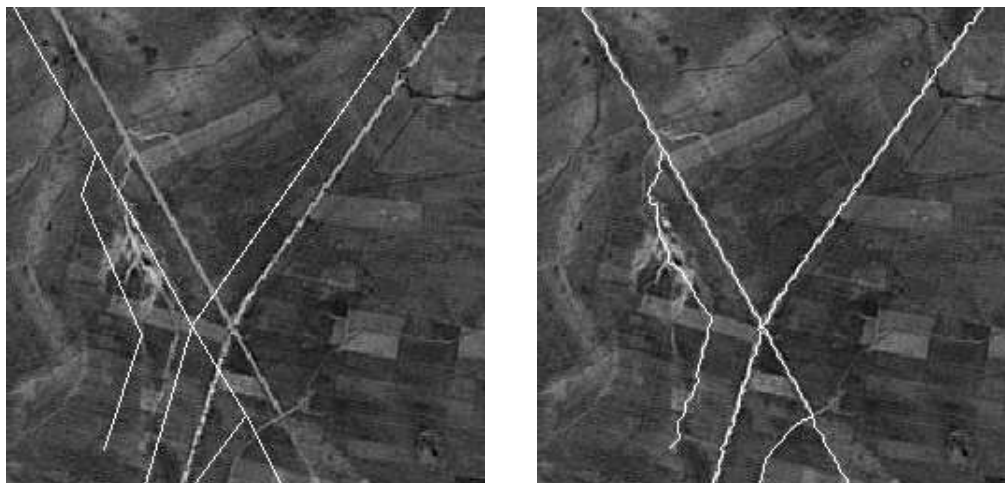


FIG. 2.89 – Exemple de résultat (à droite) obtenu à partir de la carte translaturée (à gauche)

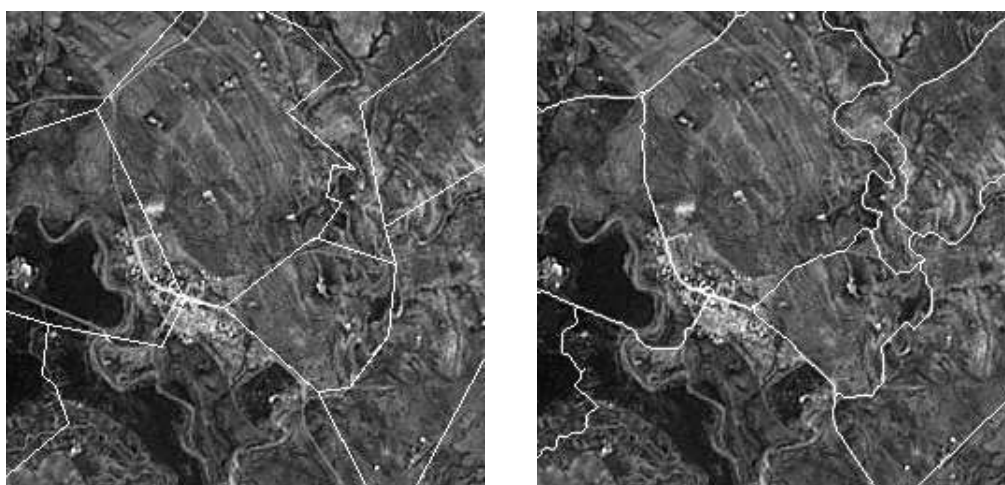


FIG. 2.90 – Exemple de résultat (à droite) obtenu à partir de la carte ayant subi une rotation (à gauche)



FIG. 2.91 – Exemple de résultat (à droite) obtenu à partir de la carte ayant subi une distorsion (à gauche)

de l'optimisation. Ainsi, un tel modèle ne prend pas en compte l'évolution du réseau, lors de la création de nouvelles routes par exemple.



### 2.3.5 Shape from Shading

<sup>20</sup> Le “Shape from shading” (SFS), ou forme à partir de l’ombrage, consiste à reconstruire la forme 3D d’une scène par l’analyse des niveaux de gris d’une seule image bidimensionnelle de cette scène. Sous certaines conditions, le niveau de gris donne directement la pente de la forme, la difficulté consistant alors à trouver la direction de plus forte pente. Ce problème est mal posé et ne peut être résolu sans certaines hypothèses [Dur93, DM96].

Les hypothèses usuelles [HB89] qui rendent le problème bien posé concernent :

- la scène qui est supposée avoir une carte de réflectance uniforme et connue (la plupart du temps la surface est supposée lambertienne et d’albedo connu),
- la source lumineuse qui est supposée unique et ponctuelle, éloignée de la scène de sorte que le faisceau incident puisse être approximé par un faisceau parallèle uniforme  $\vec{S}$  (la plupart du temps les réflexions sont négligées),
- la caméra qui est supposée donner une projection orthogonale de la scène sur le plan de l’image et avoir une réponse linéaire.

Sous ces hypothèses, en notant l’axe optique  $Oz$  ( $z$  croît en s’approchant de la caméra) complétée des axes  $Ox$  et  $Oy$  définissant le plan de l’image de sorte que  $Oxyz$  forme une base orthogonale, la partie visible de la scène peut être décrite par l’équation :

$$z = h(x, y) \quad (2.169)$$

où l’inconnue est la fonction d’élévation  $h$ , qui *a priori* n’est pas dérivable (dans le cas de bords) et même non continue (dans le cas d’occlusions). En chaque point  $(x, y)$  où  $h$  est dérivable, l’équation du SFS est l’équation d’irradiance [Hor75] :

$$R(\vec{S}, p(x, y), q(x, y)) = E(x, y) \quad (2.170)$$

où  $E$  désigne le niveau de gris,  $R$  la carte de réflectance, et  $p$  et  $q$  sont les notations usuelles pour  $\partial h / \partial x$  et  $\partial h / \partial y$ . Dans cette équation, seuls  $\vec{S}$  et  $h$ , par l’intermédiaire de ses dérivées premières, sont inconnus.

**Energies-** Nous supposons maintenant que  $\vec{S}$  est connu et vertical ( $\vec{S} = (0 \ 0 \ 1)^t$ ) et que la surface est lambertienne, de sorte que l’équation d’irradiance (2.170) s’écrit sous une forme particulière appelée “équation de l’Eikonale” :

$$\frac{E_{\max}}{\sqrt{1 + p(x, y)^2 + q(x, y)^2}} = E(x, y) \quad (2.171)$$

où  $E_{\max}$  est la valeur maximale de  $E$  qui est atteinte pour les points où  $\nabla h = 0$ . La version discrète de l’équation (2.171) est donnée par :

$$\frac{E_{\max}}{\sqrt{1 + p_{i,j}^2 + q_{i,j}^2}} = E_{i,j} \quad (2.172)$$

Sur les images de synthèse utilisées pour les tests, les niveaux de gris sont calculés à partir de l’équation (2.172), avec  $E_{\max} = 255$ .

Nous introduisons la distance  $\delta$  entre pixels voisins. Nous considérons arbitrairement que chaque scène carrée utilisée à une longueur de 12.8, de sorte que  $\delta = 0.1$  pour une image  $128 \times 128$ ,  $\delta = 0.2$  pour une image  $64 \times 64$ , etc... En utilisant un schéma direct de différences finies, la contrainte d’intégrabilité de Schwarz s’écrit :

$$\frac{p_{i,j+1} - p_{i,j}}{\delta} = \frac{q_{i+1,j} - q_{i,j}}{\delta} \quad (2.173)$$

---

<sup>20</sup> Ce travail est le fruit d’une collaboration avec J.D. Durou et A. Crouzil, maîtres de conférence à l’IRIT, initialement soutenue par un projet Jeunes Chercheurs du CNRS. Bibliographie : [4]

La norme de la dérivée en  $x$  s'écrit :

$$|\nabla p|_{i,j}^2 \approx \left( \frac{p_{i+1,j} - p_{i,j}}{\delta} \right)^2 + \left( \frac{p_{i,j+1} - p_{i,j}}{\delta} \right)^2 \quad (2.174)$$

et de même pour  $q$ , de sorte que l'énergie minimisée s'écrit :

$$\begin{aligned} \epsilon(\omega) = & \delta^2 \sum_{(i,j) \in D} \left[ \frac{E_{\max}}{\sqrt{1 + p_{i,j}^2 + q_{i,j}^2}} - E_{i,j} \right]^2 \\ & + \lambda_{\text{int}} \sum_{(i,j) \in \tilde{D}} [(p_{i,j+1} - p_{i,j}) - (q_{i+1,j} - q_{i,j})]^2 \\ & + \lambda_{\text{smo}} \sum_{(i,j) \in \tilde{D}} [(p_{i,j+1} - p_{i,j})^2 + (p_{i+1,j} - p_{i,j})^2 + (q_{i,j+1} - q_{i,j})^2 + (q_{i+1,j} - q_{i,j})^2] \end{aligned} \quad (2.175)$$

où  $\omega = (p_{i,j}, q_{i,j})_{(i,j) \in D}$  et  $\tilde{D}$  est le sous-ensemble de  $D$  contenant les pixels  $(i,j)$  tels que  $(i+1,j)$  et  $(i,j+1)$  sont dans  $D$ . L'énergie ainsi définie contient le terme d'attache aux données provenant de l'équation de l'Eikonale, un terme d'intégrabilité et un terme de lissage. La dépendance vis-à-vis de la résolution est réduite du fait de l'introduction du terme  $\delta$ , ce qui sera utile pour l'approche multi-résolution décrite plus bas. Nous traitons ici de la minimisation de l'énergie  $\epsilon$ , les résultats finaux sont obtenus par une phase d'intégration des dérivées en minimisant l'énergie suivante [Dan00] :

$$\epsilon'(\omega') = \sum_{(i,j) \in \tilde{D}} [(h_{i+1,j} - h_{i,j} - \delta p_{i,j})^2 + (h_{i,j+1} - h_{i,j} - \delta q_{i,j})^2] \quad (2.176)$$

où  $\omega' = (h_{i,j})_{(i,j) \in D}$ .

**Méthode déterministe : M1-** Soit une image, un ensemble de pixels  $D$ , des valeurs des paramètres  $\lambda_{\text{int}}$  et  $\lambda_{\text{smo}}$ , et une configuration initiale  $\omega^0 = (p_{i,j}^0, q_{i,j}^0)_{(i,j) \in D}$  ; la première étape du traitement de la méthode **M1** est un processus itératif défini comme suit :

- Calculer  $\nabla \epsilon$  pour la configuration courante  $\omega^k$
- Trouver le minimum local  $d^k$  de la fonction  $\phi_k(d) = \epsilon(\omega^k - d \nabla \epsilon(\omega^k))$
- Calculer la nouvelle configuration  $\omega^{k+1} = \omega^k - d^k \nabla \epsilon(\omega^k)$

Le processus est stoppé dès qu'une configuration  $\omega^*$  est telle que  $|\nabla \epsilon(\omega^*)| < 2N\beta$  (le seuil  $\beta$  est égal à 0.001 dans nos tests). Le facteur  $2N$  provient du fait que la norme euclidienne de  $\nabla \epsilon$  est approximativement proportionnelle au nombre de coordonnées.

A partir de la configuration obtenue  $\omega^*$ , la seconde étape consiste à minimiser l'énergie par un processus identique au précédent.

La surface test est montrée sur la figure 2.92(a), et l'image associée  $64 \times 64$  sur la figure 2.92(b). Notons que l'interprétation visuelle 3D de cette image est quasiment impossible. A partir d'une initialisation proche de la forme réelle, représentée sur la figure 2.93(a), nous obtenons la figure 2.93(b), ce qui n'est pas tout à fait satisfaisant, car le creux au centre est inversé. A partir de la configuration de l'image 2.94(a), le résultat obtenu sur la figure 2.94(b) est même médiocre d'un point de vue qualitatif. En revanche pour  $\lambda_{\text{int}} = 500$  et  $\lambda_{\text{smo}} = 20$ , les résultats obtenus à partir des figures 2.93(a) et 2.94(a) sont montrés sur les figures 2.95(a) et 2.95(b). Il est clair que le résultat de la figure 2.95(a) est très bon alors même que celui de la figure 2.95(b) est très mauvais. La méthode **M1** a donc deux inconvénients :

- Le choix de  $\lambda_{\text{int}}$  et  $\lambda_{\text{smo}}$  est relativement arbitraire et a un effet important sur la solution (comparons les figures 2.93(b) et 2.95(a)).
- Le choix de la forme initiale est capital (comparons les figures 2.95(a) et 2.95(b)).

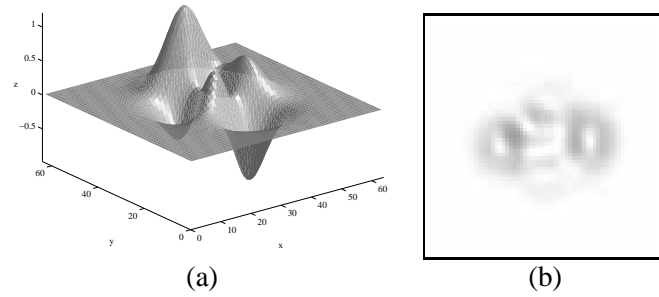
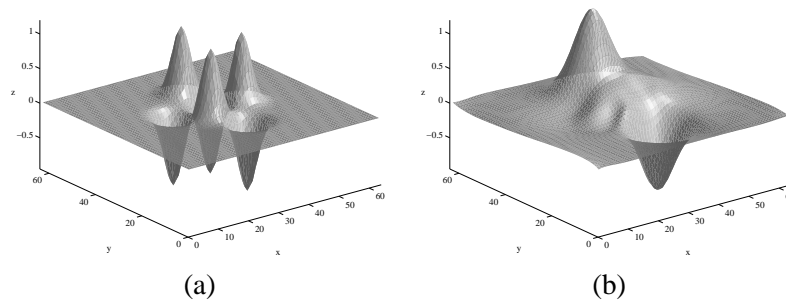


FIG. 2.92 – (a) surface test (b) image associée.

FIG. 2.93 – (a) forme initiale (b) résultat associé ( $\lambda_{int} = 10$ ,  $\lambda_{smo} = 50$ ).

Pour résoudre le second problème, nous proposons d'optimiser l'énergie par un recuit simulé.

**Une approche Bayésienne du “shape from shading” : méthode M2-** Le problème du SFS appartient à la classe des problèmes mal-posés qui peuvent être reformulés comme un problème d'optimisation dans un contexte bayésien. Nous adoptons cette démarche et proposons le recuit simulé comme outil d'optimisation. Nous interprétons l'énergie  $\epsilon$  comme celle d'un champ de Gibbs. L'interprétation bayésienne associée induit quatre termes  $P(E|\omega)P_1(\omega)P_2(\omega)P_3(\omega)$ . Le premier, prenant en compte les données s'interprète comme la vraisemblance et les trois suivants définissent le terme a priori contenant une contrainte de lissage, une contrainte d'intégrabilité et un a priori sur la distribution des pentes  $(p, q)$ .

L'énergie  $\epsilon$ , définie par une somme de termes locaux, est associée à la définition d'un champ de Gibbs de la manière suivante :

$$P(E|\omega)P_1(\omega)P_2(\omega) = \frac{1}{Z} \exp[-\epsilon(\omega)] \quad (2.177)$$

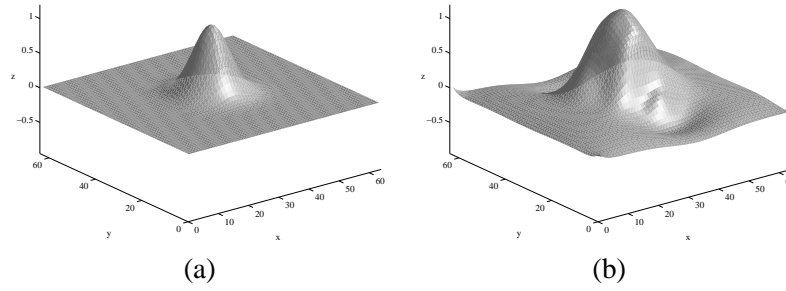
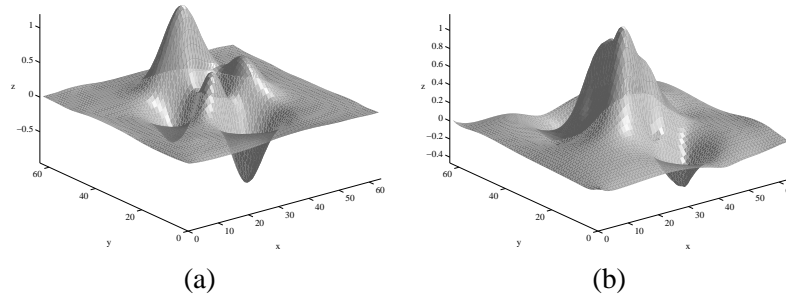
Un a priori est défini sur  $(p, q)$  en supposant les sites indépendants (par rapport à ce terme) :

$$P_3(\omega) = \prod_{s \in S} \pi_3(\omega_s) \quad (2.178)$$

Pour définir  $\pi_3(\omega_s)$ , nous supposons que la normale  $\vec{N}$  à la surface est uniformément distribuée sur la sphère de Gauss.

Changeons le système de coordonnées comme suit :

$$\begin{cases} p_s = \rho_s \cos \phi_s \\ q_s = \rho_s \sin \phi_s \end{cases} \quad (2.179)$$

FIG. 2.94 – (a) Seconde forme initiale et (b) résultat associé ( $\lambda_{int} = 10$ ,  $\lambda_{smo} = 50$ ).FIG. 2.95 – Avec  $\lambda_{int} = 500$  et  $\lambda_{smo} = 20$ : résultat à partir de 2.93(a) (a) et de 2.94(a) (b).

Sur la sphère de Gauss, la surface de la couronne  $(\theta, \theta + d\theta)$  (voir figure 2.96) est égale à  $2\pi \sin \theta d\theta$ . Nous considérons alors les distributions  $\pi_\theta(\theta) \propto \sin \theta$  and  $\pi_\phi(\phi)$  uniformes sur  $[0, 2\pi[$  pour définir l'a priori sur  $(p, q)$ . Nous avons  $\rho = \tan \theta$ , ce qui donne  $\sin \theta d\theta = \frac{\rho}{(1+\rho^2)^{3/2}} d\rho$ . La distribution a priori sur  $\rho$  est donc donnée par :

$$\pi_\rho(\rho) \propto \frac{\rho}{(1+\rho^2)^{\frac{3}{2}}} \quad (2.180)$$

En pratique, l'espace d'état associé à  $\rho$  est borné à  $[0, 2\rho_{\max}]$ . La valeur de  $\rho_{\max}$  est calculée en appliquant l'équation de l'Eikonale (2.172) aux données ( $\rho_{\max}$  correspond au niveau de gris le plus faible). Nous obtenons :

$$\pi_\rho(\rho) = \frac{\frac{\rho}{(1+\rho^2)^{\frac{3}{2}}}}{\int_0^{2\rho_{\max}} \frac{\rho}{(1+\rho^2)^{\frac{3}{2}}} d\rho}, \quad \rho \in [0, 2\rho_{\max}] \quad (2.181)$$

Pour réduire le temps de calcul, il convient de trouver un bon compromis entre le nombre d'itérations nécessaire à la convergence et le temps de calcul pour simuler la loi de proposition. La seconde contrainte justifie l'utilisation d'une loi de proposition similaire au modèle simulé (cas de l'échantillonneur de Gibbs). Cependant, en présence d'interactions une telle loi est très difficilement simulable, Nous considérons donc uniquement le terme sans interaction du modèle a priori comme loi de proposition. La loi utilisée dans le recuit simulé est donc la suivante :

$$Q(\omega_s^k = cur \rightarrow \omega_s^{k+1} = new) = Q(\omega_s^{k+1} = new) = \pi_3(new) \quad (2.182)$$

Notons  $\omega \setminus \{s\} = (\omega_1^{new}, \dots, \omega_{s-1}^{new}, \omega_{s+1}^{new}, \dots, \omega_{card(S)}^{new}) = (\omega_1^{cur}, \dots, \omega_{s-1}^{cur}, \omega_{s+1}^{cur}, \dots, \omega_{card(S)}^{cur})$ . Le taux d'acceptation est alors donné par :

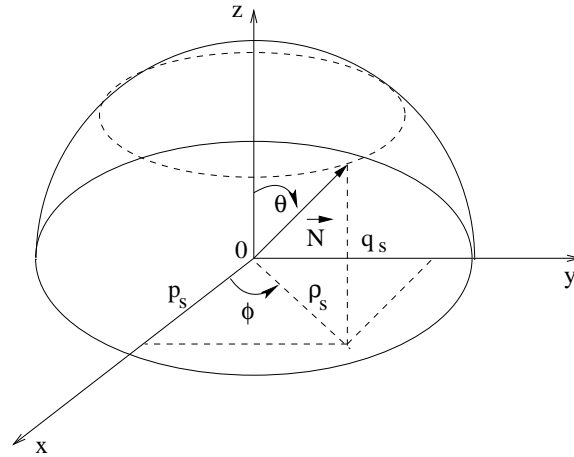


FIG. 2.96 – La sphère de Gauss.

$$R = \left( \frac{\rho_{new}(1 + \rho_{cur}^2)^{3/2}}{\rho_{cur}(1 + \rho_{new}^2)^{3/2}} \right)^{1/T-1} \left( \frac{p_e(i_s|\omega_s^{new})P_1(\omega_s^{new}|\omega\setminus\{s\})P_2(\omega_s^{new}|\omega\setminus\{s\})}{p_e(i_s|\omega_s^{cur})P_1(\omega_s^{cur}|\omega\setminus\{s\})P_2(\omega_s^{cur}|\omega\setminus\{s\})} \right)^{1/T} \quad (2.183)$$

où  $\rho_{new} = \sqrt{(p_s^{new})^2 + (p_s^{new})^2}$  et  $\rho_{cur} = \sqrt{(p_s^{cur})^2 + (p_s^{cur})^2}$ .

Nous utilisons une décroissance géométrique de la température  $\propto \alpha^k T_0$ . Notons qu'en général, pour des applications de type segmentation ou restauration d'image, les taux de décroissance retenus pour  $\alpha$  se trouvent entre 0.95 et 0.99. Ici, nous avons dû considérer  $\alpha = 0.999998$  pour obtenir le résultat de la figure 2.97(b) (avec  $\alpha = 0.99$  nous obtenons le résultat de la figure 2.97(a)). Ceci est vraisemblablement dû aux profonds minima locaux de l'énergie associée au SFS. Notons qu'ici la configuration initiale peut être quelconque (nous sommes partis d'un plan pour nos expériences). Le temps de calcul est ici d'une heure pour une image  $32 \times 32$ . D'un point de vue applicatif, il semble donc nécessaire de le réduire.

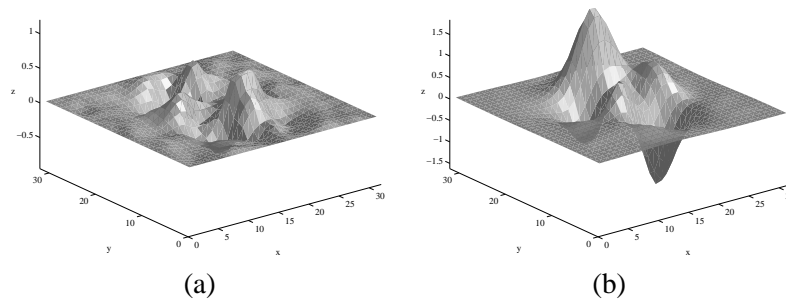


FIG. 2.97 – **M2** appliquée à l'image 2.92(b): (a)  $\alpha = 0.99$  et (b)  $\alpha = 0.999998$  ( $\lambda_{int} = 500$  et  $\lambda_{smo} = 20$ ).

**Une approche hybride multirésolution : méthode M3-** Nous combinons l'approche stochastique à l'approche déterministe pour réduire le temps de calcul. Le principe consiste à utiliser la méthode optimale (le recuit simulé) à basse résolution, puis à utiliser le résultat comme initialisation pour les

résolutions suivantes. Dans le cas du SFS, la relation entre les données (l'image) et les inconnues (la forme) n'est pas linéaire. Par conséquent, les réductions des résolutions de la forme et de l'image ne produisent pas les mêmes effets. Nous utilisons donc un algorithme multirésolution non-linéaire, qui approxime au mieux la réduction de la résolution de la forme. Peleg et Ron [SG90] ont montré qu'il est préférable de dégrader la résolution des pentes pour approximer une dégradation de la résolution de la forme plutôt que de dégrader directement l'image.

Si  $E^0$  représente l'image initiale (de résolution la plus fine) et  $E^l$  l'image de taille  $2n \times 2n$  au niveau  $l$ , l'image moins résolue  $E^{l+1}$  est calculée comme suit :

- Calculer les pentes  $\rho_{i,j}^l$  à partir des niveaux de gris :

$$\rho_{i,j}^l = \sqrt{\frac{E_{\max}^2}{E_{i,j}^l} - 1}, \quad (i,j) \in [0, 2n - 1]^2 \quad (2.184)$$

- Convolver les pentes par un noyau gaussien.
- Sous-échantillonner l'image des pentes en ne conservant qu'un pixel sur deux pour obtenir  $\beta^{l+1}$  de taille  $n \times n$ .
- Calculer l'image correspondante en utilisant la formule suivante :

$$E_{i,j}^{l+1} = \frac{E_{\max}}{\sqrt{1 + (\rho_{i,j}^{l+1})^2}}, \quad (i,j) \in [0, n - 1]^2 \quad (2.185)$$

Après avoir construit la pyramide, un premier résultat est obtenu au niveau le moins résolu par un recuit simulé. Ce résultat est sur-échantillonné par interpolation pour obtenir une initialisation au niveau suivant. La méthode M1 permet d'obtenir le résultat à ce niveau. Ce processus est itéré jusqu'au niveau de plus haute résolution. Cette approche suppose que la projection du résultat d'un niveau au niveau suivant appartient au bassin d'attraction du minimum global de l'énergie.

Pour nos expériences, nous avons utilisé une interpolation linéaire pour la projection des résultats de  $(p,q)$  entre les niveaux et le masque de convolution suivant pour les pentes lors de la construction de la pyramide :

$$\begin{bmatrix} 0 & 0.125 & 0 \\ 0.125 & 0.5 & 0.125 \\ 0 & 0.125 & 0 \end{bmatrix}$$

La méthode **M3** a été testée sur l'image de synthèse à une résolution on  $256 \times 256$  (voir figure 2.98(a)). La pyramide créée contenait quatre résolutions ( $256 \times 256$ ,  $128 \times 128$ ,  $64 \times 64$ ,  $32 \times 32$ ). Le résultat obtenu est présenté sur la figure 2.98(b). Il est légèrement moins bon que celui de la figure 2.97(b) mais d'une bien meilleure résolution. En ce qui concerne le temps de calcul, le coût est essentiellement dû au recuit simulé, donc identique à celui de la méthode M2 à basse résolution.

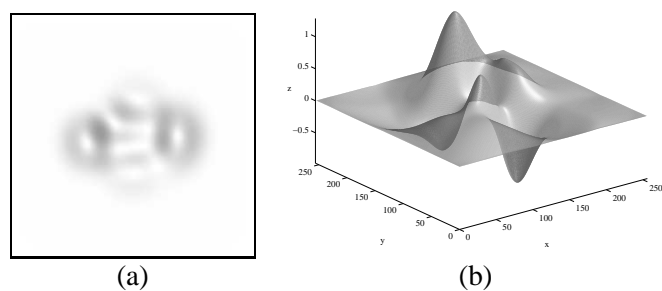


FIG. 2.98 – (a) Image  $256 \times 256$  et (b) résultat par **M3** en utilisant 4 niveaux ( $\lambda_{int} = 500$ ,  $\lambda_{smo} = 20$ ).

## 2.4 Limites des champs de Markov

Nous avons pu appréhender la généralité des champs de Markov à travers les différentes applications liées à l'analyse d'image traitées dans ce chapitre. Nous pouvons affirmer que les champs de Markov sont maintenant un outil classique dont les applications ont déjà quittées le domaine de la recherche pour aller vers le développement industriel. Est-ce à dire que la boucle est bouclée? Il serait un peu prématuré de l'affirmer. En effet, il reste sans doute des améliorations à porter à cette approche tant dans le domaine de la modélisation (recherche d'a priori plus complets et plus en rapport avec la réalité des images) que dans le domaine de l'estimation (estimation des paramètres, efficacité algorithmique). Remarquons notamment que le cadre non-stationnaire est très peu abordé. Les champs de Markov restent donc un domaine actif de recherche.

Néanmoins, nous sommes suffisamment avancés dans ce domaine pour en entrevoir les limites intrinsèques. Nous voyons deux limites principales inhérentes à la définition des champs de Markov :

- Le nombre de variables aléatoires est fixé. Cela se fait de façon naturelle lorsque la variable est le pixel, le champ aléatoire étant défini sur la trame de l'image. En revanche, lorsque l'on considère des entités de plus haut niveaux, comme des objets, la connaissance du nombre de variables est une connaissance a priori forte que nous n'avons pas dans la plupart des cas. Les champs de Markov sur graphes permettent de travailler sur de telles entités [AG92], comme nous l'avons vu pour le recalage de réseaux routiers. Néanmoins, le graphe sur lequel le champ de Markov est défini ne peut pas évoluer durant l'optimisation. L'ensemble des noeuds du graphe doit donc contenir tous les objets de la scène. Une solution consiste à définir un sur-ensemble pour lequel une étiquette binaire est ajoutée à chaque objet pour indiquer la présence effective de l'objet dans la scène [TMM<sup>+</sup>98]. Malgré tout, la recherche du graphe contenant toutes les entités présentes dans l'image doit se faire avant l'optimisation. Il conviendrait donc de définir des modèles pour lesquels le nombre et la localisation des variables aléatoires puissent évoluer au cours de l'optimisation. Ceci peut être intégré dans les processus ponctuels marqués ou les graphes aléatoires.
- La seconde limite des champs de Markov concerne la définition des contraintes à travers des interactions locales. S'il est vrai que ces contraintes peuvent s'avérer globales, comme la minimisation de la longueur des contours d'une carte de segmentation, elles doivent être définies localement. Dès lors, certaines contraintes géométriques sont difficiles à prendre en compte. Par exemple, la forme des objets ou la forme des régions lors d'une segmentation sont difficilement modélisées par des interactions locales. En outre, certaines directions sont privilégiées par le choix d'un modèle comme le montre l'étude des formes stables que nous avons réalisée pour différents modèles binaires. Ce type d'informations géométriques peut être pris en compte par des approches de type "template" ou par les processus ponctuels (voir la "pattern theory" de l'école de Grenander) [aYCK91, Mum96, Mum02].





## Chapitre 3

# Les Processus Ponctuels Objets

### 3.1 Généralités

**Processus ponctuels Marqués**<sup>1</sup> Les processus ponctuels marqués sont d'une utilisation relativement récente en analyse d'image. Une première origine est similaire à l'origine des champs de Markov en imagerie. Il s'agit, en effet, de travaux menés par des statisticiens qui ont appliqué les modèles qu'ils étudiaient à des cas plus ou moins concrets, proches des problématiques abordées en analyse d'image. Les premiers processus ponctuels proposés furent appliqués à des images de synthèse, pour lesquelles on recherchait par exemple une forme connue (disque ou autre) dans un univers bruité [BL93]. Par la suite, des applications sur données réelles, comme le comptage de cellules, ont été proposées [RH99]. Des modèles similaires sont proposés comme une généralisation de l'approche par "templates" au cas de plusieurs objets [SGJM94].

Nous considérons des ensembles non ordonnés de points d'un espace donné  $\chi$ , par exemple<sup>2</sup> $\mathbb{R}$ . Une configuration du processus est alors un ensemble d'énumérable, non ordonné de points de  $\chi$ ,  $\mathbf{x} = \{x_1, \dots, x_n, \dots\}$ .  $\chi$  est muni d'une métrique  $d$  telle que  $(\chi, d)$  soit complet et séparable. Cette métrique permet de définir une topologie et une  $\sigma$ -algèbre borélienne (en pratique,  $\chi$  sera souvent un sous-ensemble compact de  $\mathbb{R}^2$  muni de la distance euclidienne). Une configuration  $\mathbf{x} \subseteq \chi$  est dite **localement finie** si elle place un nombre fini de points dans tout borélien borné  $A \subseteq \chi$ . La famille de toutes les configurations localement finies sera notée  $\mathcal{N}^f$ .

On définit alors la notion de processus ponctuel comme suit :

**Définition 2.** *Un processus ponctuel sur  $\chi$  est une application  $X$  d'un espace probabilisé  $(\Omega, \mathcal{A}, \mathbb{P})$  dans  $\mathcal{N}^f$ , tel que pour tout borélien  $A \subseteq \chi$ , le nombre  $N(A) = N_X(A)$  de points dans  $A$  soit une variable aléatoire (finie).*

Si l'espace  $\chi$  est borné ou si  $\mathcal{N}(\chi)$  est fini presque sûrement, le processus ponctuel est dit **processus ponctuel fini**. Les réalisations d'un processus ponctuel  $X$  sont donc des configurations aléatoires de points telles que pour tout borélien  $A \subseteq \chi$  le nombre de points dans  $A$  soit une variable aléatoire.

Cela signifie qu'un processus ponctuel est une variable aléatoire à valeur dans l'espace mesurable  $(\mathcal{N}^f, \mathcal{N}^f)$ , où  $\mathcal{N}^f$  est la plus petite  $\sigma$ -algèbre telle que pour tout borélien borné  $A \subseteq \chi$  l'application  $\mathbf{x} \rightarrow N_{\mathbf{x}}(A)$  soit mesurable.

La mesure de probabilité induite sur  $\mathcal{N}^f$  est appelée la **loi** de  $X$ .

La loi d'un processus ponctuel  $X$  devrait être la mesure image par l'application  $X$  de  $\mathbb{P}$  sur  $\mathcal{N}^f$ . Mais comme  $\mathcal{N}^f$  est définie par la mesurabilité des applications  $\mathbf{x} \rightarrow N_{\mathbf{x}}(A)$  pour des boréliens  $A \subseteq \chi$ ,

1. Pour une présentation détaillée voir [vL00]

l'analogue de la loi de probabilité pour des variables aléatoires dans un contexte de processus ponctuels est l'ensemble des lois jointes des vecteurs  $(N(A_1), \dots, N(A_m))$  où les  $A_i$  sont des boréliens bornés :

**Définition 3.** La famille des lois en dimensions finie (fidis)<sup>2</sup> d'un processus ponctuel  $X$  sur un espace métrique  $(\chi, d)$  complet et séparable est la collection des lois jointes de  $(N(A_1), \dots, N(A_m))$  pour tout vecteur fini  $(A_1, \dots, A_m)$  de boréliens bornés  $A_i \subseteq \chi$ ,  $i = 1, \dots, m$  de longueur quelconque  $m \in \mathbb{N}$ .

L'intérêt de cette définition est justifié par le théorème suivant :

**Théorème 2.** La loi d'un processus ponctuel  $X$  sur un espace métrique complet et séparable  $(\chi, d)$  est entièrement déterminée par ses fidis.

Donc, deux processus ponctuels partageant les mêmes fidis ont même loi.

La notion de processus ponctuel est généralisée aux processus ponctuels marqués comme suit :

**Définition 4.** Soit  $(\chi, d)$  et  $(\mathcal{K}, d')$  deux espaces métriques, complets et séparables. Un processus ponctuel marqué dont les positions sont dans  $\chi$  et les marques dans  $\mathcal{K}$  est un processus ponctuel sur  $\chi \times \mathcal{K}$  tel que le processus des points non-marqués soit un processus ponctuel bien défini.

Pour comprendre cette définition, il faut d'abord voir que l'espace  $\chi \times \mathcal{K}$  est métrique, complet et séparable dès lors que l'on introduit la métrique  $\rho((x, k), (y, l)) = \max\{d(x, y), d'(k, l)\}$ . Tout processus ponctuel  $Y$  sur l'espace produit  $\chi \times \mathcal{K}$  fait apparaître un processus ponctuel de points non-marqués  $X$  bien défini. Pour le comprendre, il nous faut considérer un borélien  $A \subseteq \chi$ . Alors, le nombre de points non-marqués dans  $A$  peut être écrit :

$$N_X(A) = \sum_{i=1}^{i=M} N_Y(A \times \{i\})$$

Par définition, chaque  $N_Y(A \times \{i\})$  est une variable aléatoire finie, si bien que  $N_X(A)$  est aussi une variable aléatoire finie.

En revanche, dans le cas où  $\mathcal{K}$  est continu, il faut faire plus attention. Par exemple, un processus de poisson sur  $\mathbb{R}^3$  n'est pas un processus ponctuel marqué sur  $\mathbb{R}^3$  avec ses marques dans  $\mathbb{R}$ , puisque  $N_X(A) = N_Y(A \times \mathbb{R})$  n'est pas nécessairement fini pour un borélien  $A \subseteq \mathbb{R}^3$  quelconque.

On s'intéresse maintenant à la multiplicité éventuelle des points d'une configuration. En pratique, il est rare de travailler avec des processus ponctuels qui contiennent des points situés exactement au même endroit. Soit  $\mathcal{N}_s^{lf}$  l'ensemble des configurations localement finies  $\mathbf{x}$  ne contenant que des points distincts, c'est à dire telles que  $N_{\mathbf{x}}(\{x\}) \in \{0, 1\} \quad \forall x \in \mathbf{x}$ .

**Définition 5.** Un processus ponctuel  $X$  est dit simple s'il prend ses valeurs dans  $\mathcal{N}_s^{lf}$  presque sûrement.

La plupart des processus ponctuels utilisés en pratique sont observés dans une région bornée. Cette région peut être dictée par l'application tout comme elle peut résulter d'un choix volontaire pour limiter la taille de l'espace d'intérêt. Dans tous les cas, les réalisations du processus ponctuel contiennent presque sûrement un nombre fini de points.

On verra plus loin qu'il y a une raison théorique importante qui nous fait considérer cette classe de processus ponctuels. La notion de densité est en effet difficile à mettre en place pour des processus qui ne sont pas finis.

Pour construire un processus ponctuel fini, on peut utiliser :

1. Une distribution de probabilité discrète  $(p_n)_{n \in \mathbb{N}}$  pour le nombre de points

---

2. fi dis : *finite-dimensional distributions*, en anglais.

2. Une famille de densités de probabilité symétriques  $(j_n(x_1, \dots, x_n), n \in \mathbb{N})$  sur  $\chi^n$  pour les positions des points.

Le deuxième point implique que l'on suppose que  $\chi$  soit équipée d'une mesure borélienne  $\nu(\cdot)$  de manière à pouvoir définir les densités  $j_n$  par rapport à la mesure produit  $\nu(\cdot)^n$ .

On peut alors construire un processus ponctuel  $X$  de la manière suivante :

On considère  $N(\chi)$  une variable aléatoire de loi  $(j)_n$ , et on conditionne ensuite par rapport aux événements  $N(\chi) = n$  : on prend  $(X_1, \dots, X_n) \in \chi^n$  un vecteur aléatoire de loi  $j_n(\cdot, \dots, \cdot)$  indépendamment de  $N(\chi)$ . La condition de symétrie est nécessaire puisqu'une configuration est indifférente à l'ordre dans lequel ses points sont classés.

Il convient de vérifier toutefois si l'on peut passer également de vecteurs ordonnés à des vecteurs non ordonnés. Ce point est important pour nous puisqu'il nous servira dans des démonstrations ultérieures où l'on basculera des configurations aux  $n$ -uplets, et réciproquement.

Définissons d'abord  $\mathcal{N}$  comme l'ensemble des configurations finies. Ensuite, appelons  $\mathcal{N}^f$  la plus petite  $\sigma$ -algèbre pour laquelle les applications  $\mathbf{x} \rightarrow N_{\mathbf{x}}(A)$  ( $A$  borélien borné) sont mesurables.

On définit ensuite les sous-ensembles de  $\mathcal{N}^f$  suivants :

$$N_n^f = \{\mathbf{x} \in \mathcal{N}^f : N_{\mathbf{x}}(\chi) = n\}$$

et, à chacun de ces ensembles, on associe une  $\sigma$ -algèbre  $\mathcal{N}_n^f$  en prenant la trace de  $\mathcal{N}^f$  sur  $N_n^f$ .

On considère maintenant les fonctions  $f_n : \chi^n \rightarrow N_n^f$  qui, à des vecteurs de taille  $n$ , associent des configurations de  $n$  points. Les  $f_n$  sont mesurables par rapport à la  $\sigma$ -algèbre des boréliens, et grâce à leur invariance par permutation, mesurables par rapport à la  $\sigma$ -algèbre  $\mathcal{B}_s(\chi^n)$  des boréliens symétriques dans  $\chi^n$ .

Par hypothèse, pour tout  $n$ ,  $j_n(\cdot, \dots, \cdot)$  est une densité invariante par permutation et est donc  $\mathcal{B}_s(\chi^n)$ -mesurable.

Considérons la fonction  $i_n : N_n^f \rightarrow \mathbb{R}$  sur les configurations de  $n$  points :

$$i_n(\mathbf{x}) = \frac{1}{n!} \sum_{\varphi} j_n(\varphi(x))$$

où l'on somme sur toutes les applications  $\varphi$  qui, à une configuration, associent l'un des  $n$ -uplets correspondant, (ces  $n$ -uplets diffèrent par des permutations, il y a donc  $n!$   $\varphi$  différentes).

On a donc :  $i_n \circ f_n = j_n$ .  $i_n$  est mesurable.

Les **processus ponctuels de Poisson** sont les processus les plus élémentaires puisque, physiquement, ils traduisent la notion d'indépendance.

**Définition 6.** Soit  $\nu(\cdot)$  une mesure borélienne sur un espace métrique et séparable  $(\chi, d)$  tel que  $\nu(\chi) > 0$  et  $\nu(A) < \infty$  pour tout borélien borné  $A$ , (une telle mesure est dite localement finie). Un processus ponctuel  $X$  sur  $\chi$  est appelé un **processus ponctuel de Poisson** de mesure d'intensité  $\nu(\cdot)$  si :

**P1.**  $N(A)$  suit une loi de Poisson d'espérance  $\nu(A)$  pour tout borélien borné  $A \subseteq \chi$ .

**P2.** Pour  $k$  boréliens disjoints  $A_1, \dots, A_k$ , les variables aléatoires  $N(A_1), \dots, N(A_k)$  sont indépendantes.

Si  $\nu(\cdot)$  est non-atomique, le processus de Poisson est simple. Il est fini si  $\nu(\chi) < \infty$ .

Lorsque  $\chi = \mathbb{R}^d$ , on parle de processus de Poisson **homogène** lorsque la mesure d'intensité est  $\lambda\mu(\cdot)$

où  $\mu$  est la mesure de Lebesgue et  $\lambda$  un paramètre strictement positif. On appelle alors ce paramètre l'intensité du processus.

La propriété **P2** peut être interprétée comme une propriété de non-corrélation spatiale totale, puisqu'elle affirme que ce qui se passe à l'intérieur d'une fenêtre quelconque est totalement indépendant de ce qui se passe à l'extérieur. Le théorème suivant précise cette idée, en affirmant que les points d'un processus ponctuel de Poisson se comportent aléatoirement et n'interagissent pas les uns avec les autres.

**Théorème 3.** *Soit  $X$  un processus ponctuel de Poisson sur un espace métrique complet et séparable  $(\chi, d)$  de mesure d'intensité  $\nu(\cdot)$ , et  $A \subseteq \chi$  un borélien borné. Alors, conditionnellement à  $\{N(A) = n\}$ ,  $X$  restreint à  $A$  suit la loi d'un processus binomial de points indépendants, distribués sur  $A$  suivant  $\nu$ .*

Nous construisons maintenant des modèles de processus ponctuels par leur densité de probabilité (dérivée de Radon-Nikodym) par rapport à un processus de Poisson de référence.

Dans la suite, on considère  $(\chi, d)$  un espace métrique complet et séparable, et  $\pi(\cdot)$  la distribution d'un processus de Poisson sur  $\chi$  de mesure d'intensité finie et **non-atomique**  $\nu(\cdot)$ .

Soit  $p : N^f \rightarrow [0, \infty[$  une fonction positive, mesurable, à valeur dans l'espace des configurations finies de points, telle que :

$$\int_{N^f} p(\mathbf{x}) d\pi(\mathbf{x}) = 1. \quad (3.1)$$

Alors  $p(\cdot)$  est une densité de probabilité et définit un processus ponctuel  $X$  sur  $\chi$ . Comme le processus de Poisson dominant est fini et simple, il en va de même pour  $X$ . Pour interpréter et justifier la formule 3.1, il faut écrire  $N^f$  comme une union parcourant les familles  $N_n^f$  des configurations de  $n$  points :

$$N^f = \bigcup_{n=0}^{\infty} N_n^f$$

La  $\nu$ -mesure de  $N^f$  est  $\nu(\chi)^n/n!$ . Ce  $n!$  est nécessaire puisque  $\chi^n$  est ordonné tandis que  $N^f$  ne l'est pas. Et donc :

$$\nu(N^f) = \sum_{n=0}^{\infty} \frac{\nu(\chi)^n}{n!} = e^{\nu(\chi)}$$

De ces considérations et de la définition du processus de Poisson, on déduit que d'une part, la loi du nombre total de points d'un processus défini par sa densité  $p(\cdot)$  est donnée par la famille  $(p_n)_{n \in \mathbb{N}}$  :

$$p_n = \frac{e^{-\nu(\chi)}}{n!} \int_{\chi} \cdots \int_{\chi} p(\{x_1, \dots, x_n\}) d\nu(x_1) \cdots d\nu(x_n) \quad (3.2)$$

et que d'autre part, conditionnellement aux événements  $\{N(\chi) = n\}$ , les  $n$  points aléatoires ont pour densité jointe :

$$j_n(x_1, \dots, x_n) = \frac{p(\{x_1, \dots, x_n\})}{\int_{\chi} \cdots \int_{\chi} p(\{x_1, \dots, x_n\}) d\nu(x_1) \cdots d\nu(x_n)} \quad (3.3)$$

Il existe deux façons naturelles de conditionner la loi d'un processus ponctuel [SKM87]:

- **Le conditionnement intérieur** : on cherche à exprimer la loi conditionnelle d'un processus ponctuel sachant qu'il y a un de ses points en  $x \in \chi$ . Ce type de conditionnement est formalisé par la **théorie de Palm**.
- **Le conditionnement extérieur** : on cherche à exprimer la loi conditionnelle d'un point  $x \in \chi$  connaissant la configuration sur  $\chi \setminus \{x\}$ . Ce type de conditionnement est formalisé par la **densité conditionnelle de Papangelou**.

Le conditionnement intérieur est très intéressant dès lors que l'on dispose de la propriété d'invariance de la loi du processus ponctuel par translation. En particulier, il permet de définir et d'utiliser des statistiques spatiales. Dans nos applications, nous ne disposons pas de telles propriétés. Nous nous limitons donc au second type de conditionnement.

Commençons par une présentation intuitive de la **densité conditionnelle de Papangelou**  $\lambda(\cdot; \cdot)$  associée à un processus ponctuel simple  $X$ . On peut l'interpréter comme :

$$\lambda(x; \mathbf{x}) dx = \mathbf{P}(N(dx) = 1 | X \cap (dx)^c = \mathbf{x} \cap (dx)^c),$$

ce qui représente la probabilité infinitésimale de trouver un point dans une région  $dx$  autour de  $x \in \chi$  connaissant la configuration  $\mathbf{x}$  du processus ponctuel partout ailleurs qu'en  $dx$ .

Pour définir cette densité conditionnelle plus proprement, il est nécessaire d'introduire la notion de **mesure de Campbell réduite** [SKM87].

**Définition 7.** Soit  $X$  un processus ponctuel simple, sur un espace métrique complet et séparable  $(\chi, d)$ . On définit :

$$C^1(A \times F) = \mathbb{E} \left[ \sum_{x \in X \cap A} \mathbf{1}_{\{X \setminus \{x\} \in F\}} \right]$$

Pour tout borélien borné  $A \subseteq \chi$  et  $F \in \mathcal{N}^{lf}$ .

De la même manière que dans le paragraphe précédent, on peut prolonger cette fonction en une mesure  $\sigma$ -finie unique sur la  $\sigma$ -algèbre produit  $\mathcal{B}(\chi) \times \mathcal{N}^{lf}$ . L'équivalent de la formule de Campbell-Mecke donne :

$$\mathbb{E} \left[ \sum_{x \in X} g(x, X \setminus \{x\}) \right] = \int_{\chi} \int_{\mathcal{N}^{lf}} g(x, \mathbf{x}) dC^1(x, \mathbf{x}). \quad (3.4)$$

pour toute fonction mesurable  $g(\cdot, \cdot)$  positive ou intégrable.

Supposons que, pour un borélien borné fixé  $A \in \mathcal{B}(\chi)$ ,  $(A \times \cdot)$  soit absolument continue par rapport à la loi  $\mathbf{P}(\cdot)$  de  $X$ . Alors,

$$C^1(A \times F) = \int_F \Lambda(A; \mathbf{x}) d\mathbf{P}(\mathbf{x})$$

pour une fonction mesurable  $\Lambda(A; \cdot)$  définie de manière unique à un ensemble  $\mathbf{P}$  négligeable près. De plus, on peut en trouver une version telle que pour  $\mathbf{x}$  fixé,  $\Lambda(\cdot; \mathbf{x})$  soit une mesure borélienne localement finie, que l'on appelle **noyau de Papangelou** du premier ordre.

Si  $\Lambda(\cdot; \mathbf{x})$  admet une densité  $\lambda(\cdot; \mathbf{x})$  par rapport à une mesure de référence  $\nu(\cdot)$  sur  $\chi$ , l'expression de l'équation (3.4) se factorise sous la forme :

$$\mathbb{E} \left[ \int_{\chi} g(x, X) \lambda(x; X) d\nu(x) \right] \quad (3.5)$$

On appelle alors la fonction  $\lambda(\cdot; \cdot)$  **l'intensité conditionnelle de Papangelou**.

Le cas où  $X$  est dominé par un processus ponctuel de Poisson est particulièrement important :

**Théorème 4.** Soit  $X$  un processus ponctuel fini défini par une densité  $p(\mathbf{x})$  par rapport à un processus ponctuel de Poisson de mesure d'intensité finie et non-atomique  $\mu(\cdot)$ . Alors  $X$  a une intensité conditionnelle de Papangelou :

$$\lambda(u; (x)) = \frac{p(\mathbf{x} \cup \{u\})}{p(\mathbf{x})} \quad (3.6)$$

pour  $u \notin \mathbf{x}$  et  $\mathbf{x} \in \mathcal{N}^f$ .

Comme pour les champs aléatoires, nous introduisons les notions de voisinage et la propriété de Markov. La définition intuitive des processus de Markov est assez simple : il s'agit de processus ponctuels finis définis par une densité et dont l'intensité conditionnelle ne dépend que d'un certain voisinage.

Les processus ponctuels de Markov sont très utilisés pour différentes applications. En traitement d'image leur avantage est de permettre une implantation informatique facile. Historiquement, ils ont également été très utilisés en physique statistique, entre autre sous le nom de processus ponctuels de Gibbs.

La particularité de ces processus de Gibbs est de présenter des densités sous une forme énergétique en utilisant des potentiels d'interaction entre les points d'une réalisation du processus ponctuel.

Nous présentons ici rapidement quelques définitions et propriétés des processus de Markov.

On considère une relation symétrique et réflexive  $\sim$  sur  $\chi$ . On dit que deux points  $u$  et  $v$  de  $\chi$  sont voisins si  $u \sim v$ . Par exemple, on peut considérer la relation de proximité sur  $\chi = \mathbb{R}$

$$u \sim v \Leftrightarrow d(u, v) \leq R$$

pour un réel  $R$  quelconque.

**Définition 8.** Le voisinage  $\partial(A)$  d'un ensemble  $A \subseteq \chi$  est défini comme l'ensemble :

$$\partial(A) = \{x \in \chi : \exists a \in A, x \sim a\}$$

Ripley et Kelly donnent la définition suivante d'un processus ponctuel de Markov [RK77] :

**Définition 9.** Soit  $(\chi, d)$  un espace métrique complet et séparable,  $\nu(\cdot)$  une mesure borélienne finie non-atomique, et  $\pi_\nu(\cdot)$  la loi d'un processus ponctuel de Poisson de mesure d'intensité  $\nu(\cdot)$ . Soit  $X$  un processus ponctuel sur  $\chi$  défini par sa densité  $p(\cdot)$  par rapport à  $\pi_\nu(\cdot)$ . Alors  $X$  est un **processus ponctuel de Markov** sous la relation symétrique et réflexive  $\sim$  sur  $\chi$  si, pour tout  $\mathbf{x} \in N^f$  tel que  $p(\mathbf{x}) > 0$ ,

- (a)  $p(\mathbf{y}) > 0$  pour tout  $\mathbf{y} \subseteq \mathbf{x}$ ;
- (b) pour tout  $u \in \chi$ ,  $p(\mathbf{x} \cup \{u\})/p(\mathbf{x})$  ne dépend que de  $u$  et  $\partial(\{u\}) \cap \mathbf{x} = \{x \in \mathbf{x} : u \sim x\}$ .

Le théorème suivant permet d'exprimer la densité d'un processus de Markov sous une forme plus pratique. Pour cela, il faut d'abord poser une définition :

**Définition 10.** Soit  $\sim$  une relation de voisinage symétrique et réflexive sur  $\chi$ . Une configuration  $\mathbf{x} \in N^f$  est appelée **clique** si tous les éléments de  $\mathbf{x}$  sont voisins les uns des autres, c'est à dire si :

$$\forall u, v \in \mathbf{x} \quad u \sim v$$

Par convention, la configuration vide est une clique également.

On peut ensuite démontrer le théorème suivant, qui est l'équivalent du théorème d'Hammersley-Clifford pour les processus ponctuels :

**Théorème 5.** Une densité de processus ponctuel  $p : N^f \rightarrow [0, \infty[$  est markovienne sous une relation de voisinage  $\sim$  si et seulement si il existe une fonction mesurable  $\phi : N^f \rightarrow [0, \infty[$  telle que :

$$p(\mathbf{x}) = \prod_{\text{cliques } \mathbf{y} \subseteq \mathbf{x}} \phi(\mathbf{y}) \quad (3.7)$$

pour tout  $\mathbf{x} \in N^f$ .

Une autre formulation peut être donnée avec un produit sur tous les sous-ensembles de  $\mathbf{x}$  en imposant à  $\phi$  de valoir 1 sur les sous-ensembles n'étant pas des cliques.

Donnons un exemple classique, le processus de Strauss. On se place sur  $\chi = \mathbb{R}^2$ , et on considère un sous-ensemble borné  $K \subseteq \chi$  muni de la mesure de Lebesgue  $\mu(\cdot)$ .

On définit des densités par rapport à un processus ponctuel de Poisson sur  $K$ , d'intensité  $\mu(\cdot)$ . Un tel processus est simple et fini, comme il se doit pour pouvoir définir des densités de processus.

On considère d'abord des processus ayant une densité  $h(\cdot)$  de la forme:

$$h(\mathbf{x}) = \alpha \cdot \beta^{n(\mathbf{x})}$$

où  $\beta > 0$  et  $n(\mathbf{x})$  représente le nombre de points d'une configuration  $\mathbf{x}$ . Une telle densité est intégrable et  $\alpha$  est la constante de normalisation. La constante  $\beta$  est un paramètre qui permet de jouer sur l'intensité du processus. Celle-ci vaut alors  $\beta\mu(\cdot)$ .

Il est intéressant de remarquer que  $\beta$  est un paramètre d'échelle : de manière équivalente, on peut changer le borné  $K$  d'intérêt en posant  $K'$  tel que  $\mu(K') = \beta\mu(K)$ .

On considère une densité de la forme:

$$p(\mathbf{x}) = \alpha \cdot \beta^{n(\mathbf{x})} \prod_{i < j} g(x_i, x_j) \quad \mathbf{x} = \{x_1, \dots, x_{n(\mathbf{x})}\} \quad (3.8)$$

Où la fonction  $g$  vérifie :

$$g(x_i, x_j) = \begin{cases} \gamma, & \text{si } d(x_i, x_j) < r \\ 1, & \text{si } d(x_i, x_j) \geq r \end{cases}$$

Avec  $0 \leq \gamma$  et  $r > 0$ . Un tel processus est appelé **processus de Strauss**. Il s'agit d'un processus de Markov sous la relation  $\sim$  de proximité :  $u \sim v \Leftrightarrow d(u, v) < r$ . On peut avoir une expression plus simple de cette densité sous la forme donnée par le théorème de Hammersley-Clifford étendu aux processus ponctuels :

$$p(\mathbf{x}) = \alpha \beta^{n(\mathbf{x})} \gamma^{s(\mathbf{x})} \quad (3.9)$$

où  $s(\mathbf{x})$  représente le nombre de paires de points en relation dans la configuration  $\mathbf{x}$ . Reste à étudier l'influence de  $\gamma$ :

- Pour  $\gamma = 1$  on retrouve un processus ponctuel de Poisson sur notre borélien borné  $K$  d'intensité  $\beta\lambda(\cdot)$ .
- Pour  $\gamma \in ]0, 1[$ , le processus montre une répulsion entre les points proches au sens de la relation  $\sim$ .
- Avec  $\gamma = 0$ , on obtient ce que l'on appelle un **"hard core process"**. La densité interdit d'avoir deux points de la configuration qui soient voisins.

Le cas  $\gamma > 1$  est le cas qui intéresse Strauss puisqu'il devait permettre de manifester une attraction entre les points d'une réalisation. Malheureusement, dans ce cas là, la densité n'est pas intégrable, à moins de multiplier cette densité par un terme du type  $\mathbf{1}\{n(\mathbf{x}) \leq n_0\}$  pour un entier  $n_0$  fixé.

**Simulation de processus ponctuels par méthodes MCMC**- Les méthodes de simulation par chaînes de Markov sont multiples et variées. L'objectif sous-jacent à ces méthodes est la simulation de variables aléatoires suivant une certaine loi de probabilité. Toutes ces méthodes construisent donc des chaînes de Markov dont la loi stationnaire est la loi objectif. Parmi les méthodes les plus connues, on retient *l'échantillonneur de Gibbs* et les *dynamiques de Metropolis Hastings* et de *Metropolis Hastings Green*. Les deux dernières méthodes ont un avantage : elles permettent de simuler des lois connues au facteur de normalisation près.

La méthode de *Metropolis Hastings Green* présente l'énorme avantage dans le cadre de la simulation de processus ponctuels de permettre la simulation de variables aléatoires sur des espaces dont la dimension varie, ce qui est justement notre cadre d'intérêt puisque les configurations sont de taille variable.



L'algorithme a été introduit par Green en 1995 [Gre95], comme une généralisation de l'algorithme de Metropolis Hastings. Nous le présentons ici directement, sans présenter Hasting-Metropolis, puisqu'une des innovations de Green a été de remplacer les densités d'intérêt par des mesures, ce qui nous convient mieux dans le cadre des processus ponctuels.

On se donne:

1. un espace d'état  $\mathcal{C}$  et une mesure  $\pi(\cdot)$  non normalisée sur cet espace.

Cette loi de probabilité, dont on ignore la constante de normalisation, est la **mesure objectif**. On entend donc construire une chaîne de Markov dont la mesure invariante est cette loi.

2. un noyau de transition  $Q(\cdot; \cdot)$  défini sur  $\mathcal{C} \times \mathcal{B}(\mathcal{C})$ .

Ce noyau doit être bien connu : on veut dire que, pour tout  $x \in \mathcal{C}$ , on doit pouvoir simuler suivant  $Q(x; \cdot)$  et que  $Q(x, y)$  doit être calculable pour tout  $x, y$ . On va voir que ce noyau n'est pas du tout le noyau de la chaîne de Markov construite par l'algorithme. On l'appelle donc **noyau de proposition**.

3. pour relier  $\pi(\cdot)$  et  $Q$ , on a besoin d'une mesure symétrique  $\xi(\cdot; \cdot)$  sur  $\mathcal{C} \times \mathcal{C}$  telle que  $\xi$  domine  $\pi(dx)Q(x, dy)$  et qu'il existe une dérivée au sens de Radon-Nikodym:

$$f(x, y) = \frac{\pi(dx)Q(x, dy)}{\xi(dx, dy)} \quad (3.10)$$

On appelle alors **rapport de Green** le rapport:

$$R = \frac{f(y, x)}{f(x, y)} \quad (3.11)$$

L'algorithme se présente de la manière suivante :

1. A partir de l'état courant  $x$ , on simule  $y \sim Q(x, \cdot)$ .
2. On évalue le rapport de Green  $R(x, y)$ ,
3. On accepte  $y$  suivant une loi de Bernoulli de paramètre  $\min(1, R(x, y))$ . Si  $y$  est refusée, on garde  $x$  comme nouvel état. Sinon, le nouvel état est donné par  $y$ .

Dans son article, Green se place dans un cadre très général et laisse beaucoup de liberté sur le noyau de transition  $Q$ . Il est important de comprendre l'importance de ce noyau. Hormis la condition imposée par l'existence de la mesure  $\xi$ , ce noyau n'influe pas du tout sur la mesure objectif  $\pi(\cdot)$ . En revanche, intuitivement, il est évident qu'en jouant sur  $Q$ , on va pouvoir jouer sur la vitesse de convergence de l'algorithme.

Pour bien voir cela, plaçons-nous dans un cas particulier :

Nous faisons deux hypothèses. Supposons tout d'abord que toutes les mesures soient dominées par une mesure  $\lambda(\cdot)$ . Par exemple, il suffit de prendre  $\mathcal{C} = \mathbb{R}$  et la mesure de Lebesgue comme mesure dominante. Cela revient à se placer dans le cadre plus restreint de l'algorithme de Metropolis Hastings. Supposons ensuite que :

$$Q(x, y) = Q(y, x)$$

ce qui n'est qu'une condition de réversibilité sur le noyau de proposition  $Q$ . Sous ces hypothèses, le rapport de Green devient:

$$R(x,y) = \frac{h(y)}{h(x)}$$

où  $h(\cdot)$  est la densité de la mesure objectif  $\pi(\cdot)$  sous la mesure dominante  $\lambda$ . Dans ce cas, le rapport de Green devient un rapport de densité. En quelque sorte, il représente la pertinence relative de la proposition  $y$  par rapport à l'état courant  $x$ . Donc, plus  $y$  est pertinent au sens de la mesure objectif, et plus ce rapport à de chance d'être grand. Or, plus il est grand, plus la probabilité de refuser  $y$ , donc de perdre une itération, est faible.

Il faut s'arrêter quelques instants pour réfléchir à cette notion de perte d'itération. Il faut distinguer deux versions de cette notion :

- la première concerne la partie transitoire de la chaîne de Markov, c'est à dire lorsque  $K^n$  est encore loin de  $\pi(\cdot)$ . Proposer des états pertinents au sens de  $\pi$  permet de se placer dans un de ses modes plus rapidement. Le problème devient alors tout autre: il faut être sûr de ne pas rester dans ce mode, c'est à dire que le noyau doit permettre de passer d'un mode à un autre facilement.
- la deuxième concerne la partie stationnaire de la chaîne : une fois que la convergence est atteinte (au sens d'une certaine approximation) il est plus intéressant de ne pas rester dans l'état courant. En effet, en diminuant l'autocorrélation de la chaîne, on diminue la variance de l'erreur empirique.

On comprend donc l'intérêt de fabriquer des noyaux de proposition qui correspondent bien à la mesure objectif.

La condition de réversibilité sur le noyau de proposition impose de faire une mauvaise proposition avec la même probabilité que de faire une bonne proposition. C'est pourquoi, dans un cadre plus général, c'est à dire sans les deux hypothèses trop simplificatrices ci-dessus, Brooks, Guidici et Roberts proposent des méthodes qui permettent d'assurer d'avoir un bon coefficient de Green [BGR03]. Il semble toutefois qu'il manque encore un cadre mathématique formel pour justifier proprement ces heuristiques. Reste à voir les manières possibles de fabriquer de bons noyaux de propositions.

### Conditions imposées au noyau :

L'objectif est de fabriquer une chaîne de Markov aperiodique, irréductible, de loi stationnaire  $\pi(\cdot)$ . On appelle  $K$  le noyau de transition de la chaîne de Markov.

L'aperiodicité est assurée dès lors que le noyau  $K$  permet de rester dans l'état courant avec une probabilité strictement positive.

L'irréductibilité est plus difficile à assurer. On verra plus loin que, dans le cas des processus ponctuels, on l'obtient facilement. En fait, ce qui nous intéresse vraiment c'est la récurrence au sens de Harris, puisqu'elle permet de se débarrasser des cas pathologiques où l'on part d'un ensemble de mesure nulle.

Reste à assurer que la chaîne possède comme mesure invariante la mesure  $\pi(\cdot)$ . Pour cela, on impose une condition forte, c'est la condition dite d'"équilibre minutieux"<sup>3</sup>:

$$\int_A \int_B \pi(dx) K(x, dx') = \int_B \int_A \pi(dx') K(x', dx) \quad (3.12)$$

---

3. "detailed balance" en anglais.

La condition (3.12) est une condition forte. L'algorithme de Green la vérifie naturellement: cette condition se traduit par la condition d'existence de la mesure symétrique  $\xi$ .

### Mélange de noyaux :

Green propose d'utiliser comme noyau de proposition  $Q$  un mélange de noyaux  $q_n(x, \cdot)$ . C'est à dire que, lorsque l'on est dans l'état  $x$ , on choisit d'abord un "mouvement" de type  $m$ , et on passe à l'état  $dx$  avec une probabilité  $q_n(x, dx')$ . On autorise  $\sum_m q_m(x, \mathcal{C}) \leq 1$  et pour chaque  $x$ , il est possible d'avoir  $q_m(x, \mathcal{C}) = 0$ . On peut donc "proposer" de rester dans l'état courant.

Pour chaque transformation possible, on définit alors un coefficient de Green  $R_m(x, x')$ . Green démontre que pour avoir la condition (3.12), il suffit que pour chacun des noyaux  $q_m$ ,  $\pi(dx)q_m(x, dx')$  admette une densité finie  $f_m(x, x')$  par rapport à une mesure symétrique  $\xi_n$  sur  $\mathcal{C} \times \mathcal{C}$  et de poser comme rapport de Green pour chacun des noyaux:

$$R_m(x, x') = \frac{f_m(x', x)}{f_m(x, x')} \quad (3.13)$$

La loi de Bernoulli qui permet d'accepter ou non la proposition prend alors pour paramètre:

$$\alpha_m(x, x') = \min \left\{ 1, \frac{f_m(x', x)}{f_m(x, x')} \right\} \quad (3.14)$$

L'algorithme se décompose alors de la manière suivante :

1. Choisir une transformation  $m$ , avec la probabilité  $p_m = q_m(x, \mathcal{C})$ . Avec la probabilité  $1 - \sum_m q_m(x, \mathcal{C})$  rester dans l'état courant  $x$ .
2. Générer  $y \sim q_m(x, \cdot)$ .
3. Calculer le rapport de Green  $R_m(x, y)$ .
4. Accepter la proposition avec une probabilité  $\alpha_m(x, y)$ .

Nous nous intéressons maintenant plus particulièrement à l'application de cet algorithme pour la simulation de processus ponctuels, c'est à dire pour  $\mathcal{C} = N^f$ .

**MHG pour les processus ponctuels-** Geyer et Moller présentent un algorithme de Metropolis Hastings Green qui permet de simuler facilement des processus ponctuels [GM94]. L'objectif est en fait de simuler des processus ponctuels simples et finis ayant une densité par rapport à un processus de Poisson de référence.

Nous présentons ici le noyau minimal nécessaire à la convergence de l'algorithme vers la loi objectif: en effet, cet algorithme, avec des conditions imposées sur la densité du processus que l'on veut simuler, permet d'assurer à la fois la récurrence au sens de Harris de la chaîne ainsi construite, et l'ergodicité géométrique.

On se place donc sur  $\mathcal{C} = N^f(\chi)$ .

On considère un processus de Poisson de référence sur  $\chi$ , simple et fini de loi de probabilité  $\mu(\cdot)$  et de mesure d'intensité  $\lambda$ . En fait, on se place sur  $S \subseteq \chi$  de mesure  $\lambda(S)$  finie, (par exemple on se place sur un fermé borné de  $\mathbb{R}^d$  avec comme mesure d'intensité la mesure de Lebesgue).

On note donc  $\mathcal{C} = N_S^f = \{x \in N^f \text{ tq } \forall x \in x, x \in S\}$ .

On veut simuler un processus ponctuel de densité  $h(\cdot)$  non normalisée par rapport à  $\mu(\cdot)$ .

On rappelle que l'on note  $x = \{x_1, \dots, x_{n(x)}\}$ . On considère un noyau  $Q$  qui propose d'ajouter à

$\mathbf{x}$  un point choisi de manière uniforme dans  $S$  avec une probabilité  $p_b$  et d'enlever un point avec une probabilité  $p_d$  ( $p_b + p_d = 1$ ). On écrit donc le noyau  $Q$  sous la forme :

$$Q = p_b Q_b + p_d Q_d \quad (3.15)$$

avec les deux noyaux de naissance et de mort :

$$Q_b(\mathbf{x}, A) = \int_{u \in S} \mathbf{1}_A(\mathbf{x} \cup u) \frac{\lambda(du)}{\lambda(S)} \quad (3.16)$$

$$Q_d(\mathbf{x}, A) = \sum_{u \in \mathbf{x}} \mathbf{1}_A(\mathbf{x} \setminus u) \frac{1}{n(\mathbf{x})} \quad (3.17)$$

Pour prouver la stationnarité de la loi objectif, nous considérons la mesure  $\xi$  suivante ( $A$  et  $B$  sont des boréliens de  $\chi$ ) :

$$\xi(A \times B) = \int_{\mathcal{C}} \int_{u \in S} \mathbf{1}_A(\mathbf{x}) \mathbf{1}_B(\mathbf{x} \cup u) \lambda(du) \mu(d\mathbf{x}) + \int_{\mathcal{C}} \mathbf{1}_A(\mathbf{x}) \sum_{u \in \mathbf{x}} \mathbf{1}_B(\mathbf{x} \setminus u) \mu(d\mathbf{x}) \quad (3.18)$$

Il faut démontrer que cette mesure est symétrique. On prend pour cela :  $A = A \cap N_n^f$  et on a donc :

$$\xi(A_n \times B_{n-1}) = \frac{e^{-\lambda(S)}}{n!} \int_{\chi^n} \sum_{u \in S} \mathbf{1}_{A_n}(\mathbf{x}) \mathbf{1}_{B_{n-1}}(\mathbf{x} \setminus u) \lambda^n(d\mathbf{x}) \quad (3.19)$$

$$= \frac{e^{-\lambda(S)}}{n!} \int_{\chi^n} n \mathbf{1}_{A_n}(\{x_1, \dots, x_n\}) \mathbf{1}_{B_{n-1}}(\{x_1, \dots, x_n\}) \lambda^n(d\mathbf{x}) \quad (3.20)$$

$$= \frac{e^{-\lambda(S)}}{(n-1)!} \int_{\chi^{n-1}} \int_S \mathbf{1}_{B_{n-1}}(\mathbf{y}) \mathbf{1}_{A_n}(\mathbf{y} \cup u) \lambda^{n-1}(d\mathbf{y}) \lambda(du) \quad (3.21)$$

$$= \xi(B_{n-1} \times A_n) \quad (3.22)$$

Cela suffit pour prouver que  $\xi$  est symétrique.

Il reste à montrer que  $\xi(d\mathbf{x}, d\mathbf{x}')$  domine  $\pi(d\mathbf{x})Q(\mathbf{x}, d\mathbf{x}')$  et à calculer la dérivée de Radon-Nykodim. Pour ce faire, il est intéressant de remarquer dans la succession d'égalité (3.19) que la mesure  $\xi(A, B)$  est nulle dès lors que  $A$  ne contient que des configurations qui ne peuvent pas être obtenues en ajoutant ou en retirant un point d'une des configurations de  $B$ . Par conséquent, dès lors que l'on prend un ensemble  $A \times B$  de  $\pi(\cdot)Q(\cdot, \cdot)$  mesure non-nulle, il est de  $\xi$  mesure non nulle.

Il y a donc deux cas à considérer :

1. Si  $\mathbf{y} = \mathbf{x} \cup u$ , alors la densité objectif  $\pi$  et le noyau  $Q$  permettent d'écrire :

$$\pi(d\mathbf{x})Q(\mathbf{x}, d\mathbf{y}) = h(\mathbf{x})\mu(d\mathbf{x})p_b \frac{\lambda(du)}{\lambda(S)}$$

la définition de  $\xi$  (cf. (3.18)) donne, quant à elle :

$$\xi(d\mathbf{x}, d\mathbf{y}) = \mu(d\mathbf{x})\lambda(du)$$

Ce qui permet de vérifier l'absolue continuité dans ce cas et d'obtenir la dérivée de Radon-Nykodym :

$$\boxed{f(\mathbf{x}, \mathbf{y}) = p_b \frac{h(\mathbf{x})}{\lambda(S)}} \quad (3.23)$$

2. L'autre cas consiste à considérer  $y = x \setminus u$ . On a alors, d'après la densité objectif et le noyau :

$$\pi(dx)Q(x, dy) = h(x)\mu(dx)p_d \frac{1}{n(x)}$$

La mesure  $\xi$  donne:

$$\xi(dx, dy) = \mu(dx)$$

Et l'on obtient l'absolue continuité et la dérivée :

$$\boxed{f(x, y) = p_d \frac{h(x)}{n(y)}} \quad (3.24)$$

Les équations (3.23) et (3.24) définissent donc la dérivée de  $\pi(\cdot)Q(\cdot, \cdot)$  par rapport à  $\xi$ . Elles permettent donc de calculer le coefficient de Green du noyau de transformation.

Ce coefficient prend deux expressions différentes suivant le cas dans lequel on se place :  $y$  peut être obtenu en ajoutant ou en retranchant un point à  $x$ .

1. Dans le cas d'une naissance,  $y = x \cup u$ , le coefficient de Green a pour expression :

$$\boxed{R(x, y) = \frac{f(y, x)}{f(x, y)} = \frac{p_d h(y) \lambda(S)}{p_b h(x) n(y)}} \quad (3.25)$$

Où l'on peut remplacer  $n(y)$  par  $n(x) + 1$ .

2. Dans le cas d'une mort,  $y = x \setminus u$ , le coefficient de Green a pour expression :

$$\boxed{R(x, y) = \frac{f(y, x)}{f(x, y)} = \frac{p_b h(y) n(x)}{p_d h(x) \lambda(S)}} \quad (3.26)$$

Les coefficients (3.25) et (3.26) permettent d'assurer la condition (3.12). On sait donc qu'un tel algorithme admet pour mesure stationnaire la version normalisée de  $\pi(\cdot)$ .

Il reste que des conditions sont encore à vérifier: l'irréductibilité et la récurrence au sens de Harris.

L'apériodicité est assurée dès lors que l'on prend  $p_d < 1$  au lieu de  $p_d + p_b = 1$ . On veut dire par là qu'avec une probabilité  $1 - (p_d + p_b) > 0$ , on reste dans l'état courant.

Des conditions de stabilité ont été apportées par Ruelle. Elles sont nées de considérations physiques : Ruelle utilisait les processus ponctuels en physique statistique. Ces conditions portent sur la densité du processus ponctuel d'intérêt, mais permettent d'assurer certaines propriétés de convergence de la chaîne de Markov produite par l'algorithme présenté.

Il se trouve que la première condition s'appliquait à des processus ponctuels définis sur un espace de volume infini. Elle est moins intéressante pour nous que la seconde:

**Définition 11.** *Un processus ponctuel de densité non-normalisée  $h$  par rapport à un processus ponctuel de référence de loi  $\mu(\cdot)$  est **stable au sens de Ruelle** si il existe un réel  $M \geq 1$  tel que :*

$$h(x) \leq M^{n(x)} \quad \forall x \in N^f(\chi) \quad (3.27)$$

Du point de vue physique, cette condition limite la force des interactions entre les points d'une configuration.

Enfin, cette condition locale permet d'assurer l'intégrabilité de  $h$  dans le cas  $\mathcal{C} = \mathbb{N}$ , puisque :

$$\int h(x)\mu(dx) \leq \int M^{n(x)}\mu(dx) = \sum_{k=0}^{\infty} \frac{M^k}{k!} \lambda(S)^k = e^{M\lambda S}$$

Par exemple, le processus de Strauss avec  $\gamma > 1$  ne vérifie pas cette condition.

Voilà la seconde condition de stabilité que l'on définit :

**Définition 12.** *Un processus ponctuel fini de densité non-normalisée  $h(\cdot)$  par rapport à un processus de Poisson de loi  $\mu$  est **stable** s'il existe un réel  $M$  tel que :*

$$h(\mathbf{x} \cup u) \leq M h(\mathbf{x}), \text{ pour tout } \mathbf{x} \in \mathcal{C} \text{ et } u \in S. \quad (3.28)$$

Cette condition implique la condition (3.27).

Geyer et Moller [GM94] ont montré, dans le cadre des processus ponctuels, que le mouvement de naissance-mort implique la récurrence au sens de Harris et la convergence géométrique ergodique de la chaîne. Green [Gre95] a étendu ces travaux dans le cadre d'un algorithme de type Metropolis Hastings. Ce type d'algorithme permet de définir d'autres mouvements (perturbations de la configuration courante) accélérant la convergence en pratique. Si ces mouvements restent dans un espace de dimension donnée (pas d'ajout ou de suppression d'objets), la convergence reste assurée. Cela nous permet de définir des transformations sur un objet de la configuration (translation, rotation, modification des marques,...). Pour des mouvements plus complexes (naissance dans le voisinage d'un objet de la configuration, fusion-division,...) il faut redémontrer l'irréductibilité.

Le recuit simulé peut également être généralisé au cadre des processus ponctuels. On rappelle que l'idée générale du recuit simulé consiste à utiliser un échantillonneur de Metropolis Hastings en utilisant une densité variable avec le temps :

$$h(\mathbf{x})^{\frac{1}{T_t}} \quad (3.29)$$

où  $(T_t)$  tend vers zéro. Cela revient à faire évoluer les modes de la densité  $h$  en diracs.

Sous une condition forte sur la vitesse de décroissance de la température, on est assuré d'obtenir la réalisation qui maximise la densité  $h$ .

Dans le cadre des processus ponctuels, la mesure de référence contre laquelle on effectue le recuit-simulé est la mesure du processus de Poisson.

Il est donc plus pertinent de sortir le paramètre  $\beta$  de  $h$ , puisque ce paramètre peut être vu comme un paramètre d'échelle de la mesure de référence. On effectue donc le recuit simulé sur des densités de la forme :

$$\beta (h(\mathbf{x}))^{\frac{1}{T_t}} \quad (3.30)$$

### 3.2 Extraction du ponctuel

<sup>4</sup> La détection de lésions focales cérébrales à partir d'images par résonance magnétique (IRM) est une tâche de segmentation difficile. Cette tâche est grandement facilitée par l'utilisation d'informations a priori sur les caractéristiques des lésions (localisation, taille, forme,...). Les méthodes proposées sont donc dédiées à un type particulier de lésions. Pour des raisons historiques et du fait de leur importance du point de vue médical, la plupart des approches se concentrent sur les lésions multiples de la matière blanche dues à la sclérose en plaques[UWS<sup>+</sup>97], alors que les approches concernant les larges lésions focales sont apparues plus récemment[CK01]. Plusieurs types de petites lésions focales sont perceptibles dans les IRM de sujets âgés. Leur caractère pathologique et leur influence sur les capacités cognitives sont toujours sujets à débat. Le projet dans lequel s'inscrit ce travail consiste à détecter un vieillissement pathologique à partir d'IRM. Alors qu'un observateur humain est plus efficace que les méthodes automatiques pour détecter et discriminer les lésions, le comptage et la localisation de ces lésions est une tâche pénible de par leur multitude. Typiquement, ces lésions sont évaluées visuellement et les images sont évaluées sur une échelle semi-quantitative, c'est à dire classées en quelques groupes suivant le nombre et l'importance des lésions. Nous nous intéressons à un type particulier de lésions appelé "espace élargi de Virchow-Robin" (EVR). Cet espace correspond à un léger vide autour des artères profondes nourrissant la matière blanche. Ces lésions apparaissent comme de petites structures tubulaires pleines de liquide céphalo-rachidien (LCR) perpendiculaires à la surface corticale. Elles sont typiquement d'une taille proche de la résolution spatiale des IRM actuelles (1-3 mm de diamètre et 3-15 mm de longueur). Un seul cerveau peut en contenir plusieurs centaines. Nous définissons un processus ponctuel marqué (PPM) pour détecter ces petites lésions, distribuées non uniformément et pour lesquelles nous avons certaines contraintes de forme et de positions relatives. Un EVR est typiquement constitué d'une structure tubulaire de quelques voxels (voir figure 3.1). Leur répartition n'est pas uniforme et un phénomène de regroupement de ces lésions est visible (voir figure 3.2). Ce caractère attractif est modélisé par les interactions du PPM ainsi que la non-superposition des lésions.

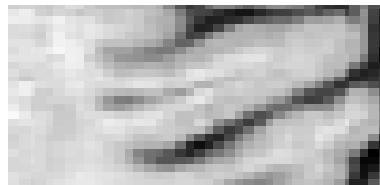


FIG. 3.1 – Un exemple de la forme tubulaire des EVR (au centre de l'image)

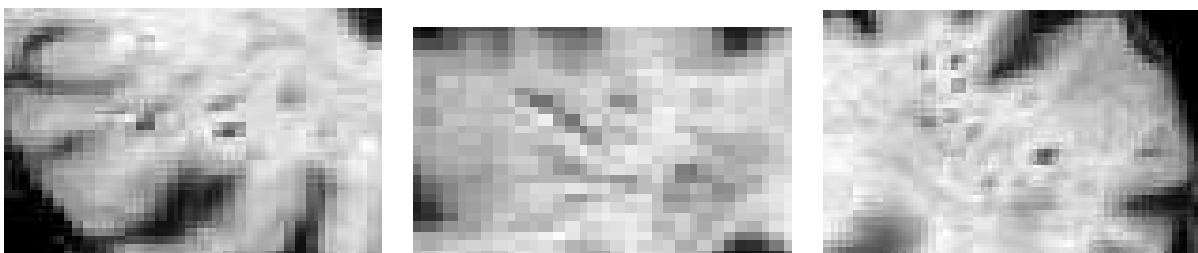


FIG. 3.2 – Propriété de regroupement des EVR : coupes axiale (à gauche), coronale (au milieu) et sagittale (à droite)

4. Ce travail a été effectué lors d'un séjour en juillet-août 2001 à l'Institut Max Planck de Leipzig. Bibliographie [18]

**Modélisation des données-** Nous définissons, en premier lieu, des filtres interprétés comme des indicateurs d'ERV. Ces filtres extraient l'information radiométrique des données. Cette information n'étant pas suffisante, nous définissons ensuite un modèle a priori pour inclure les propriétés géométriques et structurelles de ces lésions. Ces filtres nous permettent de définir le terme d'attache aux données et de restreindre l'espace de recherche des lésions.

Nous considérons des filtres bidimensionnels suivant les trois directions principales  $(u, v, w)$  prenant en compte la variabilité du diamètre et les phénomènes de volume partiel. Un filtre est perpendiculaire à la direction considérée, par exemple  $u$ . Il est constitué du voxel central  $s$ , des huit plus proches voisins  $t_1 \in \mathcal{N}_u^1(s)$  et des seize voisins suivants  $t_2 \in \mathcal{N}_u^2(s)$ . Un EVR vérifie trois propriétés radiométriques : (i) il contient du LCR qui apparaît avec une radiométrie faible (néanmoins le volume partiel peut accroître le niveau du fait des tissus voisins), (ii) les voxels voisins ont une intensité correspondant aux matières blanche ou grise, et (iii) les voisins sont contrastés avec l'EVR. Nous définissons trois fonctions correspondant à ces trois propriétés  $F^{black}(i_s)$ ,  $F^{white}(\min(i_t, t \in \mathcal{N}_u^2(s)))$  et  $F^{contrast}(\frac{\sum_{t \in \mathcal{N}_u^2(s)} i_t}{16})$ , où  $i$  représente le niveau de gris du voxel  $s$ . Remarquons que ces trois propriétés ne sont pas équivalentes. En ne considérant que l'une d'entre elles, nous obtenons des fausses alarmes parmi le LCR pour la propriété (i), la matière blanche pour (ii), et les structures fines de la matière grise pour (iii). Les trois fonctions définies sont paramétrées par les moments statistiques des données et sont définies sur la figure 3.3. Ces trois fonctions sont combinées comme suit :

$$f_u(s) = \min \left( F^{black}(i_s), F^{white}(\min(i_t, t \in \mathcal{N}_u^2(s))), F^{contrast}(\frac{\sum_{t \in \mathcal{N}_u^2(s)} i_t}{16}) \right). \quad (3.31)$$

Pour éviter des détections multiples d'un même ERV, le filtre utilisé est défini comme suit :

$$F_u(s) = \begin{cases} f_u(s) & \text{si } \forall t \in \mathcal{N}_u^1(s), i_s \leq i_t \\ \min(0, f_u(s)) & \text{sinon} \end{cases} \quad (3.32)$$

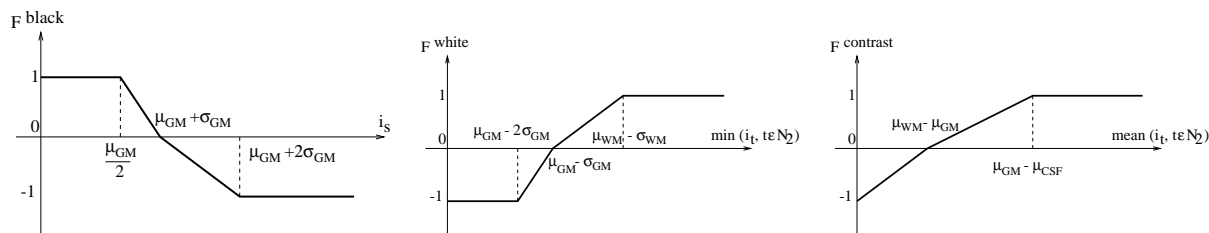


FIG. 3.3 – Les trois fonctions caractérisant les données pour un EVR :  $\mu_{CFS}$ ,  $\mu_{GM}$  et  $\mu_{WM}$  représentent les moyennes du LCR, de la matière grise et de la matière blanche,  $\sigma_*$  étant les écarts types associés.

Pour calculer les moyennes et écarts types du LCR, matières grise et blanche, nous utilisons une segmentation par croissance de régions [HK98].

**Processus Ponctuel marqué-** pour accroître la sensibilité de la détection et réduire le nombre de fausses alarmes nous considérons une information géométrique sur les EVR. En outre, les EVR ne sont pas répartis uniformément. L'information sur leur localisation obtenue par les études médicales est incorporée au modèle.

Le modèle complet  $h(x)$  consiste en un a priori  $f(x)$  et une attache aux données  $g(x)$  :

$$h(x) = f(x)g(x). \quad (3.33)$$

Chaque objet est représenté par un point dont l'attribut (ou marques) définit sa géométrie. La configuration résultante est un ensemble de points marqués  $X = \{ \overrightarrow{s(1)} = (x(1), l(1)), \dots, \overrightarrow{s(n)} = (x(n), l(n)) \}$  où  $x(i) \in V$  et  $\overrightarrow{l(i)}$  est un vecteur dans  $\mathbb{R}^3$ .



Le volume  $V$  sur lequel les points sont définis est l'espace des points vérifiant :

$$\max(F_u(x(i)), F_v(x(i)), F_w(x(i))) \geq 0,$$

où  $F_u, F_v$  et  $F_w$  sont les fonctions définies plus haut.

Nous considérons comme objets des segments orientés  $\vec{s}$  de points extrémaux  $x$  et  $x + \vec{l}$  où  $x \in V$  et  $\|\vec{l}\| \in [\rho_{min}, \rho_{max}]$ ,  $\rho_{min}$  (resp.  $\rho_{max}$ ) correspond au minimum (resp. maximum) de la longueur du segment. Nous définissons une distribution sur l'espace des configurations dont la densité par rapport à la mesure de Poisson est donnée par l'expression suivante :

$$f(X) \propto \beta^n \prod_{k \in \langle 1, n \rangle} q_1(\|\vec{l}(k)\|) \prod_{k, k' \in \langle 1, n \rangle : k \sim k'} q_2(\vec{s}(k), \vec{s}(k')), \quad (3.34)$$

où  $\sim$  définit une relation de voisinage.  $\beta$  est un paramètre de densité qui contrôle le nombre d'objets dans la configuration relativement aux réponses de l'attache aux données. Le terme  $q_1(\|\vec{l}(k)\|)$  est un a priori sur la longueur des segments défini comme suit :

$$q_1(\|\vec{l}(k)\|) = \exp - \left[ A \left( \frac{\rho_{max} - \|\vec{l}(k)\|}{\rho_{max} - \rho_{min}} \right)^2 \right]. \quad (3.35)$$

Pour nos expériences, nous avons pris  $\rho_{min} = 2$  et  $\rho_{max} = 15$ , d'après les longueurs des EVR observées dans nos données et dans la littérature. Le terme  $q_2(\vec{s}(k), \vec{s}(k'))$  définit des interactions entre segments voisins. Nous considérons trois types d'interactions : (i) un terme qui pénalise les intersections entre segments, (ii) un terme attractif qui favorise les segments voisins d'orientations similaires et (iii) un terme répulsif qui pénalise les segments voisins d'orientations différentes. Pour définir ces interactions, nous définissons la silhouette  $S(\vec{s})$  d'un segment  $\vec{s}$  comme sa projection sur la trame discrète. Pour pénaliser les intersections ("**interaction soft-core**"), nous utilisons :

$$q_2^{sc}(\vec{s}(k), \vec{s}(k')) = \exp[-B] \quad \text{si} \quad S(\vec{s}(k)) \cap S(\vec{s}(k')) \neq \emptyset. \quad (3.36)$$

Soient  $(s_u, s_v, s_w)$  les coordonnées de  $\vec{s}$ . Deux segments  $\vec{s}$  et  $\vec{s}'$  ont une direction u-similaire si et seulement si :

$$\begin{cases} |s_u| > \max(|s_v|, |s_w|) \\ |s'_u| > \max(|s'_v|, |s'_w|) \\ s_u \times s'_u > 0 \end{cases} \quad (3.37)$$

Si  $\vec{s}(k)$  et  $\vec{s}(k')$  ont une direction u-similaire, nous avons une interaction attractive :

$$q_2^{clu}(\vec{s}(k), \vec{s}(k')) = \exp [+C \min(|s_u(k)|, |s_u(k')|)] \quad \text{si} \quad |x_u(k) - x_u(k')| < c \quad (3.38)$$

Cette définition s'étend aux directions v-similaires et w-similaires. Finalement, pour deux segments n'ayant aucune direction similaire, nous définissons une interaction répulsive :

$$q_2^{rep}(\vec{s}(k), \vec{s}(k')) = \exp[-D] \quad \text{si} \quad \|x(k) - x(k')\| < d \quad (3.39)$$

Notons que  $q_2^{clu}(\vec{s}(k), \vec{s}(k'))$  et  $q_2^{rep}(\vec{s}(k), \vec{s}(k'))$  s'excluent mutuellement. Pour nos expériences, nous avons choisi  $c = 5$  et  $d = 3$  après avoir inspecté nos données.

Le terme d'attache aux données est défini par  $g(x)$ . En supposant les données indépendamment conditionnellement aux segments, nous avons :

$$g(x) \propto \prod_{k \in \langle 1, n \rangle} \vec{s}(k) \quad (3.40)$$

L'attache aux données d'un segment  $\vec{s}$  est définie en deux parties. La première est proportionnelle aux valeurs du filtre le long du segment. La seconde partie atteste que les voxels prolongeant les extrémités du segment n'appartiennent pas à l'EVR et ont donc une radiométrie plus forte. Ce terme s'écrit comme suit :

$$\begin{aligned}
 r(\vec{s}) &= \exp \left[ E \left( \sum_{t \in S(\vec{s})} F_u(t) + \frac{n_{\vec{s}}}{2} F_u^{end}(\vec{s}) \right) \right] & \text{si } |s_u| \geq \max(|s_v|, |s_w|) \\
 r(\vec{s}) &= \exp \left[ E \left( \sum_{t \in S(\vec{s})} F_v(t) + \frac{n_{\vec{s}}}{2} F_v^{end}(\vec{s}) \right) \right] & \text{si } |s_v| \geq \max(|s_u|, |s_w|) \\
 r(\vec{s}) &= \exp \left[ E \left( \sum_{t \in S(\vec{s})} F_w(t) + \frac{n_{\vec{s}}}{2} F_w^{end}(\vec{s}) \right) \right] & \text{si } |s_w| \geq \max(|s_u|, |s_v|)
 \end{aligned} \tag{3.41}$$

où  $n_{\vec{s}}$  est le nombre de voxels de  $S(\vec{s})$  et :

$$F_u^{end}(\vec{s} = (x, \vec{l})) = \begin{cases} F_{end}^{white} \left( \min \left( i_{x+\vec{l}+(1,0,0)}, i_t, t \in \mathcal{N}_u^1(i_{x+\vec{l}+(1,0,0)}) \right) \right) \\ + F_{end}^{white} \left( \min \left( i_{x+(-1,0,0)}, i_t, t \in \mathcal{N}_u^1(i_{x+(-1,0,0)}) \right) \right) & \text{si } \vec{l}_u > 0 \\ F_{end}^{white} \left( \min \left( i_{x+(1,0,0)}, i_t, t \in \mathcal{N}_u^1(i_{x+(1,0,0)}) \right) \right) \\ + F_{end}^{white} \left( \min \left( i_{x+\vec{l}+(-1,0,0)}, i_t, t \in \mathcal{N}_u^1(i_{x+\vec{l}+(-1,0,0)}) \right) \right) & \text{si } \vec{l}_u \leq 0 \end{cases} \tag{3.42}$$

où  $F_{end}^{white}(\cdot)$  est définie sur la figure 3.4. Des définitions similaires sont utilisées pour  $F_v^{end}(\vec{s})$  et  $F_w^{end}(\vec{s})$

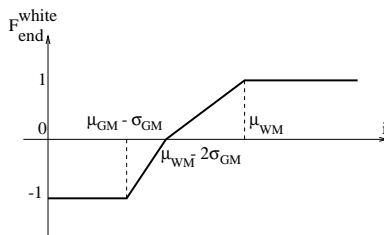
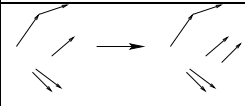
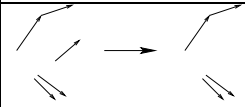
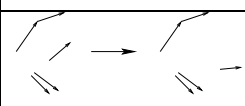
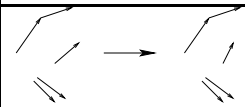
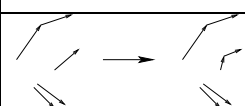
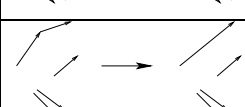
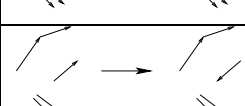


FIG. 3.4 – Le filtre de prolongement :  $\mu_{CFS}$ ,  $\mu_{GM}$  et  $\mu_{WM}$  représentent les moyennes du LCR et des matières grise et blanche,  $\sigma_*$  étant les écarts types associés

**Algorithme MCMC à sauts réversibles-** Nous considérons plusieurs types de mouvements dans le noyau de proposition. Le mouvement de naissance et mort est essentiel pour assurer le bon nombre de segments dans la configuration. En outre, il garantit l'irréductibilité de la chaîne de Markov. Ce mouvement est choisi avec une probabilité  $p$ . Il consiste à ajouter un segment avec une probabilité  $P_b$  ou à en supprimer un avec une probabilité  $P_d = 1 - P_b$ . Pour une mort, le segment est choisi uniformément parmi ceux de la configuration. Une naissance est choisie uniformément sur l'espace des paramètres. Pour éviter le faible taux d'acceptation d'une mort lorsque la configuration contient peu de segments, nous considérons un mouvement de changement de segment. Nous choisissons uniformément un segment dans la configuration et tirons aléatoirement une nouvelle position et de nouvelles valeurs pour les marques uniformément sur l'espace des paramètres. Ce mouvement est la combinaison d'une naissance et d'une mort, il est choisi avec la probabilité  $p$ . Ceci permet de déplacer les segments de faible probabilité. Pour affiner localement la position des segments, nous proposons de changer la position des points

Mouvement	Représentation	Probabilité	Taux d'acceptation
Naissance		$p_1 \times P_b$	$\frac{P_d}{P_b} \frac{h_T(x')}{h_T(x)} \frac{4\pi^2(\rho_{max}-\rho_{min})V}{n+1}$
Mort		$p_1 \times P_d$	$\frac{P_b}{P_d} \frac{h_T(x')}{h_T(x)} \frac{n}{4\pi^2(\rho_{max}-\rho_{min})V}$
Mvt d'un segment		$p_2$	$\frac{h_T(x')}{h_T(x)}$
Mvt d'un point terminal		$p_3$	$\frac{h_T(x')}{h_T(x)}$
Division		$p_4 \times P_s$	$\frac{P_m}{P_s} \frac{h_T(x')}{h_T(x)} \frac{(2D_m+1)^5( l_u -2*\rho_{min})}{n_m+1}$
Fusion		$p_4 \times P_m$	$\frac{P_s}{P_m} \frac{h_T(x')}{h_T(x)} \frac{n_m+1}{(2D_m+1)^5( l_u -2*\rho_{min})}$
Retournement		$p_5$	$\frac{h_T(x')}{h_T(x)}$

TAB. 3.1 – Les différents mouvements définissant le noyau de proposition:  $x$  est la configuration courante et  $x'$  celle proposée

terminaux. Ce mouvement est choisi avec une probabilité  $p_3$ . Une nouvelle localisation d'un point terminal est choisie uniformément dans un voisinage de la position courante. Ce mouvement permet également d'étendre des segments correspondant à des EVR partiellement détectés. Durant l'optimisation, un EVR peut être recouvert par plusieurs segments disjoints. L'extension de ses segments a alors un faible taux d'acceptation car cela introduirait des recouvrements de segments, ce qui est pénalisé par le modèle. Une mort suivie de l'extension du segment restant a également un faible taux d'acceptation car tous les segments présents correspondent bien aux données. Pour faciliter la dynamique, nous avons introduit un mouvement de fusion de segments voisins, et le mouvement inverse associé, la division. Ce mouvement est choisi avec la probabilité  $p_4$ . Une fusion est alors proposée avec la probabilité  $P_m$  et une division avec la probabilité  $P_s = 1 - P_m$ . Les segments ayant une orientation opposée ne peuvent être fusionnés du fait que le premier point  $x$  d'un segment  $(x, \vec{l})$  doit appartenir à  $V$  (ce mouvement pourrait théoriquement être défini mais conduirait à des calculs complexes pour le taux d'acceptation). Nous définissons donc un dernier mouvement qui consiste à retourner le segment  $(x, \vec{l})$  en  $(x + \vec{l}, -\vec{l})$ . Il est proposé avec une probabilité  $p_5 = 1 - (p_1 + p_2 + p_3 + p_4)$ . L'optimisation est effectuée en incluant l'échantillonneur MCMC dans un schéma de recuit simulé, avec une décroissance de la température géométrique.

**Résultats-** L'application visée est la détection des EVR. Comme les EVR sont des signes typiques chez les sujets âgés, nous avons sélectionné un sous-ensemble de données d'une étude clinique sur les dysfonctionnements cognitifs mineurs chez les personnes âgées. Toutes les données 3D ont été acquises sur un scanner Siemens Vision 1.5T avec un protocole MPRAGE pondérée  $T_1$  (TR 11.4 ms, TE 4.4 ms, 128 coupes sagittales, matrice 256x256, taille d'un voxel 0.9x0.9x1.5 mm). Une transformation affine par une interpolation utilisant des B-splines a été appliquée pour aligner les données avec le système de

coordonnées stéréotactiques à une résolution isotrope de 1 mm. Pour tous les tests, nous avons utilisé les mêmes jeux de paramètres, à savoir  $\beta = 1$ ,  $A = 10$ ,  $B = 1e10$ ,  $C = 20$ ,  $D = 60$  and  $E = 80$ . La forte valeur de  $B$  a été choisie pour interdire tout recouvrement entre segments, les autres paramètres ayant été fixés empiriquement à partir d'une IRM. Nous présentons en premier lieu le résultat sur cette IRM particulière. Cet exemple a été choisi parmi les sujets ayant un grand nombre d'EVR. La figure 3.5 montre le résultat sur une coupe axiale et les lésions détectées sont sur la figure 3.5.b. Les coupes coronale et sagittale d'une lésion particulière sont montrées sur les figures 3.5.c et 3.5.d. Une visualisation 3D du résultat est montrée sur la figure 3.6. Une analyse des formes obtenues montrent que les structures tubulaires ne sont pas très allongées. Nous détectons essentiellement la partie des lésions se trouvant dans la matière blanche. En effet, les lésions détectées n'atteignent pas la surface corticale. Une inspection visuelle montre que nous atteignons ici la limites des résolutions spatiales et radiométriques, ce qui rend non visible le prolongement des EVR dans la matière grise.

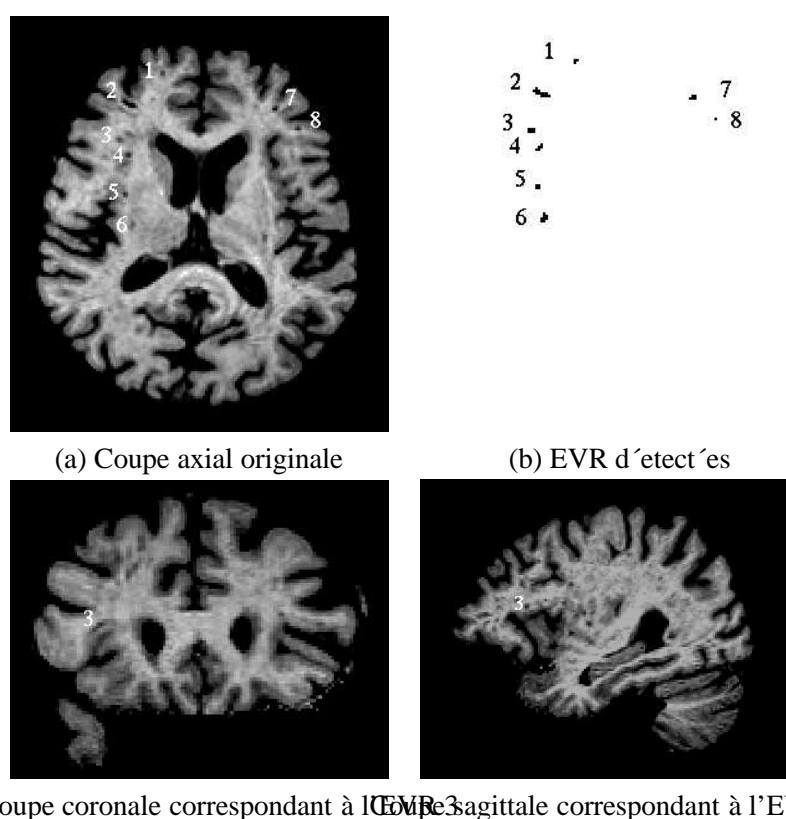


FIG. 3.5 – Exemple d'EVR détectés

Nous considérons maintenant une base de données de 37 sujets. Cette base a été classifiée par un expert neurologue en trois catégories, suivant les usages pour une telle étude, notées 1 pour peu d'EVR (9 cas), 2 pour les cas moyens (22 cas) et 3 pour beaucoup d'EVR (8 cas). Nous mettons en relation cette classification et le nombre d'EVR détectés. La figure 3.7 montre les distributions cumulées du nombre d'EVR détectés pour les groupes 1 (courbe à gauche), 2 (courbe du milieu) et 3 (courbe à droite). Le groupe 3 est clairement distingué. En revanche, il existe un certain recouvrement entre les groupes 1 et 2. Une inspection visuelle des données a montré a posteriori que ce recouvrement est en partie dû à un biais de l'expertise. Une régression linéaire entre la classification de l'expert et le nombre des EVR donne un coefficient de corrélation de 0.77 ( $p \sim 1e - 08$ ).

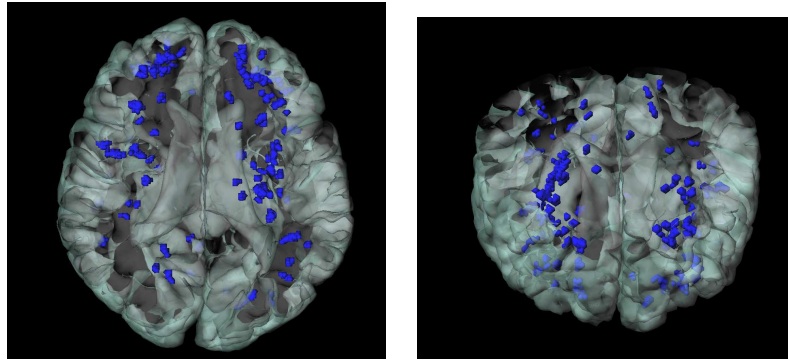


FIG. 3.6 – EVR détectés en relation avec la surface corticale sur l'exemple de la figure 3.5, vue supérieure et supérieure-frontale

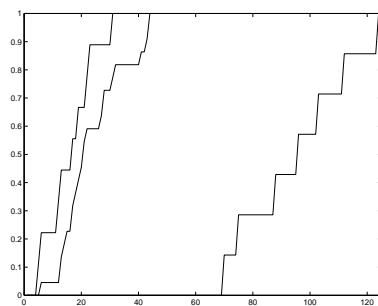


FIG. 3.7 – Distributions cumulées des EVR détectés en fonction de la classification de l'expert : classes 1 (courbe à gauche), 2 (courbe du milieu), 3 (courbe à droite)

### 3.3 Extraction du linéique

#### 3.3.1 Le Candy-modèle

<sup>5</sup> Traditionnellement, on distingue deux grands types de méthodes pour extraire le réseau routier. Le premier type concerne les méthodes semi-automatiques pour lesquelles des germes sont fournis par l'utilisateur, à partir desquels un algorithme de poursuite extrait le réseau [GJ96, MZ96, nPFISK97]. Ce type d'approche peut être rendu non supervisé par la détection automatique des germes [BC96, ZC93]. D'autres approches, dites automatiques, essaient de combiner différentes sources d'information ou différentes échelles en raison de la complexité du problème [MLBS97, FTW81]. Ici, nous abordons le problème à l'échelle de l'objet constitutif du réseau routier, c'est-à-dire en considérant des segments de route.

Nous introduisons deux modèles de processus ponctuels marqués pour l'extraction de linéique, notamment de réseaux routiers en vue d'une automatisation d'un processus cartographique. Le réseau linéique  $S$  est défini par la réalisation d'un processus Markov objet  $\mathcal{S}$  dont les objets sont des segments. Nous spécifions le processus par sa densité  $f$  relativement à un processus de Poisson homogène. La densité contient un terme d'attache aux données  $h_d$  et un terme a priori  $h_p$  contenant des contraintes sur la structure, la connexité et la courbure moyenne du réseau :

$$\boxed{f(S) \propto h_D(S) h_p(S)} \quad (3.43)$$

où  $S$  est la réalisation de  $\mathcal{S}$  et  $D$  la grille de pixels constituant l'image.

**Le "Candy" modèle-** Ce modèle comporte trois types de relations entre segments : la relation de connection  $R_c$  et deux relations de mauvaise orientation, interne  $R_{io}$  et externe  $R_{eo}$ .

Deux segments sont dits connectés si la distance la plus proche entre leurs extrémités est inférieure à une constante  $\epsilon$ . Cette relation permet de définir trois types de segments comme montré sur la figure 3.8.

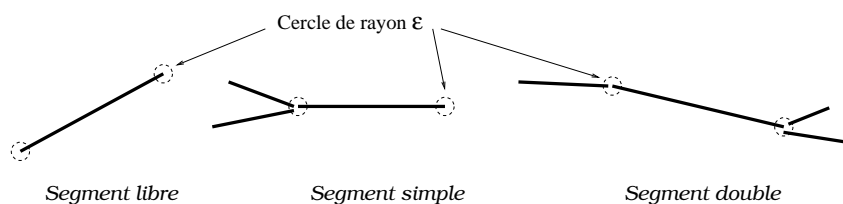


FIG. 3.8 – Les différents types de segments définis par la relation  $R_c$ .

Les segments libres sont les segments non connectés, les segments simples sont connectés à une seule extrémité et les segments doubles aux deux extrémités. La connectivité du réseau est modélisée dans la densité en pénalisant les segments libres et les segments simples. Pour éviter la superposition de segments ou les croisements à angle trop aigu tout en autorisant les croisements à angle proche de l'angle droit, nous considérons une interaction interne de mauvaise orientation,  $R_{io}$ . Les paires de segments dont les points centraux et les orientations sont trop proches sont pénalisées. Dans la partie gauche de la figure 3.9 la paire  $(s_1, s_2)$  est concernée par cette relation alors que la paire  $(s_4, s_3)$  ne l'est pas. La dernière relation  $R_{eo}$  est introduite pour contrôler la courbure du réseau. Les paires de segments concernées sont celles pour lesquelles les segments ont des extrémités proches. Dans ce cas, si les orientations sont trop différentes, comme les segments  $s_4$  et  $s_3$  dans la partie droite de la figure 3.9, la configuration est pénalisée (mauvaise orientation externe).

<sup>5</sup> Nous résumons ici le travail de thèse de R. Stoica, boursier MAE/INRIA, et de C. Lacoste, boursière BRGM/INRIA. Bibliographie : [71, 70, 49]

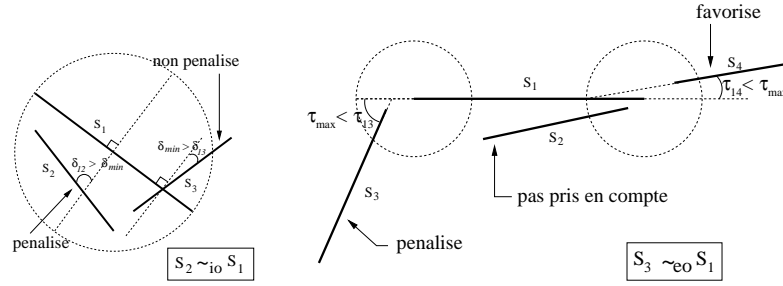


FIG. 3.9 – Mauvaise orientation interne  $R_{io}$  (à gauche) et externe  $R_{eo}$  (à droite)

Le Candy-modèle est donc un modèle d'interaction par paires (relations  $R_{io}$  et  $R_{eo}$ ) auquel est adjoint des termes d'endant du nombre de segments libres  $n_f$  et simples  $n_s$ . La densité a priori du Candy-modèle s'écrit donc comme suit :

$$h_p(S) \propto \beta^n \exp [-\omega_f n_f - \omega_s n_s - \omega_{io} n_{io} - \omega_{eo} n_{eo}] \quad (3.44)$$

avec :

$$\begin{cases} n & : \text{nombre total de segments,} \\ n_f & : \text{nombre de segments libres,} \\ n_s & : \text{nombre de segments simples,} \\ n_{io} & : \text{nombre de paires de segments en relation } R_{io}, \\ n_{eo} & : \text{nombre de paires de segments en relation } R_{eo}, \\ \omega_i & : \text{pondérations positives,} \\ \beta & : \text{facteur d'intensité.} \end{cases}$$

Cette densité spécifie un PPM bien défini au sens où la condition de stabilité de Ruelle est vérifiée tout comme la stabilité locale [vLS01]. En outre, le Candy modèle est un processus de Markov par rapport à la relation  $\sim$  définie par :

$$s \sim s' \iff d(p_s - p_{s'}) \leq 2(L_{max} + \epsilon)$$

où  $p_s$  est le point central du segment  $s$ , et  $d$  la distance euclidienne dans  $\mathbb{R}^2$ .

**Le Candy modèle à potentiels continus-** Nous généralisons le Candy modèle en remplaçant les potentiels constants définis pour les paires de segments en relation par des potentiels continus  $g$ . La densité a priori est alors définie comme suit :

$$h_p(S) \propto \beta^n \exp [-n_f \omega_f - n_s \omega_s - \sum_{r \in \mathcal{R}} \omega_r \sum_{\langle s_i, s_j \rangle_r} g_r(s_i, s_j)] \quad (3.45)$$

où :

$$\begin{cases} n, n_f, n_s & \text{sont définis comme précédemment,} \\ \langle \cdot, \cdot \rangle_r & \text{représente une paire de segments en interaction relativement à la relation } r, \\ g_r(\cdot, \cdot) & \equiv \text{fonction potentiel relativement à } r. \end{cases}$$

Nous redéfinissons la relation de connectivité  $R_c$  en ajoutant une contrainte qui accélère l'optimisation du modèle. Deux segments sont dits connectés si l'angle qu'ils forment est large. Sur la figure 3.11,  $s_1$  et  $s_3$  ne sont pas connectés. En outre, pour favoriser les paires de segments dont les extrémités et les orientations sont proches, comme pour la paire  $(s_1, s_2)$  de la figure 3.11, une fonction potentielle  $g_{R_c}$  est définie comme suit :

$$\text{pour } s_i \sim_c s_j, g_{R_c}(s_i, s_j) = \frac{g_\tau(\tau_{ij}) + g_\epsilon(d_{ij})}{2} \quad (3.46)$$

$$\text{avec } \begin{cases} g_\tau(\tau_{ij}) = \begin{cases} -\sigma(\tau_{ij}, \tau_{max}) & \text{si } |\tau_{ij}| < \tau_{max} \\ 1 & \text{sinon} \end{cases} \\ g_\epsilon(d_{ij}) = -\sigma(d_{ij}, \epsilon) \end{cases}$$

$g_\tau$  favorise les paires de segments  $(s_i, s_j)$  dont la différence d'orientation  $\tau_j$  est inférieure à un seuil  $\tau$  et pénalise les autres cas.  $g$  concerne la distance  $d_{ij}$  entre les extrémités et est de type attractif. Les termes attractifs de ces fonctions sont donnés par une fonction de qualité  $\sigma$ , d'écrite par l'équation (3.47) :

$$\begin{aligned} \sigma(\cdot, M) : [-M, M] &\longrightarrow [0, 1] \\ x &\longmapsto \sigma(x, M) = \frac{1}{M^2} \left( \frac{1 + M^2}{1 + x^2} - 1 \right) \end{aligned} \quad (3.47)$$

Cette fonction est positive sur  $[-M, M]$ , et est maximale en 0, comme montré sur la figure 3.10.

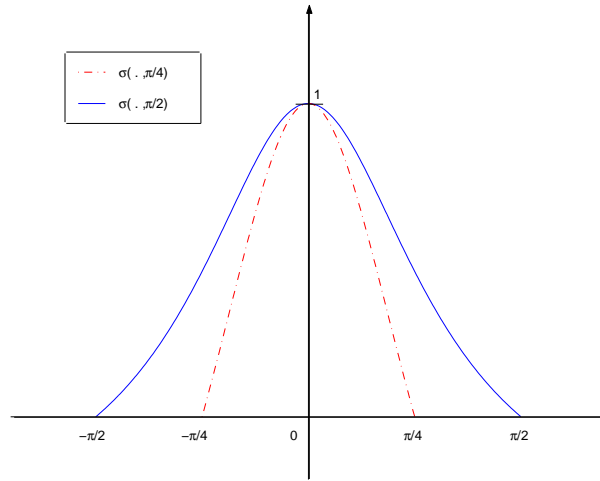


FIG. 3.10 – Fonction de qualité.

En outre, la partie répulsive de ce potentiel prend en compte les contraintes de la relation externe de mauvaise orientation  $R_{eo}$ , qui peut donc être omise dans ce nouveau modèle. La relation interne de mauvaise orientation  $R_{io}$  est conservée mais le potentiel associée est modifiée. Les paires de segments formant un angle inférieur à une (petite) constante, notée  $c$ , sont interdites pour des raisons de stabilité (potentiel de type “hard-core”). Pour les autres configurations, nous considérons la même fonction de qualité  $\sigma$  de caractère répulsif en fonction de la différence des orientations des deux segments, Ainsi, pour chaque paire  $(s_i, s_j)$  telle que  $s_i \sim_{io} s_j$ , nous considérons :

$$g_{R_{io}}(s_i, s_j) = \begin{cases} \infty & \text{if } \tau_{ij} < c \\ 1 - \sigma(\tau_{ij}, \pi/2 - \delta_{min}) & \end{cases} \quad (3.48)$$

où  $\tau_{ij}$  est la différence d'orientation entre  $s_i$  et  $s_j$ , et  $\delta_{min}$  est la différence minimale avec l'angle



droit pour que deux segments soient considérés comme mal orientés, et  $c$  est la différence minimale d'orientations qui peut se trouver dans la configuration.

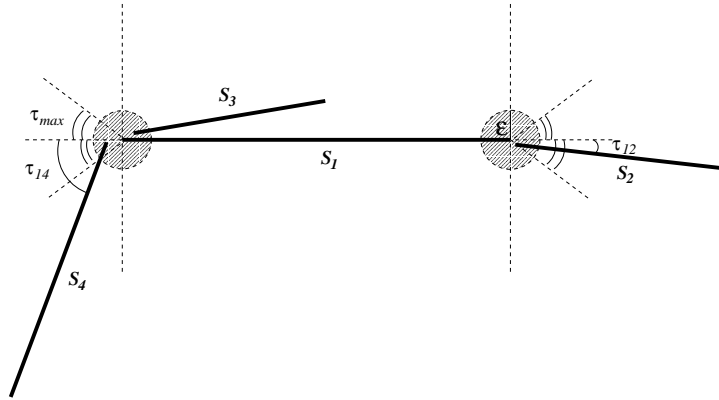


FIG. 3.11 – Différents types de connection -  $(s_1, s_2)$ : connection attractive -  $(s_1, s_3)$ : pas considérée comme une connection -  $(s_1, s_4)$ : connection répulsive par rapport à l'orientation.

**Modélisation des données-** Soit une configuration de segments  $S$  donnée, l'ensemble des pixels de chaque segment et de son voisinage sont supposés indépendants du reste de la configuration. Cette hypothèse permet de factoriser le terme d'attache aux données qui s'écrit alors :

$$h_D(S) \propto \exp \left[ -\gamma_d \sum_{s_i \in S} \delta_i \right] \quad (3.49)$$

où  $\delta_i$  est une valeur statistique calculée sur une région correspondant aux pixels du segment  $s_i$  et de son voisinage sur l'image  $D$ .  $\gamma_d$  est un coefficient de pondération positif.

Le potentiel  $\delta_i$  est fondé sur deux hypothèses par rapport au segment et à son voisinage. Nous supposons d'abord que les régions adjacentes sont différentes, au moins en moyenne, à la région correspondant au segment (voir figure 3.12 à gauche). Ensuite, l'ensemble des pixels d'un segment doit être homogène pour éviter que les contours soient détectés comme des routes (voir figure 3.12 à droite).

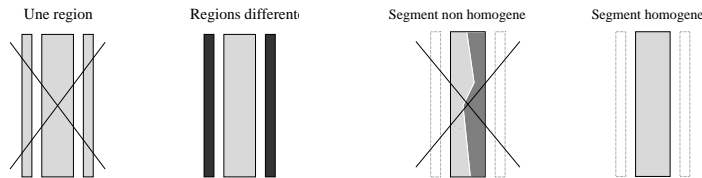


FIG. 3.12 –  $H_1$  - Différence significative avec les zones adjacentes (à gauche),  $H_2$  - Homogénéité du segment (à droite).

Pour vérifier ces hypothèses pour un segment donné, nous le divisons en plusieurs bandes  $b_1, \dots, b_n$ . Par ailleurs, deux bandes sur chaque côté  $R_1^i$  et  $R_2^i$ , à une distance  $d$  du segment  $s_i$  sont considérées, comme illustré sur la figure 3.13. Les valeurs des pixels de chaque bande constituent une population. Un t-test de Student est calculé pour déterminer si les moyennes de deux populations sont significativement différentes. Pour deux populations  $x$  et  $y$ , nous calculons donc :

$$\text{t-test}(x, y) = \frac{|\bar{x} - \bar{y}|}{\sqrt{\frac{\tilde{\sigma}_x}{n_x} + \frac{\tilde{\sigma}_y}{n_y}}} \quad (3.50)$$

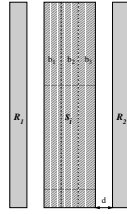


FIG. 3.13 – Division d'un segment en plusieurs bandes.

où  $\bar{x}$ ,  $\tilde{\sigma}$  et  $n$  font respectivement référence à la moyenne, l'écart type et le nombre des observations. Nous voulons une différence significative avec les deux cotés du segments, nous considérons donc le test statistique ( $H_1$ ) défini par le minimum des deux tests :

$$T_{H_1}(s_i) = \min_{l \in \{1,2\}} [ \text{t-test}(R_l^i, s_i) ] \quad (3.51)$$

L'hypothèse d'homogénéité est donnée par la valeur maximale des tests entre deux bandes adjacentes internes au segment :

$$T_{H_2}(s_i) = \max_{j \in \{1, \dots, n_b - 1\}} [ \text{t-test}(b_j, b_{j+1}) ] \quad (3.52)$$

Pour finir, le potentiel retenu est le rapport de ces deux quantités, en ayant pris la précaution de minorer  $T_{H_2}(s_i)$  :

$$T_i = \frac{T_{H_1}(s_i)}{\max [ 1, T_{H_2}(s_i) ]} \quad (3.53)$$

En outre, une fonction sigmoïde permet de convertir les valeurs de  $[0, \infty]$  dans  $[-1, 1]$ . Nous avons donc :

$$\delta_i = \begin{cases} 1 & \text{si } T_i < t_1 \\ 1 - 2 \frac{T_i - t_1}{t_2 - t_1} & \text{si } t_1 < T_i < t_2 \\ -1 & \text{si } T_i > t_2 \end{cases} \quad (3.54)$$

où  $t_1$  et  $t_2$  ( $t_1 < t_2$ ) sont deux seuils paramétrant l'attache aux données.

**Noyau de proposition-** Dans le noyau de proposition, nous considérons en premier lieu un mouvement de naissance-mort. Il consiste en une naissance uniforme sur le compact  $F \subset \mathbb{R}^2$  proposée avec une probabilité  $p$  et une mort uniforme sur l'ensemble des segments de  $S$  proposée avec une probabilité  $1 - p_b$ . Nous considérons, ensuite, la modification d'un objet quelconque suivant une transformation symétrique. Soit  $\mathcal{T} = \{T_a : a \in E\}$  une famille de transformations paramétrée par un vecteur  $a \in E$ . L'objet  $s$  est modifié par la transformation  $T_a$  :

$$s \leftarrow T_a(s)$$

où  $a$  est choisi uniformément dans  $E$ .

Nous considérons la famille des rotations dans  $[-\Delta_\theta, \Delta_\theta]$ , une rotation  $T_{d_\theta}$  étant définie par :

$$T_{d_\theta} \left( s = \begin{bmatrix} x \\ y \\ L \\ \theta \end{bmatrix} \right) = \begin{bmatrix} x \\ y \\ L \\ (\theta + d_\theta) [\pi] \end{bmatrix}$$

où  $d_\theta \in [-\Delta_\theta, \Delta_\theta]$  est la différence des orientations du segments  $s$  et de sa transformée  $\mathcal{T}_\theta(s)$ , et  $[\cdot]$  représente la fonction modulo.

De même, nous considérons la famille des translations, paramétrée par le vecteur  $[dx, dy]$ ,  $dx \in [-\Delta_x, \Delta_x]$ ,  $dy \in [-\Delta_y, \Delta_y]$ :

$$T_{d_{[dx, dy]}} \left( \begin{bmatrix} x \\ y \\ L \\ \theta \end{bmatrix} \right) = \begin{bmatrix} (x + dx) [X_{max}] \\ (y + dy) [Y_{max}] \\ L \\ \theta \end{bmatrix}$$

et les homothéties paramétrées par  $L \in [-\Delta_L, \Delta_L]$ :

$$T_{d_L} \left( \begin{bmatrix} x \\ y \\ L \\ \theta \end{bmatrix} \right) = \begin{bmatrix} x \\ y \\ L_{min} + (L - L_{min} + d_L) [L_{max} - L_{min}] \\ \theta \end{bmatrix}$$

Pour proposer des configurations de forte probabilité, nous considérons également un noyau de naissance dans un voisinage d'un segment de la configuration.

**Résultats-** L'algorithme RJMCMC est intégrée dans un schéma de recuit simulé pour les tests sur des données réelles. La première image (voir figure 3.14) est une image SPOT. Le réseau routier principal est très linéaire et bien contrasté, excepté pour la partie en bas à droite de l'image. Les seuils  $t_1$  et  $t_2$  du potentiel d'attache aux données sont  $t_1 = 4$  et  $t_2 = 8$ .

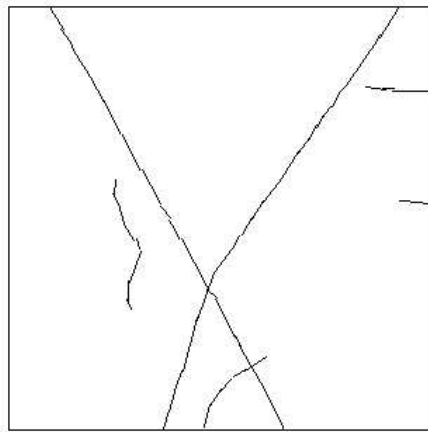


FIG. 3.14 – Test 1 : image SPOT (256 × 256 pixels).

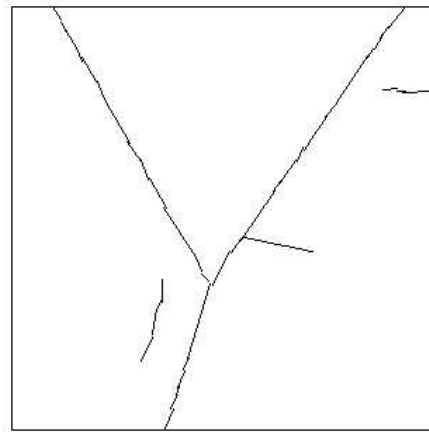
La figure 3.15 montre les résultats obtenus pour deux vitesses de décroissance de la température. Les résultats sont sensibles au schéma de décroissance de la température ; une décroissance trop rapide de celle-ci aboutit à un résultat incomplet (voir la figure 3.15(b)).

En revanche, la figure 3.16 montre que le Candy modèle à potentiels continus est plus robuste vis-à-vis de la décroissance de la température, ce qui permet un gain en temps de calcul.

Nous considérons maintenant des images plus complexes. L'image aérienne de la figure 3.17 a un réseau partiellement obstrué par les ombres des arbres. En outre, la texture de certains champs peut

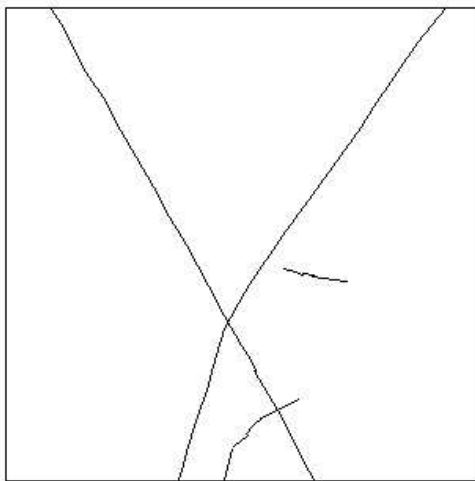


(a) D'écroissance lente de la température

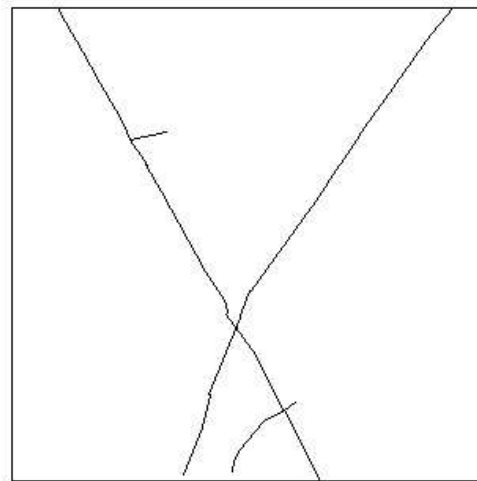


(b) D'écroissance rapide de la température

FIG. 3.15 – Résultats de l'extraction du réseau routier sur l'image SPOT de la figure 3.14 pour le Candy modèle avec une décroissance géométrique de la température de taux : (a)  $c = 0.999999$  - (b)  $c = 0.99999$ .



(a) D'écroissance lente de la température



(b) D'écroissance rapide de la température

FIG. 3.16 – Résultats de l'extraction du réseau routier sur l'image SPOT de la figure 3.14 pour le Candy modèle à potentiels continus avec une décroissance géométrique de la température de taux : (a)  $c = 0.999999$  - (b)  $c = 0.99999$ .



FIG. 3.17 – Données 2 : image aérienne ( $892 \times 652$  pixels).

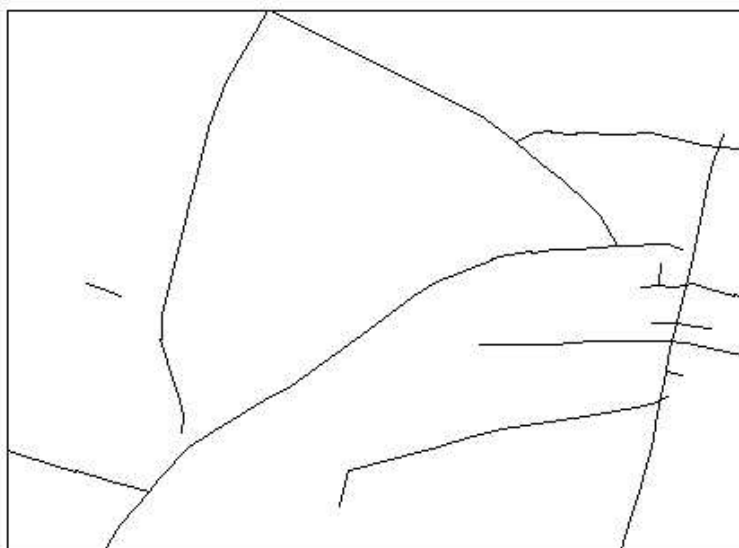


FIG. 3.18 – Résultat de l'extraction du réseau routier de l'image 3.17 par le Candy modèle à potentiels continus.

amener à de fausses alarmes. Les seuils du terme d'attache aux données sont ici  $t_1 = 8$  et  $t_2 = 15$ . Un résultat très satisfaisant est obtenu avec le Candy modèle à potentiels continus (voir figure 3.18).

Pour finir, nous considérons une image radar ERS1. La difficulté provient ici du bruit de chatolement (voir figure 3.19). Pour cette image, nous avons  $t_1 = 3$  et  $t_2 = 9$  pour prendre en compte le fort taux de bruit. Les deux modèles restent robustes malgré ce bruit.

### 3.3.2 Estimation des paramètres

<sup>6</sup> Nous n'avons pas encore abordé le problème de l'estimation des paramètres associés aux processus ponctuels marqués. Nous montrons ici que l'approche MVMCMC présentée au paragraphe 2.2.3 se généralise dans ce cadre. Dès lors, il peut être intéressant de formuler les approches par processus ponctuels marqués dans un cadre bayésien, quand cela s'avère possible, pour dériver des algorithmes, de type EM, totalement non-supervisés.

Reprenons le Candy modèle et écrivons le sous la forme d'une famille exponentielle à cinq paramètres :

$$p_\omega(\mathbf{s}) = \alpha(\omega) \exp [t(\mathbf{s})^T \omega] h(\mathbf{s}) \quad (3.55)$$

de statistique canoniques suffisantes :

$$t(\mathbf{s}) = (n_f(\mathbf{s}), n_s(\mathbf{s}), n_d(\mathbf{s}), n_{io}(\mathbf{s}), n_{eo}(\mathbf{s}))^T, \quad (3.56)$$

et de paramètres :

$$\omega = (\log \gamma_f, \log \gamma_s, \log \gamma_d, \log \gamma_{io}, \log \gamma_{eo})^T, \quad (3.57)$$

avec  $h(\mathbf{s}) = \prod_{i=1}^n \frac{l_i - l_{\max}}{l_{\max}}$ . Rappelons que  $n_f(\mathbf{s})$ ,  $n_s(\mathbf{s})$ ,  $n_d(\mathbf{s})$ ,  $n_{io}(\mathbf{s})$ ,  $n_{eo}(\mathbf{s})$  représentent le nombre de segments libres, le nombre de segments connectées à une seule extrémité, le nombre de segments connectés aux deux extrémités le nombre de paires de segments se croisant avec un angle trop aigu et le nombre de paires de segments voisins mal orientés.

Par échantillonnage préférentiel, le rapport des constantes de normalisation peut s'écrire :

$$\alpha(\omega_0)/\alpha(\omega) = E_{\omega_0} \exp [t(\mathbf{s})^T (\omega - \omega_0)] \quad (3.58)$$

et le rapport des log-vraisemblances relativement à une valeur de référence  $\omega_0$  pour une configuration  $\mathbf{s}^*$  s'écrit comme suit :

$$\begin{aligned} l(\omega) &= \log \frac{p_\omega(\mathbf{s}^*)}{p_{\omega_0}(\mathbf{s}^*)} \\ &= t(\mathbf{s}^*)^T (\omega - \omega_0) - \log E_{\omega_0} \exp [t(\mathbf{s})^T (\omega - \omega_0)]. \end{aligned} \quad (3.59)$$

L'équation "score"  $\nabla l(\omega) = t(\mathbf{s}^*) - E_{\omega_0} t(\mathbf{s})$  et la matrice d'information de Fisher  $I(\omega) = -\nabla^2 l(\omega) = \text{Var}_{\omega_0} t(\mathbf{s})$  se calculent facilement et l'estimée au sens du maximum de vraisemblance est obtenue lorsque les espérances des statistiques suffisantes sont égales aux valeurs observées. Comme la matrice de covariance de  $t(\mathbf{s})$  est définie positive, la log-vraisemblance donnée par l'équation (3.59) est concave dans  $\omega$ . Dès lors, dès que l'équation "score" a une solution  $\hat{\omega} \in \mathbb{R} \times \mathbb{R}^4$ , il existe une unique estimée au sens du maximum de vraisemblance  $\hat{\omega}$ . Dans le cas contraire, un maximum peut être obtenu sur le bord de l'espace des paramètres.

Numériquement, l'espérance dans l'équation (3.59) peut être approximée par une estimée de type Monte Carlo :

$$\frac{1}{n} \sum_{i=1}^n \exp [t(\mathbf{s}^{(i)})^T (\omega - \omega_0)] \quad (3.60)$$

<sup>6</sup> Nous résumons ici un travail, soutenu par ERCIM, effectuée en collaboration avec R. Stoica et M.N.M. Van Lieshout du CWI à Amsterdam. Bibliographie : [37]

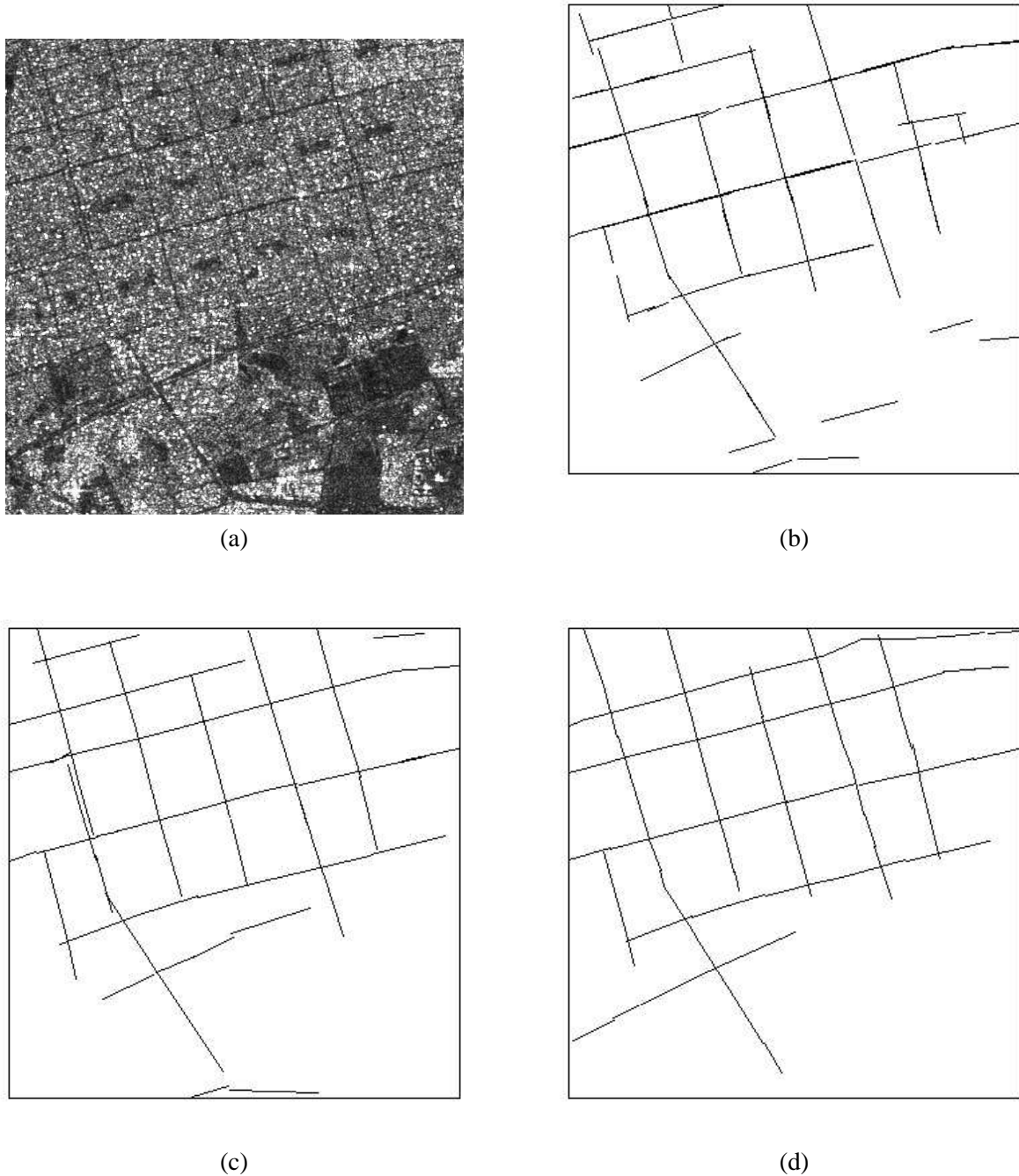


FIG. 3.19 – Résultats de l'extraction du réseau routier à partir d'une image radar : (a) Test 3 : image radar ERS ( $525 \times 546$  pixels), (b) Candy modèle avec un fort a priori, (c) Candy modèle à potentiels continus avec un faible a priori et (d) Candy modèle à potentiels continus avec un fort a priori.

fondée un échantillonnage de  $s^{(1)}, \dots, s^{(n)}$  de  $p_{\omega_0}$ .

Considérant la véritable inconnue  $\hat{\omega}$  (estimateur au sens du maximum de vraisemblance), l'estimée du maximum de vraisemblance par Monte Carlo est consistante et satisfait au théorème central limite :

$$\sqrt{(n)}(\hat{\omega}^n - \hat{\omega}) \rightarrow \mathcal{N}(0, I(\hat{\omega})^{-1} \Sigma I(\hat{\omega})^{-1}) \quad (3.61)$$

où  $\Sigma$  est la matrice de covariance asymptotique du “score” normalisé de Monte Carlo  $\sqrt{(n)}\nabla l_n(\hat{\omega})$  et  $I(\hat{\omega})$  est la matrice d'information de Fisher de l'estimateur du maximum de vraisemblance.

Cette méthode s'appuie sur une valeur de référence qui ne doit pas être trop éloignée de la solution. Nous utilisons le schéma de gradient itératif suivant :

$$\begin{cases} \rho(\omega_k) = \arg \max_{\rho \in \mathbb{R}} l_n(\omega_k + \rho \nabla l_n(\omega_k)) \\ \omega_{k+1} = \omega_k + \rho(\omega_k) \nabla l_n(\omega_k) \end{cases} \quad (3.62)$$

pour obtenir une valeur raisonnable.  $\rho(\omega_k)$  est le pas optimal, calculé par une minimisation mono-dimensionnelle de la fonction de vraisemblance.

Les résultats obtenus sur un exemple sont résumés dans le tableau de la figure 3.20. Nous avons effectué 1000 itérations de l'équation (3.62) pour obtenir le vecteur décrit dans la troisième colonne. Ensuite, à partir d'un ensemble d'échantillons de  $p_{\omega_0}$  de taille  $n = 2 \times 10^7$ , nous calculons l'approximation de Monte Carlo  $l_n(\omega)$ . Le maximum de  $l_n(\omega)$  est obtenu pour  $\hat{\omega}^n$  donné dans la quatrième colonne de la figure 3.20.

Vraies valeurs	Paramètres initiaux	Méthode Itérative	Estimée du MV par Monte Carlo
$\omega_f^i = -8.52$	$\omega_f^i = -9.5$	$\hat{\omega}_f^0 = -8.37$	$\hat{\omega}_f^n = -8.32$
$\omega_s^i = -2.99$	$\omega_s^i = -4.0$	$\hat{\omega}_s^0 = -2.74$	$\hat{\omega}_s^n = -2.73$
$\omega_d^i = 2.5$	$\omega_d^i = 1.5$	$\hat{\omega}_d^0 = 2.46$	$\hat{\omega}_d^n = 2.47$
$\omega_o^i = -2.53$	$\omega_o^i = -3.5$	$\hat{\omega}_o^0 = -2.13$	$\hat{\omega}_o^n = -2.17$
$\omega_r^i = -3.5$	$\omega_r^i = -3.5$	$\hat{\omega}_r^0 = -2.42$	$\hat{\omega}_r^n = -2.53$

FIG. 3.20 – Exemple d'estimation des paramètres.

Cet exemple montre qu'il est donc envisageable de développer des algorithmes totalement non supervisés en analyse d'image à partir d'une modélisation par processus ponctuels marqués.

## 3.4 Extraction de surfacique

### 3.4.1 Extraction des houppiers

<sup>7</sup> Même si le problème de l'extraction automatique des arbres reste, dans sa généralité, largement ouvert, certains travaux ont été proposés. Nous supposons ici que nous n'avons pas accès à des données de type modèle numérique d'élévation, qui simplifient énormément le problème. Une première approche, proposée dans la littérature, consiste à assimiler un arbre à chaque maximum local de l'image et à délimiter les zones d'influence des différents maxima par un algorithme de suivi de vallées [Gou95, Gou98]. D'autres approches sont fondées sur la corrélation de l'image avec un modèle optique des arbres par une procédure de type “template matching” [LR97, Lar99]. Nous considérons ici une approche objet

7. Nous résumons ici le travail de DEA de G. Perrin dans le cadre d'une action COLORS de l'INRIA Sophia Antipolis



permettant de prendre en compte des informations sur la géométrie et la radiométrie des houppiers ainsi que sur la répartition spatiale de l'ensemble des arbres.

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 certains, 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 photographies aériennes que nous utilisons, et sur lesquelles travaillent les experts de l'*Inventaire Forestier National* (IFN), sont des clichés analogiques de  $23\text{cm} \times 23\text{cm}$ , que l'on peut scanner jusqu'à une précision de 50cm au sol. C'est la résolution qui est atteinte pour nos images, scannées avec une définition de 600 dpi (dots per inch).

Les 3 bandes de fréquences sont situées entre 520 et 900  $\mu\text{m}$ , le bleu est donc filtré. Une des bandes spectrales des capteurs aériens est la bande proche infra-rouge (700 à 900  $\mu\text{m}$ ), 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  $\mu\text{m}$  (vert-jaune) et de 620 à 700  $\mu\text{m}$  (rouge).

La figure 3.21 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 figure 3.22 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 dite de taillis sous futaie, c'est à dire un étage supérieur, la futaie, composée ici de chênes principalement, et un étage inférieur, sans doute composée de charmes et de chênes. Cette structure étagée est classique dans nos forêts.

Dans un premier temps, il a été convenu avec l'IFN et le Cemagref, 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 révélera très utile dans la mise en place de notre modèle.

**Espace Objet-** Sur la partie droite de l'image de la figure 3.22, 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.



FIG. 3.21 – Image de Saône et Loire



FIG. 3.22 – Image de Saône et Loire : à gauche des taillis sous futaies principalement composés de chênes, à droite des peupliers.

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.

Les images de données étant finies, les coordonnées des objets seront inscrites dans un pavé de  $\mathbb{R}^3$ . On note  $\mathcal{P}$  cet espace des positions :  $\mathcal{P} = [0, X_M] \times [0, Y_M]$ . Il en sera de même pour les marques des objets, c'est à dire les rayons des cercles que nous pouvons supposer bornés. L'espace des marques, noté  $\mathcal{K}$ , est alors simplement  $\mathcal{K} = [R_m, R_M]$ .

L'espace objet a donc la forme suivante :

$$U = \mathcal{P} \times \mathcal{K} = [0, X_M] \times [0, Y_M] \times [R_m, R_M]. \quad (3.63)$$

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

**Terme a priori-** Le terme a priori est défini de façon à contrôler le nombre d'objets et leur plus ou moins grand recouvrement. En effet, nous désirons limiter la superposition des objets tout en conservant une certaine flexibilité pour d'éventuelles légères intersections entre objets voisins.

Un terme de contrôle sur l'intensité du processus, et donc influant sur l'espérance du nombre d'objets détectés, est défini en premier lieu. Il s'écrit simplement  $\beta$ , avec  $\beta > 1$  si l'on désire favoriser la naissance d'objets dans la configuration, ou  $\beta < 1$  dans le cas contraire. Pour éviter de biaiser le terme de vraisemblance à basse température, ce terme est inclus dans la mesure de référence  $\mu(\cdot)$  et non dans la densité. Il n'est donc pas concerné par le paramètre de température lors du recuit simulé, ce qui fait que son influence diminue avec la température. C'est donc un terme essentiellement utile pour régler la dynamique du processus.

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 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 où les arbres sont très proches les uns des autres.

Afin de caractériser dans l'espace des objets  $U$  ce phénomène de superposition, nous introduisons quelques notions telles qu'une relation d'équivalence d'intersection sur  $U$ , et la silhouette d'un objet ou d'une configuration.

Tout d'abord, nous définissons un lien entre l'espace  $U$  et  $\mathbb{R}^2$ , espace dans lequel ont lieu ces intersections.

**Définition 13.** On appelle silhouette d'un objet  $u \in U$  l'ensemble  $\mathcal{S}_{\mathcal{P}}(u) \in \mathcal{P} \subset \mathbb{R}^2$ , intersection de l'espace des positions et de la boule fermée dans  $\mathbb{R}^2$  (ou disque), de centre  $C_u = (X_u, Y_u)$  et de rayon  $R_u$  :

$$\mathcal{S}_{\mathcal{P}}(u) = \mathbb{B}_f(C_u, R_u) \cap \mathcal{P}.$$

La silhouette d'un objet  $u$  n'est donc pas toujours un disque, celui-ci peut en effet se retrouver tronqué si le centre  $C_u \in \mathcal{P}$  est à une distance inférieure à  $R_u$  des limites de l'espace des positions. Par extension, on définit la notion de silhouette d'un ensemble d'objets, ou d'une configuration d'objets.

**Définition 14.** On appelle silhouette d'un ensemble d'objets  $\mathcal{U} = (u_1, \dots, u_n)$  (respectivement d'une configuration d'objets  $\mathbf{x} = (x_1, \dots, x_n)$ ), et on note  $\mathcal{S}_{\mathcal{P}}(\mathcal{U})$  (resp.  $\mathcal{S}_{\mathcal{P}}(\mathbf{x})$ ), l'ensemble défini par la

réunion des silhouettes de chacun des objets :

$$\mathcal{S}_{\mathcal{P}}(\mathcal{U}) = \bigcup_{u \in \mathcal{U}} \mathcal{S}_{\mathcal{P}}(u).$$

$$\mathcal{S}_{\mathcal{P}}(\mathbf{x}) = \bigcup_{u \in \mathbf{x}} \mathcal{S}_{\mathcal{P}}(u).$$

On d'efinit ensuite une *relation de voisinage*  $\sim_I$ , symétrique et réflexive, qui met en relation deux objets de  $U$  dont les silhouettes s'intersectent sur un ensemble non réduit à un point :

$$x_i \sim_I x_j \Leftrightarrow d(C_{x_i}, C_{x_j}) < R_{x_i} + R_{x_j},$$

où  $d(\dots)$  est la distance euclidienne sur  $\mathbb{R}^2$ .

**Définition 15.** Soient  $\mathbf{x}$  une configuration d'objets sur  $U$ , et  $\sim$  une relation d'équivalence définie sur  $U$ . On appelle *voisinage* d'un objet  $u \in \mathbf{x}$  au sens de la relation  $\sim$  l'ensemble

$$\mathcal{V}_{\sim, \mathbf{x}}(u) = \{x_i \in \mathbf{x} \mid x_i \neq u, x_i \sim u\}.$$

Ainsi, le voisinage  $\mathcal{V}_{\sim_I, \mathbf{x}}(u)$  sera constitué de l'ensemble des voisins de  $u$  au sens de  $\sim_I$ .

On note alors  $n_{\sim_I}(\mathbf{x})$  le nombre de paires d'objets (ou cliques) d'une configuration  $\mathbf{x}$  voisins au sens de  $\sim_I$ . D'après cette définition, ainsi que celle du voisinage, on a :

$$\begin{aligned} n_{\sim_I}(\mathbf{x}) &= \sum_{x_i, x_j \in \mathbf{x}, x_i \neq x_j} \mathbf{1}_{x_i \sim_I x_j} \\ &= \frac{1}{2} \sum_{u \in \mathbf{x}} \#(\mathcal{V}_{\sim_I, \mathbf{x}}(u)). \end{aligned}$$

**Définition 16.** On appelle *ensemble d'intersection* d'un objet  $u$  dans une configuration  $\mathbf{x}$ ,  $u \in \mathbf{x}$ , et on note  $\cap_{\mathbf{x}}(u)$  l'ensemble :

$$\cap_{\mathbf{x}}(u) = \mathcal{S}_{\mathcal{P}}(u) \cap \mathcal{S}_{\mathcal{P}}(\mathcal{V}_{\sim_I, \mathbf{x}}(u)).$$

Par définition du voisinage  $\mathcal{V}_{\sim_I, \mathbf{x}}(u)$ , on a aussi :

$$\cap_{\mathbf{x}}(u) = \mathcal{S}_{\mathcal{P}}(u) \cap \mathcal{S}_{\mathcal{P}}(\mathbf{x})$$

puisque seuls les voisins de  $u$  au sens de  $\sim_I$  l'intersectent.

Cette dernière définition nous permet ainsi de caractériser, pour chaque objet  $u$  d'une configuration, le sous-ensemble de  $\mathcal{P}$  où sa silhouette intersecte celles de ses voisins.

Pour contrôler les intersections entre objets nous pénalisons l'aire de l'ensemble d'interaction de chaque objet avec la configuration  $\mathbf{x}$  :

$$\mathcal{A}(\mathbf{x}) = \sum_{u \in \mathbf{x}} \lambda_2(\cap_{\mathbf{x}}(u)) \tag{3.64}$$

$h_p(\cdot)$  s'écrit alors :

$$h_p(\mathbf{x}) = e^{\gamma_a \mathcal{A}(\mathbf{x})} \quad (3.65)$$

**Vraisemblance-** Le terme de vraisemblance  $\mathcal{L}(\mathcal{J}|\mathbf{x})$  correspond à la probabilité, étant donnée une configuration d'objets  $\mathbf{x}$ , d'observer une image  $\mathcal{J}$ . Nous distinguons 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. L'image de données étant discrète, 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  $i^{\text{ème}}$  colonne et de la  $j^{\text{ème}}$  ligne, recouvre le sous-ensemble  $[i, i+1] \times [j, j+1]$  de  $\mathcal{P}$ .

**Définition 17.** 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  $(i + \frac{1}{2}, j + \frac{1}{2})$ .

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

**Définition 18.** On appelle représentant sur l'image  $\mathcal{J}$  de  $x_i = (X_{x_i}, Y_{x_i}, R_{x_i})$ , l'ensemble  $\mathcal{T}_{\mathcal{J}}(x_i)$  des pixels de  $\mathcal{J}$  ayant leur centre dans le support de  $x_i$ .

Si les  $x_i$  sont des cercles, ces pixels vérifient l'inégalité suivante :

$$\sqrt{(i + \frac{1}{2} - X_{x_i})^2 + (j + \frac{1}{2} - Y_{x_i})^2} \leq R_{x_i} \implies (i, j) \in \mathcal{T}_{\mathcal{J}}(x_i). \quad (3.66)$$

On peut alors attribuer à chacun des pixels de l'image  $\mathcal{J}$  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  $\mathcal{T}_{\mathcal{J}}(x_i)$  des objets de la configuration  $\mathbf{x}$  :

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

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. 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.

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

$$\mathcal{L}(\mathcal{J}|\mathbf{x}) = \prod_{p \in \mathcal{C}_A} \frac{1}{\sqrt{2\pi}\sigma_A} \exp\left(-\frac{(y_p - \mu_A)^2}{2\sigma_A^2}\right) \prod_{p \in \mathcal{C}_F} \frac{1}{\sqrt{2\pi}\sigma_F} \exp\left(-\frac{(y_p - \mu_F)^2}{2\sigma_F^2}\right). \quad (3.67)$$

**Noyau de propositions-** En th eorie, l'algorithme de Metropolis-Hastings-Green, pourvu des mouvements de naissance et de mort seulement, converge vers la loi objectif. Les propri etes math ematiques sont en effet toutes v erifi ees. Dans la pratique, le noyau de propositions est plus complexe pour nous permettre d'acc el erer cette convergence, d'ej a ralentie par l'utilisation d'un recuit simul e et de la lente d'ecroissance des temp eratures.

Notre objectif est, en effet, de mod eliser des noyaux de propositions adapt es  a la loi objectif, afin d'eviter au maximum des mouvements inutiles et de minimiser le taux de rejet. Ces mouvements seront group es en deux cat egories dans cette partie : les mouvements *simples*, c'est  a dire ceux modifiant un des objets de la configuration courante, et les mouvements *complexes*, qui changent le nombre d'objets de la configuration (comme la naissance et la mort).

Soient  $\mathbf{x}$  une configuration, et  $A \in \mathcal{B}(\Psi)$ . Le noyau de propositions  $Q(\mathbf{x}, A)$ , d'efini sur  $\Psi \times \mathcal{B}(\Psi)$ , peut se d ecomposer sous la forme d'une somme de noyaux de propositions :

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

o u chacun des noyaux  $Q_i(\mathbf{x}, \cdot)$  d'ecrit un mouvement, choisi avec la probabilit e  $p_i$

Principalement deux mouvements simples n'ecessitent d'etre 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, ils font donc gagner du temps  a l'algorithme, qui n'a plus besoin d'accepter ces deux mouvements, mais juste un seul.

*Translation-* 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 coordonn ees  $(X_{x_i}, Y_{x_i})$  dans l'espace des positions  $\mathcal{P}$ .

Pour ce faire, nous introduisons une variable al eatoire  $\theta$  sur un compact sym etrique de  $\mathbb{R}^2$  : ainsi, si  $\theta \in \Theta$ , alors  $-\theta \in \Theta$ . Cette variable al eatoire suit une loi  $\lambda$ -uniforme sur  $\Theta$ ,  $\lambda_2(\cdot)$   etant la mesure de Lebesgue sur  $\mathbb{R}^2$ . Dans notre algorithme, nous avons choisi pour  $\Theta$  le disque de rayon 1.

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

1. Choix uniforme d'un objet  $x_i$  de la configuration, avec la probabilit e  $\frac{1}{n(\mathbf{x})}$ , que l'on va translater.
2. G en eration de la variable al eatoire  $\theta = (\theta_X, \theta_Y) \in \Theta$ .
3. Calcul de l'objet  $x_i$  translatt e, not e  $\theta(x_i) = (X_{x_i} + \theta_X, Y_{x_i} + \theta_Y, R_{x_i})$ . Si celui-ci n'appartient pas  a  $U$  (car sa position est hors de l'espace des positions  $\mathcal{P}$ ), le mouvement est aussit ot rejet e.

4. Calcul du rapport de Green  $R_t$ .
5. Acceptation ou non de la nouvelle configuration  $\mathbf{y} = \mathbf{x} \cup \{\theta(x_i)\} \setminus \{x_i\}$ .

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

$$Q_t(\mathbf{x}, A) = \sum_{x_i \in \mathbf{x}} \frac{1}{n(\mathbf{x})} \int_{\theta \in \Theta} \mathbf{1}_A(\mathbf{x} \cup \{\theta(x_i)\} \setminus \{x_i\}) \frac{\lambda_2(d\theta)}{\lambda_2(\Theta)}. \quad (3.68)$$

La mesure symétrique  $\xi_t(\cdot, \cdot)$  définie sur  $\Psi \times \Psi$ , par rapport à laquelle  $\pi(\cdot)Q(\cdot, \cdot)$  sera absolument continu, est :

$$\xi_t(A, B) = \int_{\Psi} \mathbf{1}_A(\mathbf{x}) \sum_{x_i \in \mathbf{x}} \int_{\theta \in \Theta} \mathbf{1}_B(\mathbf{x} \cup \{\theta(x_i)\} \setminus \{x_i\}) \lambda_2(d\theta) \mu(d\mathbf{x}). \quad (3.69)$$

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 :

$$f_t(\mathbf{x}, \mathbf{y}) = \frac{\pi(d\mathbf{x})Q_t(\mathbf{x}, d\mathbf{y})}{\xi_t(d\mathbf{x}, d\mathbf{y})} = \frac{h(\mathbf{x})\mu(d\mathbf{x})Q_t(\mathbf{x}, d\mathbf{y})}{\mu(d\mathbf{x})\lambda_2(d\theta)} = \frac{h(\mathbf{x})}{n(\mathbf{x})\lambda_2(\Theta)}. \quad (3.70)$$

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

$$\boxed{R_t(\mathbf{x}, \mathbf{x} \cup \{\theta(x_i)\} \setminus \{x_i\}) = \frac{h(\mathbf{x} \cup \{\theta(x_i)\} \setminus \{x_i\})}{h(\mathbf{x})}.} \quad (3.71)$$

*Dilatation*- 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 ce faire, nous introduisons une variable aléatoire  $\delta$  sur un compact symétrique de  $\mathbb{R}$ ,  $\Delta$  : ainsi, si  $\delta \in \Delta$ , alors  $-\delta \in \Delta$ . Cette variable aléatoire suit une loi  $\lambda$ -uniforme sur  $\Delta$ ,  $\lambda(\cdot)$  étant la mesure de Lebesgue sur  $\mathbb{R}$ . Dans notre algorithme,  $\Delta$  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  $\mathbf{y} = \mathbf{x} \cup \{\delta(x_i)\} \setminus \{x_i\}$ .

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  $\mu$ . On a :

$$Q_d(\mathbf{x}, A) = \sum_{x_i \in \mathbf{x}} \frac{1}{n(\mathbf{x})} \int_{\delta \in \Delta} \mathbf{1}_A(\mathbf{x} \cup \{\delta(x_i)\} \setminus \{x_i\}) \frac{\lambda(d\delta)}{\lambda(\Delta)}. \quad (3.72)$$

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

$$\xi_d(A, B) = \int_{\Psi} \mathbf{1}_A(\mathbf{x}) \sum_{x_i \in \mathbf{x}} \int_{\delta \in \Delta} \mathbf{1}_B(\mathbf{x} \cup \{\delta(x_i)\} \setminus \{x_i\}) \lambda(d\delta) \mu(d\mathbf{x}). \quad (3.73)$$

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 :

$$f_d(\mathbf{x}, \mathbf{y}) = \frac{\pi(d\mathbf{x})Q_d(\mathbf{x}, d\mathbf{y})}{\xi_d(d\mathbf{x}, d\mathbf{y})} = \frac{h(\mathbf{x})\mu(d\mathbf{x})Q_d(\mathbf{x}, d\mathbf{y})}{\mu(d\mathbf{x})\lambda(d\delta)} = \frac{h(\mathbf{x})}{n(\mathbf{x})\lambda(\Delta)}. \quad (3.74)$$

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

$$\boxed{R_d(\mathbf{x}, \mathbf{x} \cup \{\delta(x_i)\} \setminus \{x_i\}) = \frac{h(\mathbf{x} \cup \{\delta(x_i)\} \setminus \{x_i\})}{h(\mathbf{x})}}. \quad (3.75)$$

*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.

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  $\mu_d$ . On a :

$$Q_{td}(\mathbf{x}, A) = \sum_{x_i \in \mathbf{x}} \frac{1}{n(\mathbf{x})} \int_{\theta \in \Theta} \int_{\delta \in \Delta} \mathbf{1}_A(\mathbf{x} \cup \{\theta(\delta(x_i))\} \setminus \{x_i\}) \frac{\lambda(d\delta)}{\lambda(\Delta)} \frac{\lambda_2(d\theta)}{\lambda_2(\Theta)}. \quad (3.76)$$

Le rapport de Green associ e   cette transformation s'obtient de la m me fa on que les pr c dents :

$$\boxed{R_{td}(\mathbf{x}, \mathbf{x} \cup \{\theta(\delta(x_i))\} \setminus \{x_i\}) = \frac{h(\mathbf{x} \cup \{\theta(\delta(x_i))\} \setminus \{x_i\})}{h(\mathbf{x})}.} \quad (3.77)$$

*Mouvements complexes*- On appelle mouvement complexe tout mouvement aboutissant   un changement de dimension de la configuration  $\mathbf{x}$ .

Les mouvements de naissance et de mort uniformes ont d j   t  pr sent es dans une partie ant rieure. Ce ne sont pas les seuls   m riter d' tre propos es. Ainsi nous d finissons un mouvement de fusion-division.

On rappelle que la relation d'intersection  $\sim_I$  a  t  d finie comme suit :

$$x_i \sim_I x_j \Leftrightarrow d(C_{x_i}, C_{x_j}) < R_{x_i} + R_{x_j},$$

o   $d(\dots)$  est la distance euclidienne sur  $\mathbb{R}^2$ . Le nombre de paires d'objets (ou cliques) d'une configuration  $\mathbf{x}$ , voisins au sens de  $\sim_I$ , avait  t  not e  $n_{\sim_I}(\mathbf{x})$  :

$$n_{\sim_I}(\mathbf{x}) = \sum_{x_i, x_j \in \mathbf{x}, x_i \neq x_j} \mathbf{1}_{x_i \sim_I x_j}.$$

A partir de la configuration courante  $\mathbf{x}_t$ , une fusion consiste   supprimer deux objets  $x_i$  et  $x_j$  de  $\mathbf{x}_t$ , voisins au sens de la relation  $\sim_I$ , et de rajouter   la configuration un nouvel objet  $u \in U$  obtenu   partir des deux objets supprim es.

Une division consiste   supprimer un objet  $x_i$  de la configuration, et   rajouter deux objets  $u$  et  $v$ , voisins au sens de  $\sim_I$ , obtenus   partir de  $x_i$ .

Les deux mouvements de fusion et de division doivent  tre construits de fa on    tre sym triques afin de pr server la r versibilit  de la cha ne de Markov.

*Fusion :*

Les  tapes de la fusion de deux objets  $x_i$  et  $x_j$  d'une configuration sont les suivantes :

1. Choix d'une clique d'objets  $(x_i, x_j)$  dans la configuration, avec une probabilit  uniforme sur l'ensemble des cliques  $\frac{1}{n_{\sim_I}(\mathbf{x})}$ . Ces deux objets sont d crits par leurs coordonn es dans l'espace objet  $U$  :

$$x_i = (X_{x_i}, Y_{x_i}, R_{x_i}) \text{ et } x_j = (X_{x_j}, Y_{x_j}, R_{x_j}).$$

2. On calcule l'objet  $u = (X_u, Y_u, R_u)$ , r sultat de la fusion de  $x_i$  et  $x_j$ .

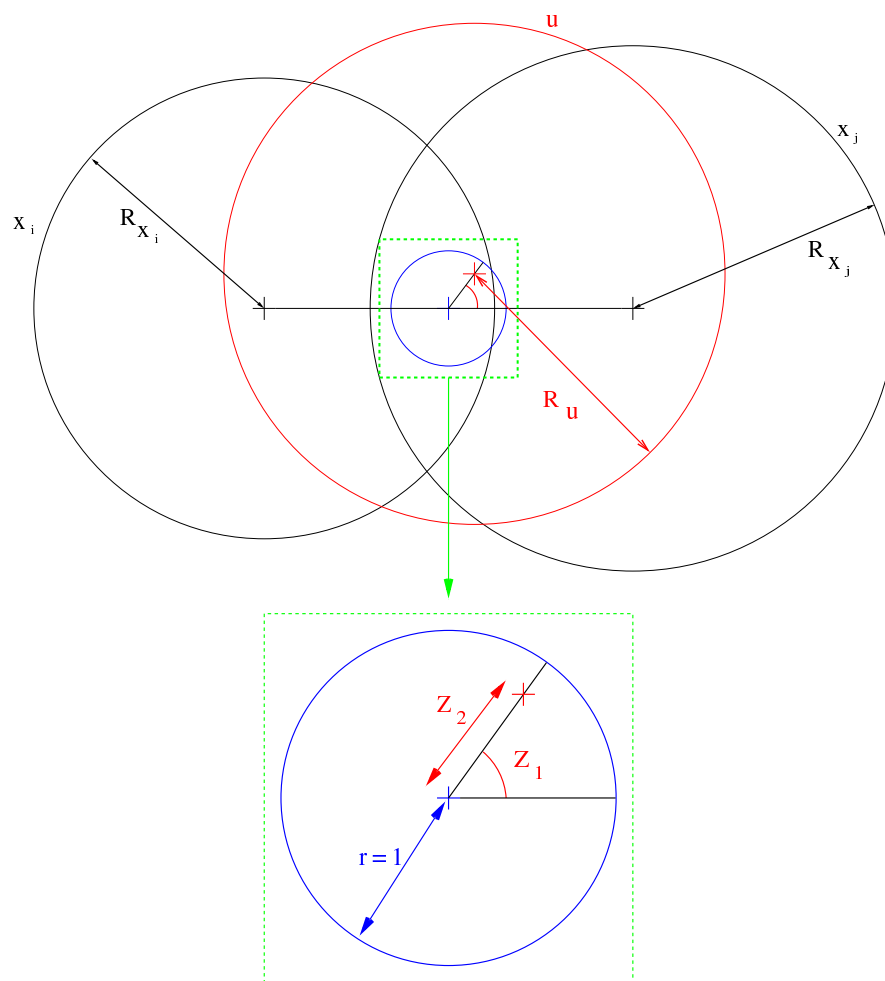


FIG. 3.23 – Schéma explicatif du mouvement de fusion.

Le centre de cet objet  $(X_u, Y_u) \in \mathcal{P}$  se situe dans un disque de centre le milieu des deux objets à fusionner,  $\left(\frac{X_{x_i} + X_{x_j}}{2}, \frac{Y_{x_i} + Y_{x_j}}{2}\right)$ , et de rayon 1. Ce centre est caractérisé par ses coordonnées polaires dans le disque. Pour les obtenir, on génère une variable aléatoire  $z_1$  uniformément sur  $[0, 2\pi[$ , ainsi qu'une variable aléatoire  $z_2 \in ]0, 1]$  (0 exclu pour assurer la différentiabilité de la transformation inverse) selon une loi normale tronquée de moyenne 0 et de variance  $\frac{1}{9}$  :

$$f_{z_2}(x) = \frac{\frac{3}{\sqrt{2\pi}} \exp\left(\frac{-9x^2}{2}\right)}{\int_0^1 \frac{3}{\sqrt{2\pi}} \exp\left(\frac{-9t^2}{2}\right) dt}$$

Nous choisissons une loi normale pour  $z_2$ , et non pas une simple loi uniforme sur tout le disque, car nous souhaitons donner plus de poids aux centres situés près du milieu des deux centres des objets  $x_i$  et  $x_j$ . Les expériences menées avec ces deux types d'approches donnent raison à celle choisie.

Si, suite au choix des variables  $z_1$  et  $z_2$ ,  $(X_u, Y_u) \notin \mathcal{P}$ , le mouvement de fusion est refusé. Ce cas de figure peut se présenter si les objets à fusionner sont près de la frontière de l'espace des positions.

Puis, on génère une troisième variable aléatoire  $z_3$  qui nous permet de calculer le rayon du nouvel objet  $R_u$ , en l'ajoutant à la moyenne des deux précédents  $R_{moy} = \frac{R_{x_i} + R_{x_j}}{2}$ . Comme nous souhaitons pouvoir augmenter ou diminuer cette valeur, nous permettons au nouveau cercle d'avoir un rayon situé dans la fourchette suivante :  $[R_{moy} - 1, R_{moy} + 1]$ . Ceci, compte tenu des restrictions liées aux bornes de l'espace des marques, fait que l'on choisit  $z_3$  uniformément dans  $[\max(-1, R_m - R_{moy}), \min(1, R_M - R_{moy})] = [m_{z_3}, M_{z_3}]$ .

Remarquons dès maintenant que  $M_{z_3} - m_{z_3} \geq \min(1, R_M - R_m)$ , ceci nous sera utile par la suite.

Les opérations d'écrites plus haut peuvent se résumer à l'aide des équations suivantes, avec la variable aléatoire  $z = (z_1, z_2, z_3) \in Z_f = [0, 2\pi[ \times ]0, 1] \times [m_{z_3}, M_{z_3}]$  :

$$\begin{pmatrix} X_u & = & \frac{X_{x_i} + X_{x_j}}{2} + z_2 \cos(z_1) \\ Y_u & = & \frac{Y_{x_i} + Y_{x_j}}{2} + z_2 \sin(z_1) \\ R_u & = & \frac{R_{x_i} + R_{x_j}}{2} + z_3 \end{pmatrix}$$

La loi de probabilité de la variable aléatoire  $z$  est donc :

$$d\mathbf{P}_f(z) = f_f(z) \lambda(dz) = \frac{1}{2\pi} f_{z_2}(z_2) \frac{1}{M_{z_3} - m_{z_3}} \lambda(dz)$$

La fonction  $\eta_{x_i, x_j}$ , qui à un élément de  $Z_f$  associe l'objet résultat de la fusion, est un difféomorphisme :

$$\eta_{x_i, x_j} : \begin{pmatrix} Z_f & \rightarrow & U \\ z & \rightarrow & \eta_{x_i, x_j}(z) \end{pmatrix} \quad (3.78)$$

Le calcul de son Jacobien s'effectue ainsi :

$$|J_{\eta_{x_i, x_j}}(z)| = \left| \det \begin{pmatrix} -z_2 \sin(z_1) & z_2 \cos(z_1) & 0 \\ \cos(z_1) & \sin(z_1) & 0 \\ 0 & 0 & 1 \end{pmatrix} \right| = z_2 > 0$$

Ce calcul nous sera utile par la suite, lorsque nous souhaiterons exprimer le rapport de Green associée à cette transformation.

3. Calcul du rapport de Green  $R_f$ .
4. Avec la probabilité  $\alpha_f = \min(1, R_f)$ , on accepte la nouvelle configuration  $\mathbf{y} = \mathbf{x} \setminus \{x_i \cup x_j\} \cup \{u\}$ . Avec la probabilité  $1 - \alpha_f$ , on reste dans la configuration courante :  $\mathbf{y} = \mathbf{x}$ .

### Division

Les étapes de la division d'un objet  $x$  d'une configuration sont les suivantes :

1. Choix d'un objet  $x_i$  dans la configuration, avec une probabilité uniforme sur la configuration  $\frac{1}{n(\mathbf{x})}$ . Cet objet est décrit par ses coordonnées sur  $U$  :

$$x_i = (X_{x_i}, Y_{x_i}, R_{x_i})$$

2. On calcule les objets  $u_1 = (X_{u_1}, Y_{u_1}, R_{u_1})$  et  $u_2 = (X_{u_2}, Y_{u_2}, R_{u_2})$ , résultats de la division de  $x$ .

Pour que ce mouvement soit symétrique avec le mouvement de fusion, le milieu des centres de ces objets  $\left(\frac{X_{u_1} + X_{u_2}}{2}, \frac{Y_{u_1} + Y_{u_2}}{2}\right) \in \mathcal{P}$  doit se situer dans un disque de centre  $(X_{x_i}, Y_{x_i})$ , et de rayon 1. Pour l'obtenir, on raisonne toujours en coordonnées polaires, et on génère une variable aléatoire  $z$  uniformément sur  $[0, 2\pi[$ , ainsi qu'une variable aléatoire  $z \in ]0, 1]$  selon la loi  $f_{z_2}$  explicitée plus haut.

Si, suite au choix des variables  $z_1$  et  $z_2$ , le milieu des centres des objets  $u_1$  et  $u_2$  n'appartient pas à l'espace des positions  $\mathcal{P}$ , le mouvement de division est refusé.

Puis, on génère uniformément sur  $[\max(-1, R - R_{x_i}), \min(1, R_M - R_{x_i})] = [m_{z_3}, M_{z_3}]$  une variable aléatoire  $z_3$ , qui nous permet de calculer la moyenne des rayons des objets créés  $\frac{R_{u_1} + R_{u_2}}{2}$ , en fonction du rayon  $R_{x_i}$ .

A ce stade des opérations, nous connaissons le milieu des centres et la moyenne des rayons des objets  $u_1$  et  $u_2$  :

$$\begin{pmatrix} \frac{X_{u_1} + X_{u_2}}{2} & = & X_{x_i} + z_2 \cos(z_1) \\ \frac{Y_{u_1} + Y_{u_2}}{2} & = & Y_{x_i} + z_2 \sin(z_1) \\ \frac{R_{u_1} + R_{u_2}}{2} & = & R_{x_i} + z_3 \end{pmatrix}$$

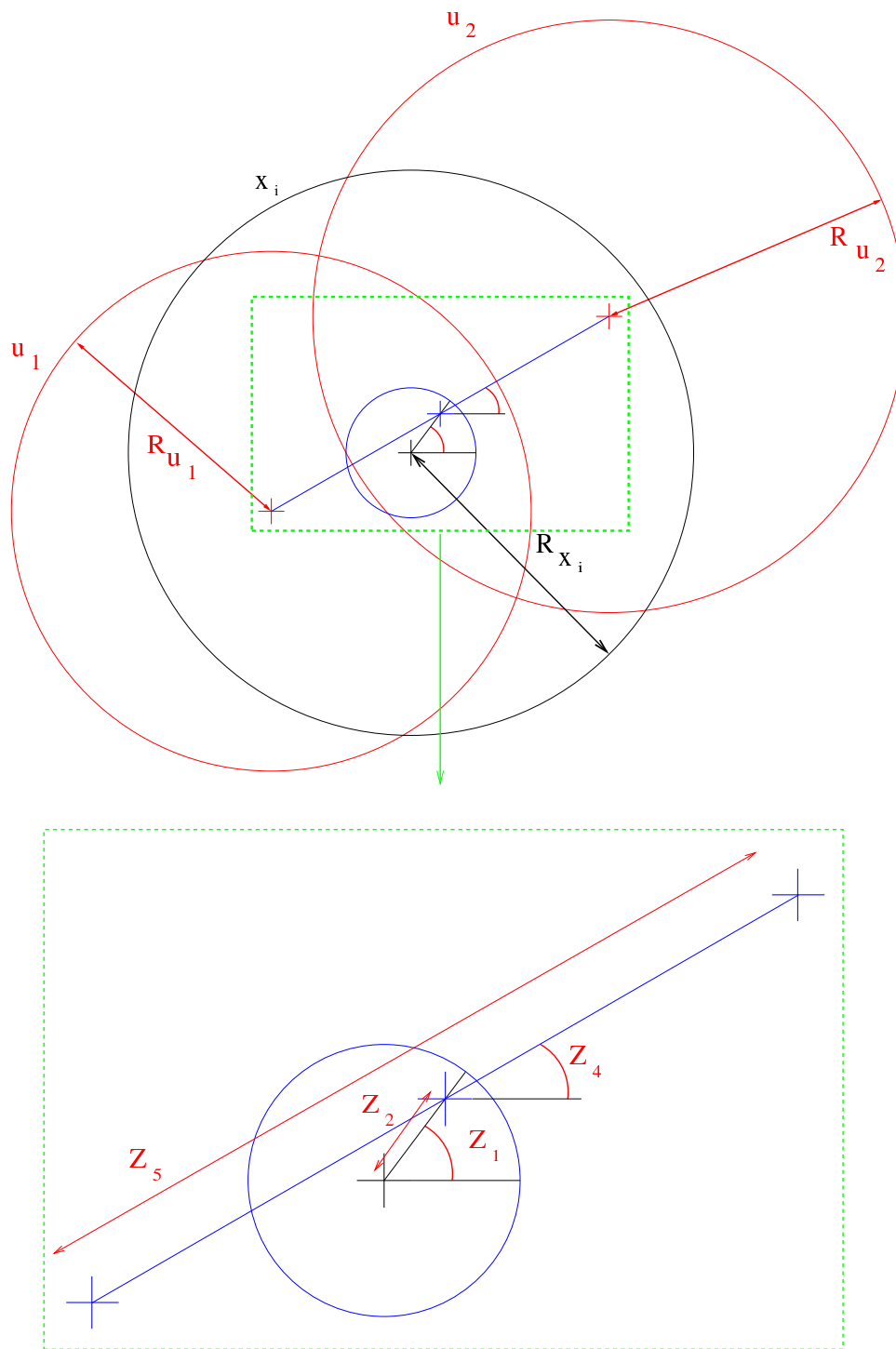


FIG. 3.24 – Schéma explicatif du mouvement de division.

Pour un même milieu des centres et une même moyenne des rayons, il existe une infinité de divisions possibles. Il nous faut introduire trois nouvelles variables aléatoires pour que cette transformation soit bijective.

Remarquons qu'en terme de positionnement dans l'espace  $\mathcal{P}$ , il nous suffit de placer l'un des deux objets puisque l'autre sera son symétrique par rapport au milieu des centres explicité plus haut. Il en est de même pour les rayons, puisque la connaissance d'un des deux rayons nous suffit. C'est pourquoi nous nous attacherons par la suite à calculer les coordonnées de  $u_1$ , à partir desquelles nous déduirons celles de  $u_2$ .

Plaçons nous alors dans un repère d'origine le milieu des centres, et raisonnons une nouvelle fois en terme de coordonnées polaires. Une variable aléatoire tirée uniformément sur  $[0, \pi[$  nous donne l'orientation de  $u_1$ . Restent à fixer la distance entre les 2 objets, ainsi que l'écart entre leurs rayons.

Pour cela, remarquons que dans le but que  $u_1 \sim_I u_2$ , il faut que :

$$d(u_1, u_2) < R_{u_1} + R_{u_2} = 2(R_{x_i} + z_3).$$

Ainsi, on tire une variable aléatoire  $z_5$  uniformément sur  $]0, 2(R_{x_i} + z_3)[$ , afin de fixer la distance en question :  $d(u_1, u_2) = z_5$ . Celle-ci doit être non nulle de manière à respecter notre choix de simuler un processus ponctuel marqué simple, processus ponctuel simple sur l'espace des positions, ne pouvant comporter deux objets avec la même position.

Une dernière variable  $z_6$  nous donne la différence entre les rayons  $R_{u_1} - R_{u_2}$ . Celle-ci est choisie uniformément sur  $[-M_{z_6}, M_{z_6}]$ , où  $M_{z_6} = \min(R_M - (R_{x_i} + z_3), (R_{x_i} + z_3) - R_m)$ , afin que l'on ait  $R_m \leq \frac{R_{u_1}}{R_{u_2}} \leq R_M$ .

Les opérations décrites plus haut peuvent se résumer à l'aide des équations suivantes, avec la variable aléatoire  $z = (z_1, z_2, z_3, z_4, z_5, z_6) \in Z_d = [0, 2\pi[ \times ]0, 1] \times [-1, 1] \times [0, \pi[ \times [0, 2(R_{x_i} + z_3)[ \times [-M_{z_6}, M_{z_6}]$  :

$$\begin{pmatrix} X_{u_1} & = & X_{x_i} + z_2 \cos(z_1) + \frac{z_5 \cos(z_4)}{2} \\ X_{u_2} & = & X_{x_i} + z_2 \cos(z_1) - \frac{z_5 \cos(z_4)}{2} \\ Y_{u_1} & = & Y_{x_i} + z_2 \sin(z_1) + \frac{z_5 \sin(z_4)}{2} \\ Y_{u_2} & = & Y_{x_i} + z_2 \sin(z_1) - \frac{z_5 \sin(z_4)}{2} \\ R_{u_1} & = & R_{x_i} + z_3 + \frac{z_6}{2} \\ R_{u_2} & = & R_{x_i} + z_3 - \frac{z_6}{2} \end{pmatrix}$$

La loi de probabilité de la variable aléatoire  $z$  est donc :

$$d\mathbf{P}_d(z) = f_d(z) \lambda(dz) = \left(\frac{1}{2\pi}\right)^2 f_{z_2}(z_2) \frac{1}{M_{z_3} - m_{z_3}} \frac{1}{2(R_{x_i} + z_3)} \frac{1}{M_{z_6}} \lambda(dz)$$

La fonction  $\zeta_{x_i}$ , qui à un élément de  $Z$  associe les objets résultats de la division, est un difféomorphisme :

$$\zeta_{x_i} : \begin{pmatrix} Z_d & \rightarrow & U \times U \\ z & \rightarrow & \zeta_{x_i}(z) \end{pmatrix} \quad (3.79)$$

Le calcul de son Jacobien s'effectue ainsi :

$$|J_{\zeta_{x_i}}(z)| = \det \begin{pmatrix} -z_2 \sin(z_1) & -z_2 \sin(z_1) & z_2 \cos(z_1) & z_2 \cos(z_1) & 0 & 0 \\ \cos(z_1) & \cos(z_1) & \sin(z_1) & \sin(z_1) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ -\frac{z_5 \sin(z_4)}{2} & \frac{z_5 \sin(z_4)}{2} & \frac{z_5 \cos(z_4)}{2} & -\frac{z_5 \cos(z_4)}{2} & 0 & 0 \\ \frac{\cos(z_5)}{2} & -\frac{\cos(z_5)}{2} & \frac{\sin(z_5)}{2} & -\frac{\sin(z_5)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{pmatrix} = z_2 z_5 > 0$$

Ce calcul nous sera utile par la suite, lorsque nous souhaiterons exprimer le rapport de Green associée à cette transformation.

3. Calcul du rapport de Green  $R_d$ .

4. Avec la probabilité  $q = \min(1, R_d)$ , on accepte la nouvelle configuration  $\mathbf{y} = \mathbf{x} \setminus \{x_i\} \cup \{u_1 \cup u_2\}$ . Avec la probabilité  $1 - q$ , on reste dans la configuration courante :  $\mathbf{y} = \mathbf{x}$ .

### Bijection

Il est à noter qu'il existe une bijection associant les deux mouvements de fusion et de division à travers les objets et les variables aléatoires mises en jeu. En effet, à un vecteur de  $U \times U \times \mathcal{Z}$ , on associe un et un seul vecteur sur  $U \times Z_d$ , et réciproquement :

$$\left( \begin{array}{l} U \times U \times Z_f \\ (X_{x_i}, Y_{x_i}, R_{x_i}, X_{x_j}, Y_{x_j}, R_{x_j}, z_1, z_2, z_3) \end{array} \leftrightarrow \begin{array}{l} U \times Z_d \\ (X_u, Y_u, R_u, z'_1, z'_2, z'_3, z'_4, z'_5, z'_6) \end{array} \right) \quad (3.80)$$

Ceci montre bien la symétrie des transformations proposées. On remarque alors que  $z_2 = z'_2$ , ce qui fait que, par la suite, nous noterons simplement  $z_2$  dans les deux cas. On rappelle aussi que par construction,  $z'_5 = d(x_i, x_j)$ .

### Noyaux de propositions du fusion et de division

Soit  $Q_{fd}(\cdot, \cdot)$  le noyau de propositions du mouvement de fusion/division. Il s'écrit :

$$Q_{fd}(\mathbf{x}, \cdot) = p_f Q_f(\mathbf{x}, \cdot) + p_d Q_d(\mathbf{x}, \cdot).$$

Explicitons pour chacun des mouvements le noyau propre qui lui est associée :

Pour la fusion :

$$\begin{aligned} Q_f(\mathbf{x}, A) &= \frac{1}{n_{\sim_I}(\mathbf{x})} \sum_{x_i \sim_I x_j} \int_{Z_f} \mathbf{1}_A(\mathbf{x} \setminus \{x_i \cup x_j\} \cup \{\eta_{x_i, x_j}(z)\}) \overbrace{f_f(z) \lambda(dz)}^{d\mathbf{P}_f(z)} \\ &= \frac{1}{n_{\sim_I}(\mathbf{x})} \sum_{x_i \sim_I x_j} \int_{\eta_{x_i, x_j}(Z_f) \subset U} \mathbf{1}_A(\mathbf{x} \setminus \{x_i \cup x_j\} \cup \{u\}) f_f(\eta_{x_i, x_j}^{-1}(u)) |J_{\eta_{x_i, x_j}^{-1}}(u)| \lambda(du) \\ &= \frac{1}{n_{\sim_I}(\mathbf{x})} \int_U \sum_{x_i \sim_I x_j} \mathbf{1}_A(\mathbf{x} \setminus \{x_i \cup x_j\} \cup \{u\}) \mathbf{1}_{u \in \eta_{x_i, x_j}(Z_f)} f_f(\eta_{x_i, x_j}^{-1}(u)) |J_{\eta_{x_i, x_j}^{-1}}(u)| \lambda(du) \end{aligned}$$



Pour la division :

$$\begin{aligned}
Q_d(\mathbf{x}, A) &= \frac{1}{n(\mathbf{x})} \sum_{x_i \in \mathbf{x}} \int_{Z_d} \mathbf{1}_A(\mathbf{x} \setminus \{x_i\} \cup \{\zeta_{x_i}(z)\}) \overbrace{f_d(z) \lambda(dz)}^{d\mathbf{P}_d(z)} \\
&= \frac{1}{n(\mathbf{x})} \sum_{x_i \in \mathbf{x}} \int_{\zeta_{x_i}(Z_d) \subset U \times U} \mathbf{1}_A(\mathbf{x} \setminus \{x_i\} \cup \{u_1 \cup u_2\}) f_d(\zeta_{x_i}^{-1}(u_1, u_2)) |J_{\zeta_{x_i}^{-1}}(u_1, u_2)| \lambda(du_1) \lambda(du_2) \\
&= \frac{1}{n(\mathbf{x})} \int_{U^2} \sum_{x_i \in \mathbf{x}} \mathbf{1}_A(\mathbf{x} \setminus \{x_i\} \cup \{u_1 \cup u_2\}) \mathbf{1}_{(u_1, u_2) \in \zeta_{x_i}(Z_d)} f_d(\zeta_{x_i}^{-1}(u_1, u_2)) |J_{\zeta_{x_i}^{-1}}(u_1, u_2)| \lambda(du_1) \lambda(du_2)
\end{aligned}$$

Mesure symétrique  $\xi_{fd}$

Pour le mouvement de fusion/division, on définit comme suit la mesure symétrique  $\xi_d(\cdot, \cdot)$  sur  $\Psi \times \Psi$  :

$$\begin{aligned}
\xi_{fd}(A, B) &= \int_{\Psi} \int_{u \in U} \mathbf{1}_A(\mathbf{x}) \sum_{(x_i, x_j) \in \mathbf{x}} \mathbf{1}_B(\mathbf{x} \setminus \{x_i \cup x_j\} \cup \{u\}) \lambda(du) \mu(d\mathbf{x}) \\
&+ \int_{\Psi} \int_{(u_1, u_2) \in U} \mathbf{1}_A(\mathbf{x}) \sum_{u \in \mathbf{x}} \mathbf{1}_B(\mathbf{x} \setminus \{u\} \cup \{u_1 \cup u_2\}) \frac{\lambda(du_1) \lambda(du_2)}{2} \mu(d\mathbf{x}).
\end{aligned}$$

Le  $\frac{1}{2}$  apparaît dans la deuxième partie de l'expression puisque, du point de vue de l'espace des configurations  $\Psi$ , étant donné qu'il est non ordonné, apporter à une configuration  $\mathbf{x}$  deux objets  $\{u_1, u_2\}$  est équivalent à lui apporter les deux objets  $\{u_2 \cup u_1\}$ . Il convient de ne compter qu'une fois ce couple pour obtenir la symétrie de la mesure  $\xi_d(\cdot, \cdot)$ .

Pour montrer que cette mesure est symétrique, il nous suffit alors de vérifier l'égalité suivante :

$$\xi_{fd}(A_{n+1}, B_n) = \xi_{fd}(B_n, A_{n+1}),$$

où l'on rappelle que  $A_n = A \cap \Psi_n$ .

Or, d'après la définition de la mesure  $\mu(\cdot)$  de référence :

$$\begin{aligned}
\xi_{fd}(A_{n+1}, B_n) &= \int_{\Psi_{n+1}} \int_{u \in U} \mathbf{1}_{A_{n+1}}(\mathbf{x}) \sum_{(x_i, x_j) \in \mathbf{x}} \mathbf{1}_{B_n}(\mathbf{x} \setminus \{x_i \cup x_j\} \cup \{u\}) \lambda(du) \mu(d\mathbf{x}) \\
&= \frac{e^{-\lambda(U)}}{(n+1)!} \int_{U^{n+1}} \int_{u \in U} \mathbf{1}_{A_{n+1}}(\mathbf{x}) \sum_{(x_i, x_j) \in \mathbf{x}} \mathbf{1}_{B_n}(\mathbf{x} \setminus \{x_i \cup x_j\} \cup \{u\}) \underbrace{\lambda(du) \lambda^{n+1}(d\mathbf{x})}_{\lambda^{n+2}(\cdot)} \\
&= \frac{e^{-\lambda(U)}}{(n+1)!} \int_{U^{n+1}} \int_{u \in U} C_{n+1}^2 \mathbf{1}_{A_{n+1}}(x_1, \dots, x_{n+1}) \mathbf{1}_{B_n}(\mathbf{x} \setminus \{x_n \cup x_{n+1}\} \cup \{u\}) \lambda^{n+2}(\cdot) \\
&= \frac{e^{-\lambda(U)}}{n!} \int_{U^n} \int_{(x_n, x_{n+1}) \in U} \frac{n}{2} \mathbf{1}_{B_n}(\mathbf{y} = \{x_1, \dots, x_{n-1}, u\}) \mathbf{1}_{A_{n+1}}(\mathbf{y} \setminus \{u\} \cup \{x_n \cup x_{n+1}\}) \lambda^{n+2}(\cdot) \\
&= \frac{e^{-\lambda(U)}}{n!} \int_{U^n} \int_{(x_n, x_{n+1}) \in U} \mathbf{1}_{B_n}(\mathbf{y}) \frac{1}{2} \sum_{u \in \mathbf{y}} \mathbf{1}_{A_{n+1}}(\mathbf{y} \setminus \{u\} \cup \{x_n \cup x_{n+1}\}) \lambda^{n+2}(\cdot) \\
&= \int_{\Psi_n} \int_{(x_n, x_{n+1}) \in U} \mathbf{1}_{B_n}(\mathbf{y}) \sum_{u \in \mathbf{y}} \mathbf{1}_{A_{n+1}}(\mathbf{y} \setminus \{u\} \cup \{x_n \cup x_{n+1}\}) \frac{\lambda(dx_n) \lambda(dx_{n+1})}{2} \mu(d\mathbf{y}) \\
&= \xi_{fd}(B_n, A_{n+1}).
\end{aligned}$$

Cela montre bien que la mesure  $\xi_{fd}(\cdot, \cdot)$  ainsi définie sur  $\Psi \times \Psi$  est symétrique.

#### Dérivées de Radon Nikodym

Avant d'accéder à l'expression des rapports de Green, nous devons calculer les dérivées de Radon Nikodym associées à chacun des mouvements.

Pour la fusion :

Afin d'alléger les notations, nous noterons :

$$\mathbf{y} = \mathbf{x} \setminus \{x_i \cup x_j\} \cup \{u\}.$$

Comme requis dans l'algorithme de Green,  $\pi(\cdot)Q_{fd}$  est absolument continu par rapport à la mesure symétrique  $\xi_d(\cdot, \cdot)$  puisque :

$$\pi(d\mathbf{x})Q_{fd}(\mathbf{x}, d\mathbf{y}) = h(\mathbf{x})\mu(d\mathbf{x}) \frac{pf}{n_{\sim_I}(\mathbf{x})} f_f(\eta_{x_i, x_j}^{-1}(u)) |J_{\eta_{x_i, x_j}^{-1}}(u)| \lambda(du)$$

et

$$\xi_{fd}(d\mathbf{x}, d\mathbf{y}) = \lambda(du)\mu(d\mathbf{x})$$

La dérivée de Radon Nikodym associée à ce mouvement est donc :

$$f(\mathbf{x}, \mathbf{y}) = pf \frac{h(\mathbf{x})}{n_{\sim_I}(\mathbf{x})} \frac{f_f(\eta_{x_i, x_j}^{-1}(u))}{z_2}$$

Pour la division :

Afin d'alléger les notations, nous noterons dans la suite de cette sous-partie :

$$\mathbf{y} = \mathbf{x} \setminus \{x_i\} \cup \{u_1 \cup u_2\}$$

Comme requis dans l'algorithme de Green,  $\pi(\cdot)Q_{fd}$  est absolument continu par rapport à la mesure symétrique  $\xi_d(\cdot, \cdot)$  puisque :

$$\pi(d\mathbf{x})Q_{fd}(\mathbf{x}, d\mathbf{y}) = h(\mathbf{x})\mu(d\mathbf{x})\frac{p_d}{n(\mathbf{x})}f_d(\zeta_{x_i}^{-1}(u_1, u_2))|J_{\zeta_{x_i}^{-1}}(u_1, u_2)|\lambda(du_1)\lambda(du_2)$$

et

$$\xi_{fd}(d\mathbf{x}, d\mathbf{y}) = \lambda(du_1)\lambda(du_2)\mu(d\mathbf{x})$$

La dérivée de Radon Nikodym associée à ce mouvement est donc :

$$f(\mathbf{x}, \mathbf{y}) = p_d \frac{h(\mathbf{x}) f_d(\zeta_{x_i}^{-1}(u_1, u_2))}{n(\mathbf{x}) z_2 d(u_1, u_2)}$$

*Rapports de Green :*

Nous avons désormais accès aux rapports de Green, et donc aux probabilités d'acceptation des mouvements de fusion et de division.

Pour la fusion :

$$R_f(\mathbf{x}, \mathbf{x} \setminus \{x_i \cup x_j\} \cup \{u\}) = \frac{f(\mathbf{y}, \mathbf{x})}{f(\mathbf{x}, \mathbf{y})} = \frac{p_d h(\mathbf{y})}{p_f h(\mathbf{x})} \frac{n_{\sim_I}(\mathbf{x})}{n(\mathbf{x}) - 1} \frac{f_d(\zeta_u^{-1}(x_i, x_j))}{f_f(\eta_{x_i, x_j}^{-1}(u))} \frac{1}{d(x_i, x_j)} \quad (3.81)$$

Pour la division :

$$R_d(\mathbf{x}, \mathbf{x} \setminus \{x_i\} \cup \{u_1 \cup u_2\}) = \frac{f(\mathbf{y}, \mathbf{x})}{f(\mathbf{x}, \mathbf{y})} = \frac{p_f h(\mathbf{y})}{p_d h(\mathbf{x})} \frac{n(\mathbf{x})}{n_{\sim_I}(\mathbf{y})} \frac{f_f(\eta_{u_1, u_2}^{-1}(x_i))}{f_d(\zeta_{x_i}^{-1}(u_1, u_2))} d(x_i, x_j) \quad (3.82)$$

**Résultats-** Les premiers résultats sont très encourageants comme le montre la figure 3.25. Il nous reste maintenant à généraliser le modèle pour aborder des cas plus difficiles (espèces d'arbres différentes, angles de prise de vue et reliefs différents). Pour cela, un objet géométrique plus complet, par exemple elliptique, devra être intégré. En outre, la prise en compte d'informations supplémentaires dans le modèle ou le noyau de proposition, comme par exemple concernant la régularité des espacements dans le cadre d'une plantation d'origine antropique, pourra s'avérer nécessaire.

### 3.4.2 Extraction des bâtiments

<sup>8</sup> L'extraction de bâtiments à partir de photographies aériennes et leur reconstruction automatique ont de nombreuses applications, comme par exemple en cartographie, en simulation de vols, préparation de missions militaires,... Néanmoins, la grande densité des espaces urbains et la complexité des objets construits par l'homme rendent cette tâche difficile et la cartographie urbaine 3D est toujours un problème ouvert. Les différentes approches rencontrées dans la littérature sont résumées dans [FKL, May99]. Nous pouvons distinguer deux grands axes de recherche suivant que l'on cherche à reconstruire une description détaillée de quelques bâtiments [FKL98], à partir de données précises sur de petites zones (en utilisant par exemple des données de type cadastral), ou au contraire une description moins précise (mais à plus grande échelle) de la scène urbaine, en utilisant moins de données [FRM99, SC00].

Nous nous intéressons ici à l'extraction de caricatures de bâtiments, correspondant au second problème. Notre but n'est donc pas d'obtenir un modèle 3D précis d'un bâtiment (modèle de toit par exemple) mais

8. Nous résumons ici le travail de DEA et de thèse de M. Ortner, boursier DGA. Bibliographie : [59, 60]

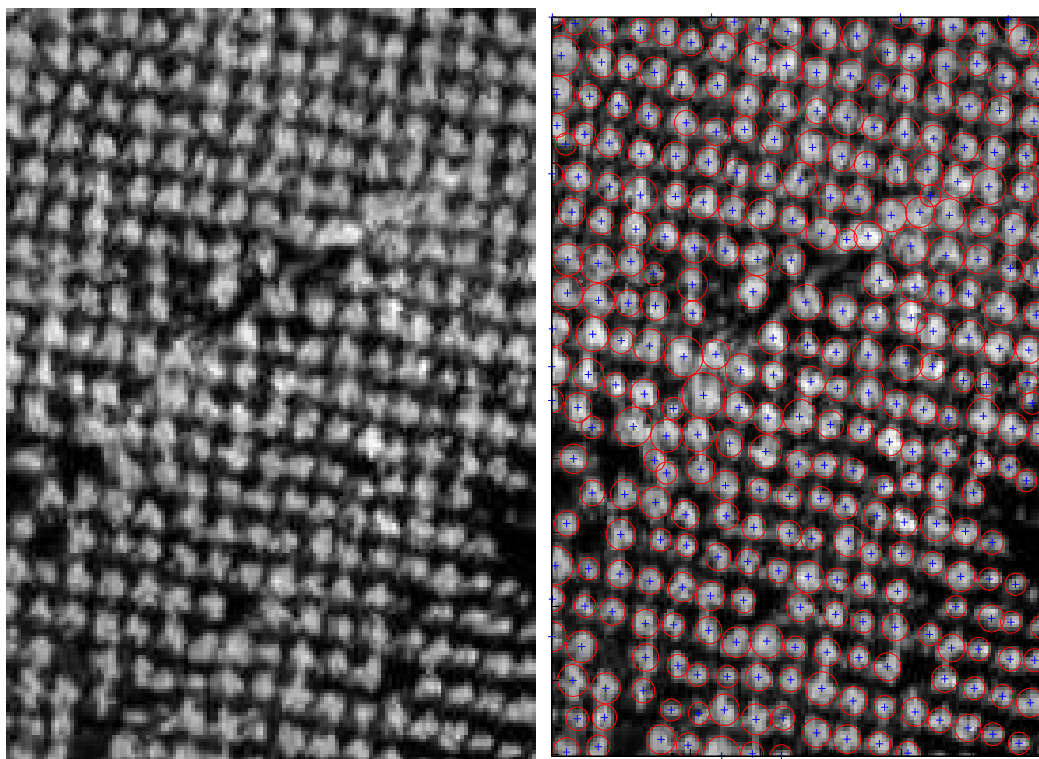
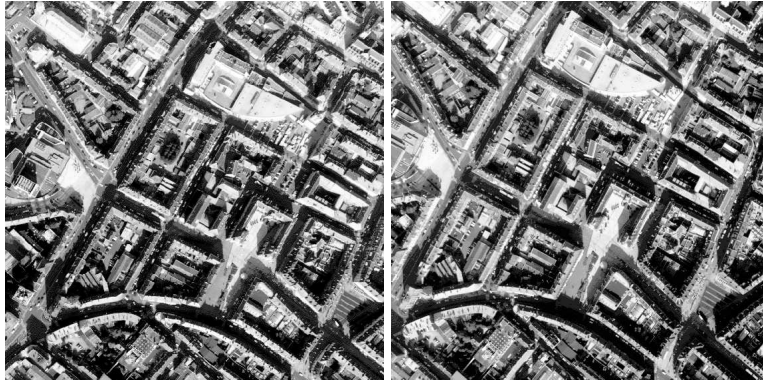


FIG. 3.25 – *Peupleraie (à gauche) et détection obtenue (à droite)*



a) Paire d'images stéréo de la ville d'Amiens.



b) Modèle Numérique d'Élévation d'Amiens (résolution de 20 cm).

FIG. 3.26 – Images aériennes et MNE fournis par l'IGN

plutôt de fournir un algorithme de détection le plus exhaustif possible, c'est-à-dire d'extraire essentiellement la trace au sol de tous les bâtiments de la scène. Nous obtenons ainsi une description vectorielle d'une scène urbaine dense. Une approche objet semble donc adaptée.

Nous considérons un modèle numérique d'élévation (MNE) fourni par l'IGN (figure 3.26 b) et obtenu par une technique de corrélation à partir d'un couple stéréo d'images aériennes de la ville d'Amiens (figure 3.26 a). La résolution moyenne du MNE est de 20cm au sol et de 10cm en hauteur. Nous pouvons distinguer un bruit résiduel, non pixélique, de l'algorithme de corrélation.

Nous considérons des objets définissant la trace au sol des bâtiments. Cela suffit à notre objectif d'extraction de caricatures et nous permet de travailler en 2D pour l'étape d'optimisation. Une fois la trace au sol détectée nous pouvons envisager l'estimation du toit avec des modèles possédant plusieurs paramètres.

**Un premier modèle-** Notre but étant plus l'extraction que la description des bâtiments, nous considérons un modèle simple, comme celui décrit sur la figure 3.27. Un tel modèle comprend déjà sept paramètres réels, ce qui donne un espace de recherche pour l'optimisation d'une taille considérable.

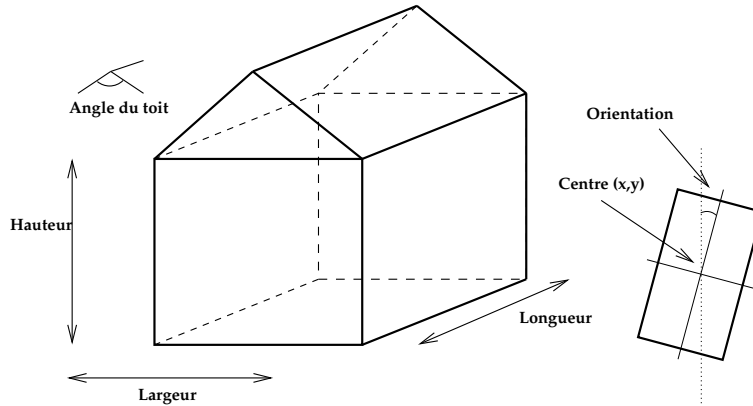


FIG. 3.27 – Exemple d'un modèle de bâtiment

Pour réduire cet espace, nous ne considérons que la trace au sol des bâtiments. Notre modèle consiste donc en :

- un modèle géométrique de la trace au sol du bâtiment,
- une fonction de coût mesurant la pertinence de la trace au sol d'un bâtiment relativement aux données,
- une fonction d'estimation du toit à partir de la trace au sol.

Nous modélisons la trace au sol par des rectangles, décrits par les éléments d'un espace à cinq dimensions données par le centre, la longueur, la largeur et l'orientation. Cet espace, noté  $M$ , est défini comme suit :

$$M = [0, X_{max}] \times [0, Y_{max}] \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \times [L_{min}, L_{max}] \times [l_{min}, l_{max}] \quad (3.83)$$

La fonction de coût est une application de l'espace  $S$  vers  $\mathbb{R}$ . Si l'on considère que le MNE est défini sur  $S = [0, X_{max}] \times [0, Y_{max}] \subset \mathbb{R}^2$  nous notons  $h$  l'application de  $S$  vers  $\mathbb{R}^+$  décrivant le MNE. Pour un point  $p$ ,  $h(p)$  est la hauteur en mètres donnée par le MNE.

Soit un ensemble  $s = \{x_1, \dots, x_n\}$  de point de  $S$ , nous notons  $\bar{s}$  la moyenne du MNE sur cet ensemble, c'est-à-dire :

$$\bar{s} = \frac{\sum_{i=1}^n h(x_i)}{n} \quad (3.84)$$

Soit un rectangle de  $M$ , nous construisons un ensemble de points associées à un rectangle, comme décrit sur la figure 3.28.

Ce masque de points est composé de cinq régions :

- quatre lignes autour du rectangle ( $g_1, \dots, g_4$ ) utilisées pour estimer la hauteur du sol  $\hat{h}_g$ ,
- l'aire centrale  $c$ , composée de  $N$  lignes colinéaires à la longueur  $c_1 \dots c_N$ .

Pour estimer le toit, nous considérons  $m$  couples de lignes  $\{l_1, \dots, l_m\}$  à partir de ces  $N$  lignes :

$$m = \left(\frac{N+1}{2}\right), \quad \forall i \in \{1, \dots, m\} \quad l_i = c_i \cup c_{N+1-i} \quad (3.85)$$

Ces couples sont donc constitués de lignes symétrique par rapport à l'axe de la longueur du rectangle.

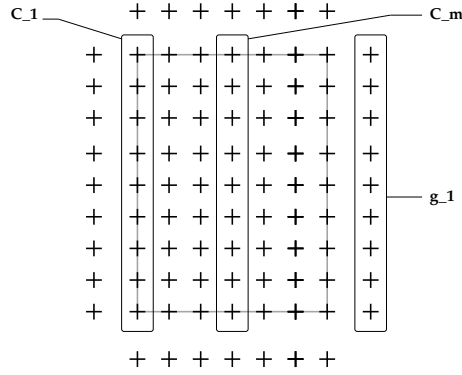


FIG. 3.28 – Points utilisés sur la trace au sol pour évaluer la fonction de coût et estimer le toit.

Pour estimer la hauteur au sol  $\widehat{h}_g$ , nous prenons la moyenne la plus basse de la hauteur des ensembles de points  $g_i$  :

$$\widehat{h}_g = \min_{i \in \{1, \dots, 4\}} \bar{g}_i \quad (3.86)$$

Pour l'aire centrale, nous définissons, en premier lieu, le taux de points de hauteur supérieure à un seuil  $v \in [0,1]$  :

$$v = \frac{\text{card} \{p \in c \text{ s. t. } (h(p) - \widehat{h}_g) \geq h_{min}\}}{\text{card } c} \quad (3.87)$$

où  $h_{min}$  est un paramètre du modèle fixant la hauteur minimale d'un bâtiment.

Les  $m$  couples de lignes nous donnent  $m$  hauteurs moyennes :

$$\forall i \in \{1, \dots, m\} \quad m_i = \bar{l}_i \quad (3.88)$$

Nous définissons un terme d'homogénéité  $t \in [0,1]$  par :

$$t = \frac{1}{m} \sum_{i=1}^m \frac{\text{card} \{p \in l_i \text{ s. t. } |h(p) - m_i| < \sigma\}}{\text{card } l_i} \quad (3.89)$$

où  $\sigma$  est un paramètre du modèle.

Pour finir, un terme de surface permet de favoriser les grands bâtiments : supposons que les longueur et largeur d'un rectangle  $(L, l)$  sont dans  $[L_{min}, L_{max}] \times [l_{min}, l_{max}]$ , nous considérons  $s \in [0,1]$  comme étant :

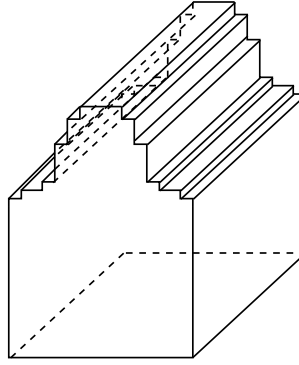
$$s = \frac{l * L}{l_{max} * L_{max}} \quad (3.90)$$

Pour finir, nous avons défini empiriquement la fonction de coût à partir de ces trois quantités. Pour un rectangle  $R$  et un MNE  $h$ , la fonction de coût  $J$  est donnée par :

$$J(R, h) = s * t^2 * v^3 \quad (3.91)$$

Cette fonction possède deux paramètres :  $h_{min}$  qui est un paramètre physique, et  $\sigma$  qui doit être réglé.

Avant de définir le terme d'attache aux données à partir de cette fonction de coût, considérons le modèle de toit. Les toits sont constitués des lignes colinéaires à l'axe de la longueur et de hauteur constante données par les  $m_0, \dots, m_m$  de la fonction de coût (cf. figure 3.29 pour un tel modèle). Cette

FIG. 3.29 – *Modèle de toit utilisé*

forme est symétrique par rapport à l'axe de la longueur du toit. L'énergie d'attache aux données est définie comme la somme de fonctions potentielles pour chaque bâtiment :

$$U_{data}(\mathbf{x}) = \sum_{u_i \in \mathbf{x}} V_d(u_i) \quad (3.92)$$

Le terme  $V_d$  doit rendre compte des points suivants :

1. les minima de  $V_d$  doivent être obtenus sur les maisons du MNE. Ceci est pris en compte par construction de la fonction de coût.
2.  $V_d$  doit être suffisamment lisse pour faciliter l'optimisation.

Nous proposons la fonction  $V_d$ , paramétrée par  $b_1, b_2$  et définie dans  $[-b_2, b_1]$ , suivante :

$$V_d(R, h) = \begin{cases} b_1 - b_1 \times \frac{v(R, h)^2}{v_{min}} & \text{si } v(R, h) \leq v_{min} \\ -b_2 * J(R, h) & \text{si } v(R, h) \geq v_{min} \end{cases} \quad (3.93)$$

pour un rectangle  $R \in S$  et une fonction de hauteur  $h(\cdot)$  données. Ce terme a été calibré empiriquement et s'avère robuste. Dans la pratique, nous avons utilisé les paramètres suivants :

$$\begin{aligned} b_2 &= 10 \\ b_1 &= 0.05 \\ v_{min} &= 0.8 \\ h_{min} &= 3 \text{ m} \\ \sigma &= 1.5 \text{ m} \end{aligned} \quad (3.94)$$

Pour un faible taux de points de hauteur supérieure à  $h_{min}$  (ie. taux inférieur à  $v_{min}$ ) le terme est répulsif. Dans le cas contraire le taux d'homogénéité et le terme de surface interviennent.

**Modèle a priori-** le terme a priori nous permet de gérer le recouvrement entre bâtiments. Nous considérons à cet effet un modèle de Strauss, défini comme suit :

$$U_{inter}(\mathbf{x}) = V_{inter} * s(\mathbf{x}) \quad (3.95)$$

où  $s(\mathbf{x})$  est le nombre de paires de rectangles de la configuration s'intersectant.  $V_{inter}$  est un paramètre réel :  $V_{inter} > 0$  et par conséquent le terme est répulsif et pénalise les intersections.

**Optimisation-** En dehors du noyau de naissance-mort, nous utilisons des transformations symétriques (ie.  $q_m(\mathbf{x}, \mathbf{y}) = q_m(\mathbf{y}, \mathbf{x})$ ). Pour ces transformations, nous choisissons uniformément un objet  $u$  dans la



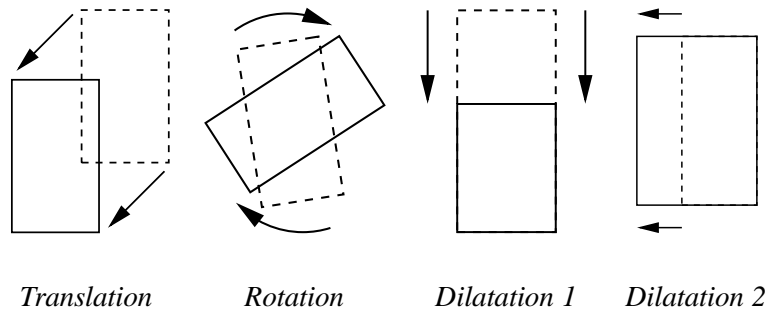


FIG. 3.30 – Transformations utilisées dans le noyau de propositions

configuration et posons  $y = x \setminus u \cup v$ , où  $v$  est g en er e suivant une loi d ependant de  $u$ . Ces transformations sont :

- La translation : Nous g en erons uniform ement  $(\delta_x, \delta_y)$  dans  $[-d_x, +d_x] \times [-d_y, +d_y]$  et, si  $u = (x, y, \theta, L, l)$  nous posons  $v = (x + \delta_x, y + \delta_y, \theta, L, l)$ , consid erant que  $K$  est un tore pour obtenir la sym etrie de la transformation.
- La rotation : Nous g en erons uniform ement dans  $[-d_\theta, +d_\theta]$  et si  $u = (x, y, \theta, L, l)$  nous posons  $v = (x, y, \theta + \delta_\theta, L, l)$ .
- La dilatation : Nous modifions   la fois la longueur ou la largeur et le centre du rectangle pour laisser invariant un des cot es du rectangle, comme montr e sur la figure 3.30.

L’optimisation est effectu ee par un recuit simul e avec une loi de d ecroissance de la temp erature g eom etrique.

**R esultats-** nous avons r egl e les param etres de fa on   accepter de faibles intersections. L’algorithme fourni une liste de b atiments dans un fichier texte, qui est ensuite lu par un algorithme de rendu 3D d evolopp e sous OpenGL. Les r esultats obtenus sur la figure 3.31 sugg erent quelques commentaires :

- des effets de bord sont pr esents : dans le coin sup erieur gauche nous avons une fausse alarme
- de petits b atiments de faible hauteur ne sont pas d etect es

La figure 3.32 (en haut) montre les r esultats de l’estimation des toits et une v erit e terrain (en bas) fournie par l’IGN. Quelques b atiments ne sont pas d etect es et les d etails de la forme de certains autres ne sont pas repr esent es du fait du mod ele rectangulaire de la trace au sol.

La figure 3.33 montre les donn ees et le r esultat en 3D, ce qui permet de voir que les erreurs sont dues aux artefacts sur les donn ees.

Ces r esultats ont  et e obtenus en 40 minutes sur une SUN-blade 2 (500 MHz, 250 MB), la taille de limage trait ee  etant de 1060 par 1024 pixels. En outre, nous avons d evolopp e un logiciel de visualisation 3D sous OpenGL<sup>9</sup> pour obtenir un meilleur rendu urbain comme le montre la figure 3.34.

**Un mod ele plus complexe de b atiment-** Pour pouvoir utiliser des caricatures de b atiments plus complexes, il faut augmenter la dimension de  $M$ . Nous proposons ici un exemple de mod ele de b atiment compos e d’un rectangle auquel on ajoute 6 param etres. La figure 3.35 pr esente ces param etres. Les deux premiers  $h_1, h_2$  correspondent respectivement   la hauteur de la goutti ere et   la hauteur du toit. Les quatres autres  $s_1, s_2, s_3, s_4$  correspondent   des param etres descriptifs de la forme du toit.  $S$  s’ ecrit donc :  $S = K \times M \times M'$ .

Puisque l’on dispose d’un mod ele plus pr ecis d’ecrivant les b atiments, on peut utiliser un terme d’attache aux donn ees mesurant la distance entre le MNE reconstruit  $h(x, \cdot)$ , et le MNE de donn ees  $h_{ef}(\cdot)$ . On se donne une grille de points de r ef erence de  $K$ , not ee  $T$ . On prend comme terme d’attache aux

9. Stage de DEA de F. Cerdat

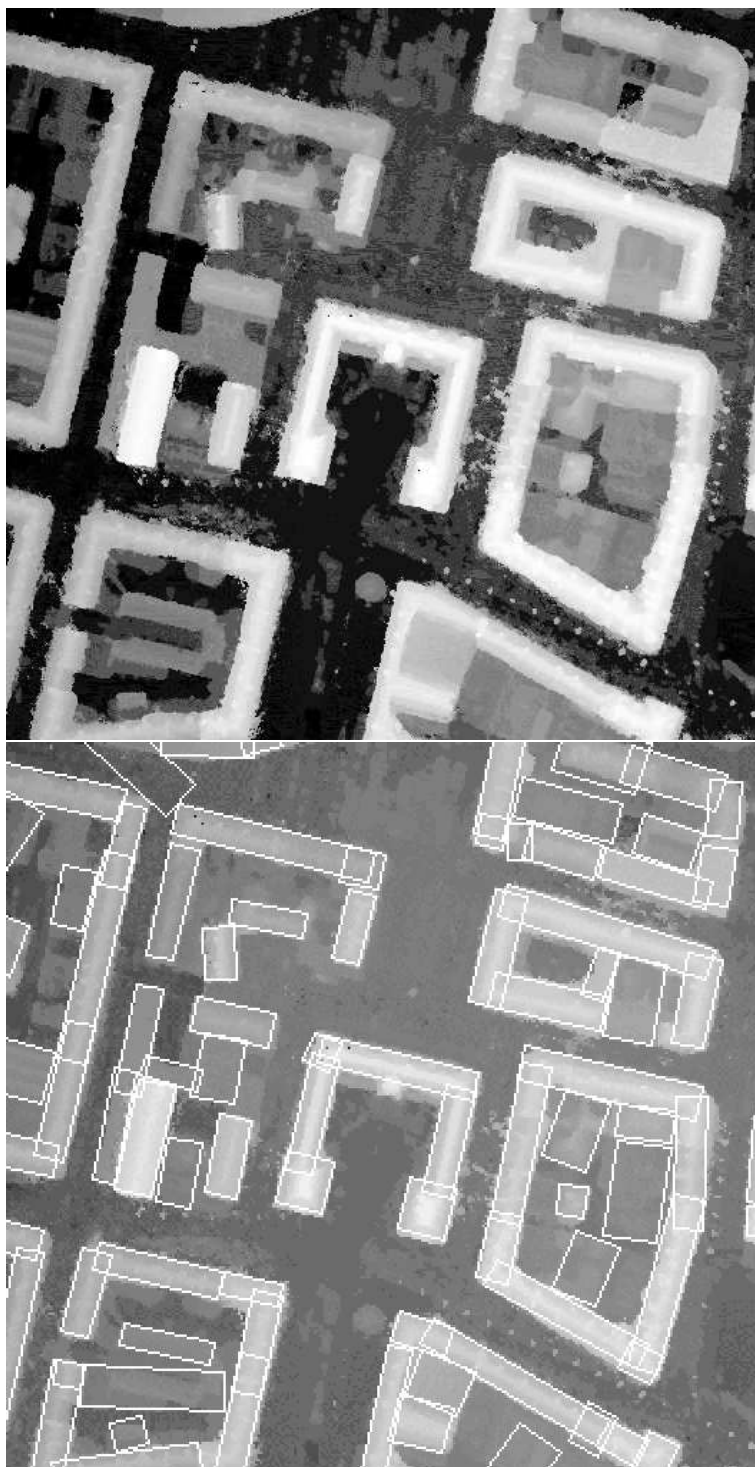


FIG. 3.31 – Résultat sur le MNE (en haut : MNE initial, en bas : résultat de l'extraction de bâtiments)

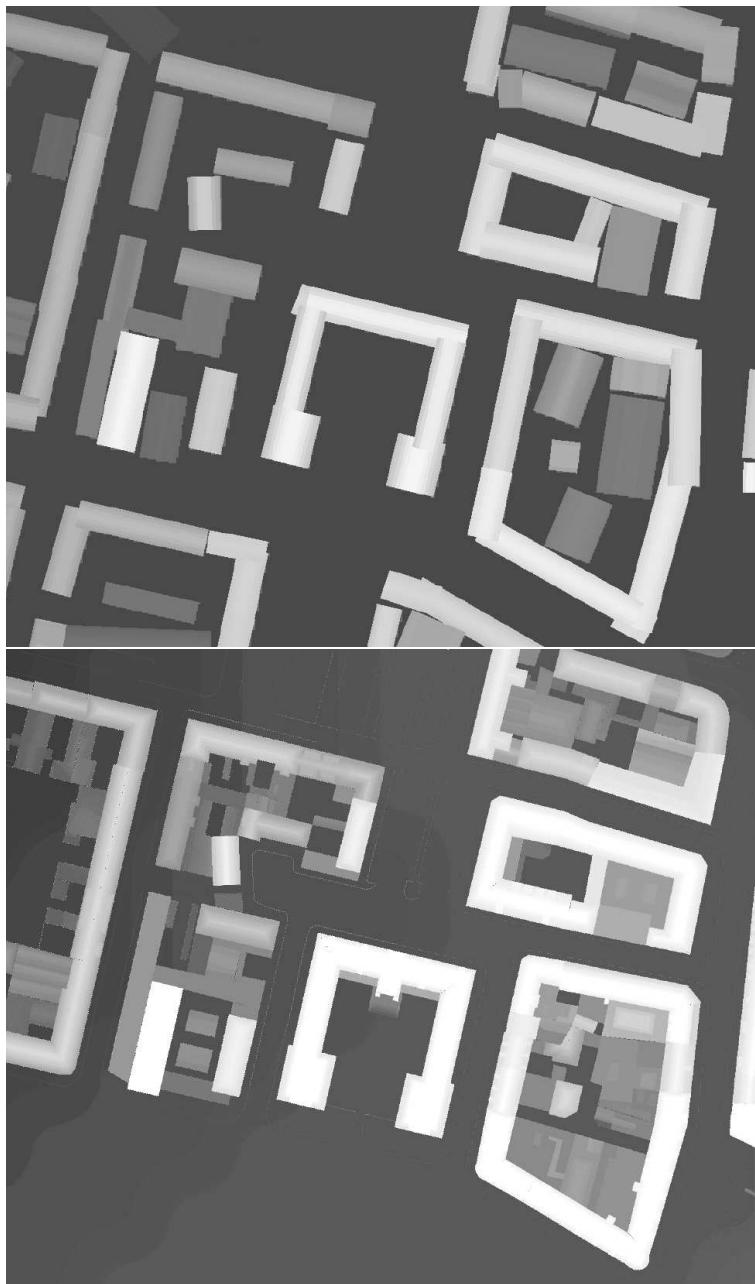


FIG. 3.32 – Validation (en haut : toits estimés, en bas : vérité terrain)

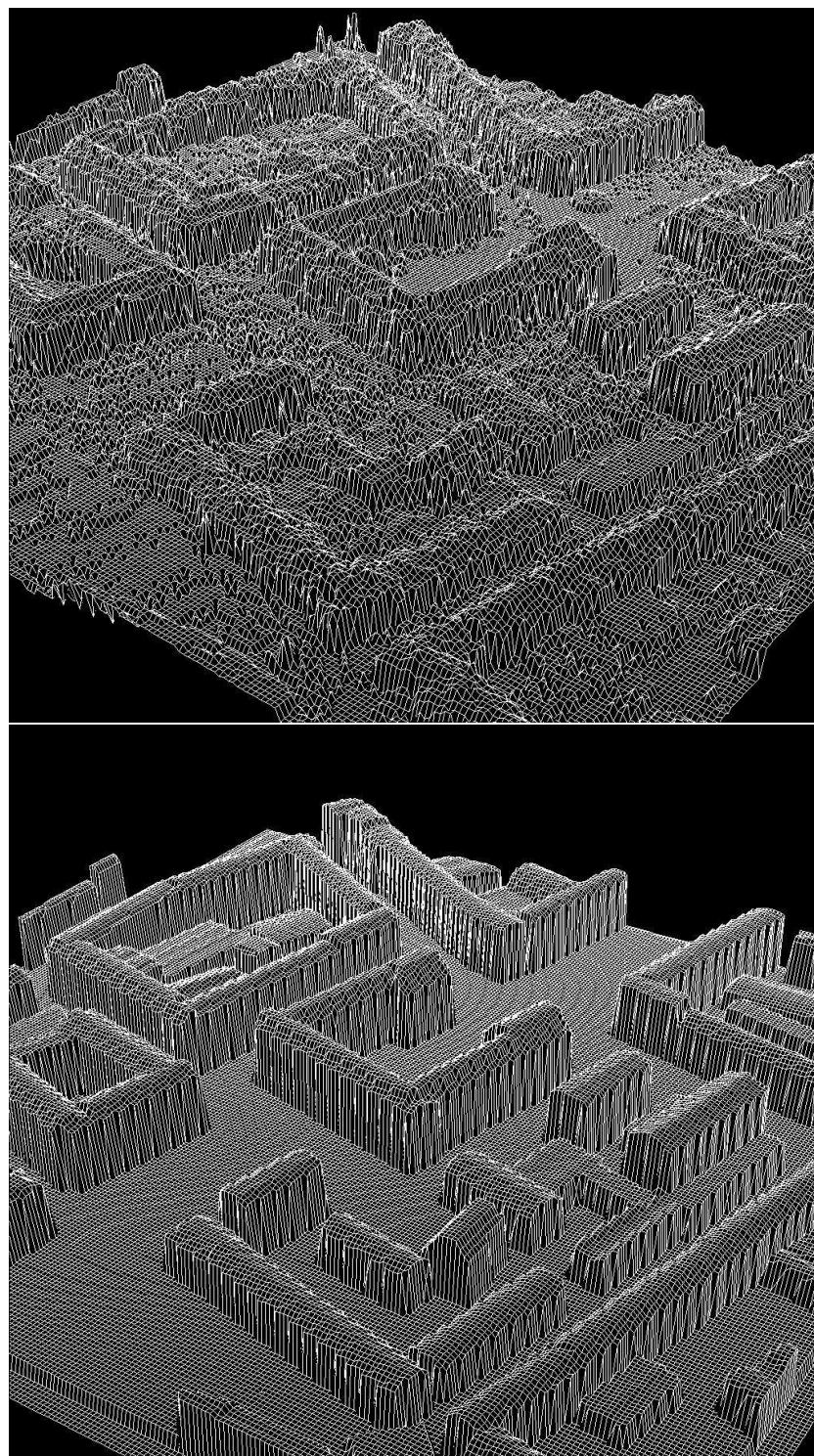


FIG. 3.33 – Visualisation en 3 dimensions (en haut : MNE initial, en bas : MNE interprété par la méthode proposée)



FIG. 3.34 – Visualisation en 3 dimensions avec un moteur graphique OpenGL

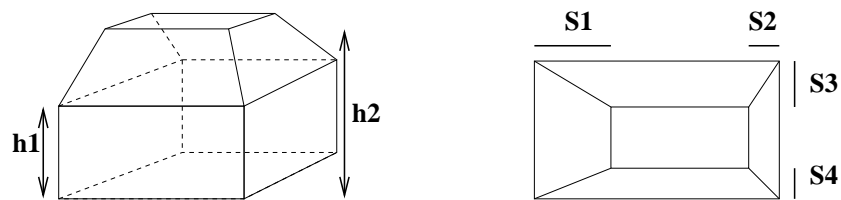


FIG. 3.35 – A gauche : bâtiment en 3 dimensions, à droite : bâtiment vu de haut

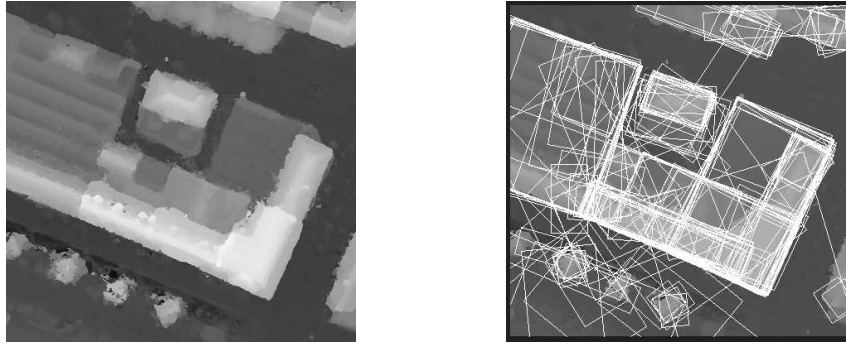


FIG. 3.36 – A gauche : détail d'un MNE de la ville d'Amiens (IGN), à droite initialisation utilisée.

données :

$$U_{données} = \frac{1}{|T|} \sum_{t \in T} d(h(\mathbf{x}, t), h_{ref}(t))$$

où  $d(\dots)$  est une distance sur  $\mathbb{R}$  (nous avons utilisé une fonction de coût qui correspond à la valeur absolue tronquée.)

Un bâtiment étant représenté par un élément de  $\mathbb{R}^3$  devient très coûteux en temps de calcul d'optimiser l'énergie sur l'ensemble des configurations de bâtiments. Pour passer outre ce problème, on utilise une prédétection : on suppose que l'ensemble des traces au sol rectangulaires plausibles est connu (on peut, par exemple, utiliser une transformée de Hough en 3D.) Cela donne un ensemble de points de  $K \times M$  :

$$\{v_1, \dots, v_N\} \quad \forall i \quad v_i \in K \times M$$

et on considère comme espace d'état un voisinage de ces points : on note  $B$  une boule symétrique de  $\mathbb{R}^3 \oplus$  l'addition au sens de Minkovski et on définit l'espace d'état réduit comme suit :

$$S_{réduit} = \left( \bigcup_{i \leq N} \{v_i\} \oplus B \right) \times M'$$

Ceci a l'avantage de considérablement réduire la taille de l'espace d'état. La boule  $B$  permet de perturber les prédétections et donc d'utiliser la partie a priori du modèle pour les régulariser. Il est, en fait, nécessaire d'avoir une sur-détection, ou alors de rendre plus complexe la notion de voisinage, traduite ici par la boule  $B$ .

L'espace d'état étant plus complexe (les boules centrées en les  $v_i$  n'étant pas forcément disjointes), il est plus difficile de s'assurer de la réversibilité des transformations utilisées par l'échantillonneur. Il faut donc recalculer les taux d'acceptation utilisés par l'algorithme de Metropolis Hastings Green et savoir à chaque instant quel  $v_i$  a permis l'apparition de chaque point.

**Résultats-** La figure 3.36 présente les données utilisées : une partie du MNE sur la ville d'Amiens d'une part, et la sur-détection initiale correspondante d'autre part. La dimension de la zone considérée est à peu près de  $100m$  sur  $100m$ . L'algorithme a mis 3 heures (prédétection incluse) sur une station SUN Blade 100<sup>10</sup> pour fournir le résultat, dont un détail est montré figure 3.37. Ce résultat se présente sous la forme d'un fichier texte décrivant 20 bâtiments.

Nous avons pu traiter ici un espace d'état très grand. Il faudrait, cependant, introduire des composantes déterministes dans l'algorithme pour accélérer les temps de calculs qui restent longs.

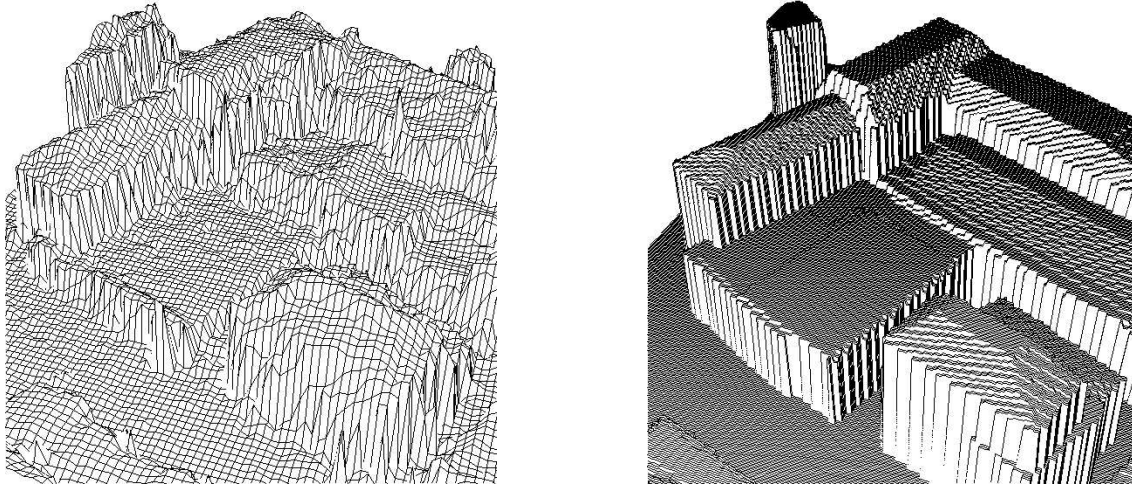


FIG. 3.37 – A gauche : détail du MNE initial en 3 dimensions, à droite : reconstruction obtenue.

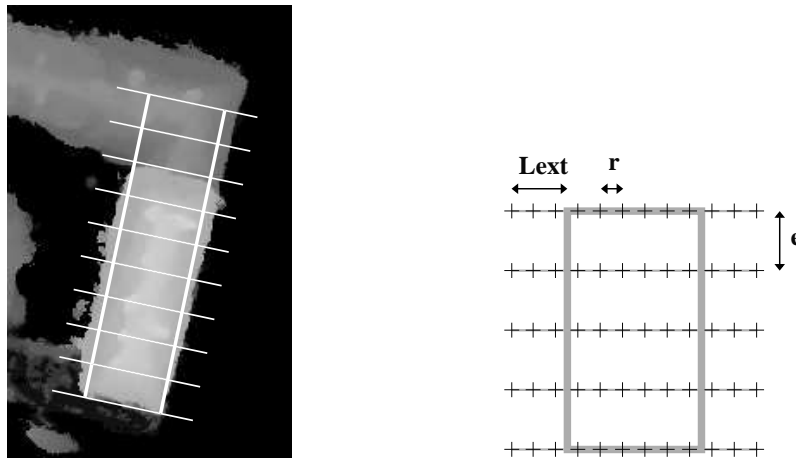
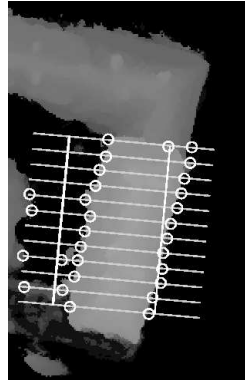


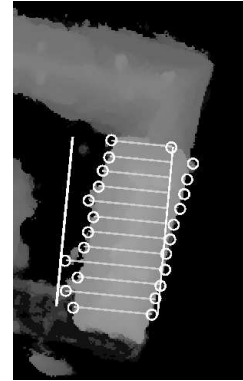
FIG. 3.38 – Un MNE un MNE et les profils retenus

**Un dernier modèle-** Dans ce dernier modèle, nous reprenons le modèle rectangulaire pour les bâtiments. Nous définissons un nouveau terme d'attache aux données pour pallier certaines inefficiences du premier modèle, notamment au niveau de la précision de la détection, et ajoutons quelques contraintes d'alignement au modèle a priori. Nous considérons des profils d'élévation perpendiculairement au rectangle considérée et régulièrement espacés (d'une distance  $e$  calculée à partir de la résolution du MNE) comme définis sur la figure 3.4.2. Nous considérons également un sous-échantillonnage des profils défini par le paramètre  $r$ , tandis que  $l_{\text{ext}}$  représente la distance minimale entre une discontinuité et le côté d'un rectangle pour que celle-ci soit prise en compte.

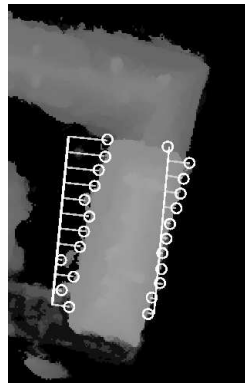
Les points de plus fort gradient sont alors détectés sur chaque profil. Cette détection fait appel à une procédure de filtrage pour éviter les détectés multiples sur un même bord de bâtiment (cas d'une façade non verticale due à la construction du MNE). Le couple parmi ces points le plus proche des côtés du rectangle candidat est alors conservé. La figure 3.39.a présente un rectangle candidat et les points de fort gradient détectés. La figure 3.39.b montre les couples retenus sur chacun des profils. Le terme d'attache aux données est alors fondé sur des quantités géométriques extraites à partir de ces couples. Un premier paramètre  $Lg(u)$  présenté sur la figure 3.39.d correspond à la proportion des points retenus suffisamment proche d'un côté du rectangle. Nous calculons, en réalité, deux quantités correspondant



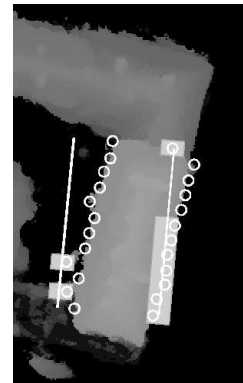
a) Points de fort gradient détectés.



b) Segments utilisés pour calculer le terme volumique.



c) Segments utilisés pour calculer le terme de moment.



d) Points suffisamment proches d'un côté du rectangle.

FIG. 3.39 – Les différents termes utilisés pour définir l'attache aux données.

aux deux côtés du rectangle et nous avons :  $L_{min} \leq Lg_1(u) \leq Lg_2(u) \leq L_{max}$ . Un terme volumique correspond au rapport des segments définis sur la figure 3.39.b sur la longueur totale des profils retenus. Ce terme va permettre de translater le rectangle candidat pour mieux coller aux données. Un terme similaire favorisant les rotations vers la forme écrite par les données est défini par un moment  $\bar{m}(u)$  correspondant à la moyenne des carrés des distances présentées sur la figure 3.39.c.

Nous définissons un ensemble  $\gamma$  d'objets attractifs (correspondant à des localisations vraisemblables des rectangles sur les données) comme l'ensemble des objets dont les côtés sont suffisamment proches des points de forts gradients. Si  $th_1 \leq th_2$  désignent deux seuils dans  $]0,1[$ ,  $\gamma$  est défini par :

$$\gamma_0 = \left\{ u \in S \quad \text{t.q.} \quad \begin{array}{l} Lg_1(u) \geq th_1 * L(u) \\ \text{et} \\ Lg_2(u) \geq th_2 * L(u) \end{array} \right\} \quad (3.96)$$

Une énergie  $J$  est associée aux éléments de cet ensemble :

$$\begin{aligned} J_0 : S &\rightarrow [0, -1] \\ u \in \gamma_0 &\rightarrow - \left( \frac{1}{4} \frac{Lg_1(u) + Lg_2(u)}{L_{max}} + \frac{1}{2} \frac{l(u)}{l_{max}} \right) \\ u \notin \gamma_0 &\rightarrow 0 \end{aligned} \quad (3.97)$$

Pour faciliter la dynamique du processus vers les objets attractifs, nous définissons un ordre sur les objets non attractifs. Nous introduisons deux seuils  $v_{min}$  et  $m_{max}$ , et les trois ensembles suivants :



$$\gamma_1 = \left\{ u \in S : \begin{array}{l} u \notin \gamma_0 \\ \bar{v}(u) \geq v_{min} \\ \bar{m}(u) \leq m_{max} \end{array} \right\} \quad (3.98)$$

$$\gamma_2 = \left\{ u \in S : \begin{array}{l} u \notin \gamma_1 \cup \gamma_0 \\ \bar{v}(u) \geq v_{min} \text{ ou } \bar{m}(u) \leq m_{max} \end{array} \right\} \quad (3.99)$$

$$\gamma_3 = \{u \in S : u \notin \gamma_2 \cup \gamma_1 \cup \gamma_0\} \quad (3.100)$$

Les quatre ensembles  $\gamma_i$  forment donc une partition de  $S$ . L'ensemble  $\gamma_0$  étant de taille faible relativement aux trois autres, il faudra en tenir lors de l'optimisation car il représente l'ensemble des objets d'intérêt.

Une énergie,  $J(u) \in ]0,1]$  si  $u \in \gamma_i$  et  $J_i(u) = 0$  sinon, est associée à chacun des ensembles  $\gamma$  (différents de  $\emptyset$ ). Le terme d'attache aux données est alors défini par :

$$U_d(\mathbf{x}) = \sum_{u \in \mathbf{x}} J_0(u) + 0.001(\mathbf{1}_{\gamma_1}(u) + J_1(u)) \\ + 0.01(\mathbf{1}_{\gamma_2}(u) + J_2(u)) + 0.1(\mathbf{1}_{\gamma_3}(u) + J_3(u)) \quad (3.101)$$

Pour définir le modèle a priori, nous considérons plusieurs interactions modélisant des propriétés différentes de la configuration. Ces propriétés concernent l'alignement de bâtiments, le contact entre bâtiments voisins et la non-superposition.

**Résultats-** Les résultats présentés sur les données Laser (cf figure 3.41) et sur le MNE ont été obtenus en ajustant les paramètres empiriquement. Néanmoins, la densité complète du modèle peut s'écrire sous forme exponentielle d'une énergie dépendant linéairement des paramètres. Une procédure de type MVMCMC, comme celle décrite dans le paragraphe 3.3.2 est donc envisageable.

Nous validons les résultats par comparaison avec une vérité terrain fournie par l'IGN. Il faut noter que cette vérité terrain, obtenue par des experts à partir d'images aériennes, est plus précise que le MNE dont nous disposons. En outre, certains bâtiments ne sont pas retranscrits sur cette vérité terrain. La comparaison est donc donnée à titre indicatif mais ne représente pas un critère absolu. Pour un résultat donné, c'est à dire une configuration de rectangles, nous calculons la surface des bâtiments non-détectés et la surface des bâtiments sur-détectés. Cela donne trois classes : les pixels bien détectés (en gris), les pixels des bâtiments non détectés (en noir) et les pixels sur-détectés (en blanc).

Un premier résultat est obtenu sur des données laser d'une résolution de 0.5m par pixel sur une zone d'intérêt montrée sur la figure 3.40.

La figure 3.41 montre les données, la vérité terrain, les bâtiments estimés et les erreurs de classification. Nous pouvons constater que le manque de précision des données induit une imprécision sur les bords des bâtiments.

Le second résultat (voir la figure 3.42) est obtenu sur un MNE construit par un algorithme de corrélation sur un couple d'images optiques. La résolution est de 0.2m par pixel et 0.1m en élévation. On peut voir de nombreux artefacts sur le MNE. Les paramètres impliqués dans le modèle a priori et l'optimisation sont les mêmes que pour le résultat précédent. Ici, certains arbres ont été assimilés à des bâtiments.

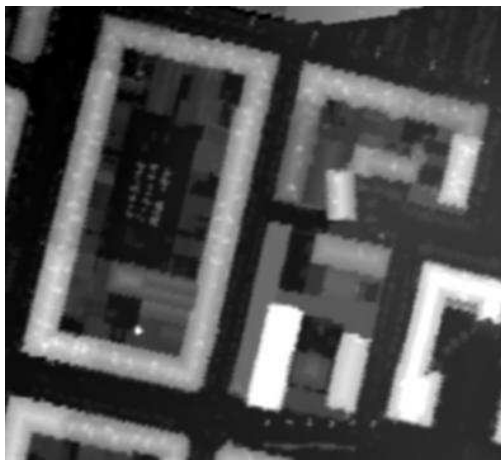
Le troisième résultat est obtenu sur une partie de la vieille ville pour laquelle de nombreux arbres et des maisons de petite taille rendent le problème difficile. Les résultats sont montrés sur la figure 3.43.

Le dernier résultat, montré sur la figure 3.44, est obtenu à partir du MNE optique sur une zone contenant beaucoup de cours d'immeubles. Les discontinuités sont ici difficiles à détecter.

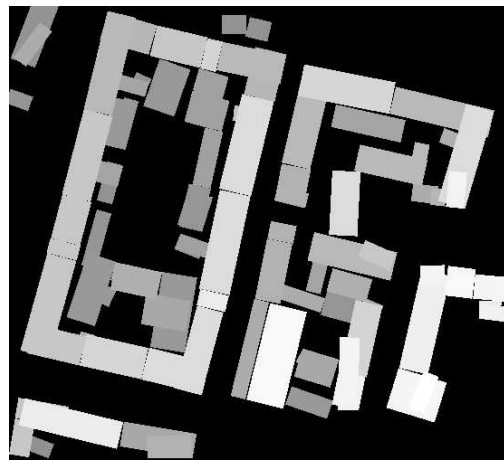
Le tableau 3.2 récapitule les différents taux de mauvaise détection. Notons que chacun de ces résultats a nécessité un temps de calcul de trois heures pour un processeur de 1Ghz.



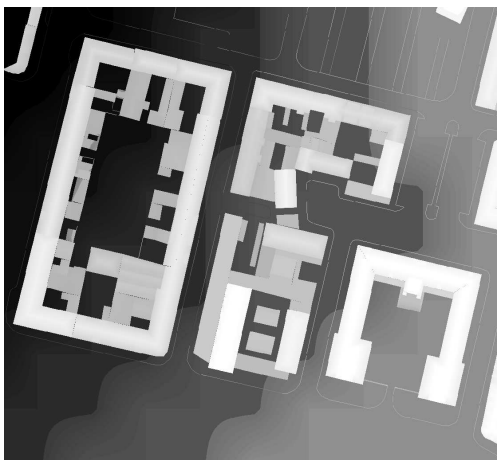
FIG. 3.40 – Photographie aérienne (20cm) de la première zone d'intérêt sur Amiens (données fournies par l'IGN).



Données laser



Plan cadastral estimé



Vérité terrain



Classification des erreurs

FIG. 3.41 – Résultat sur la première zone d'intérêt à partir des données laser.

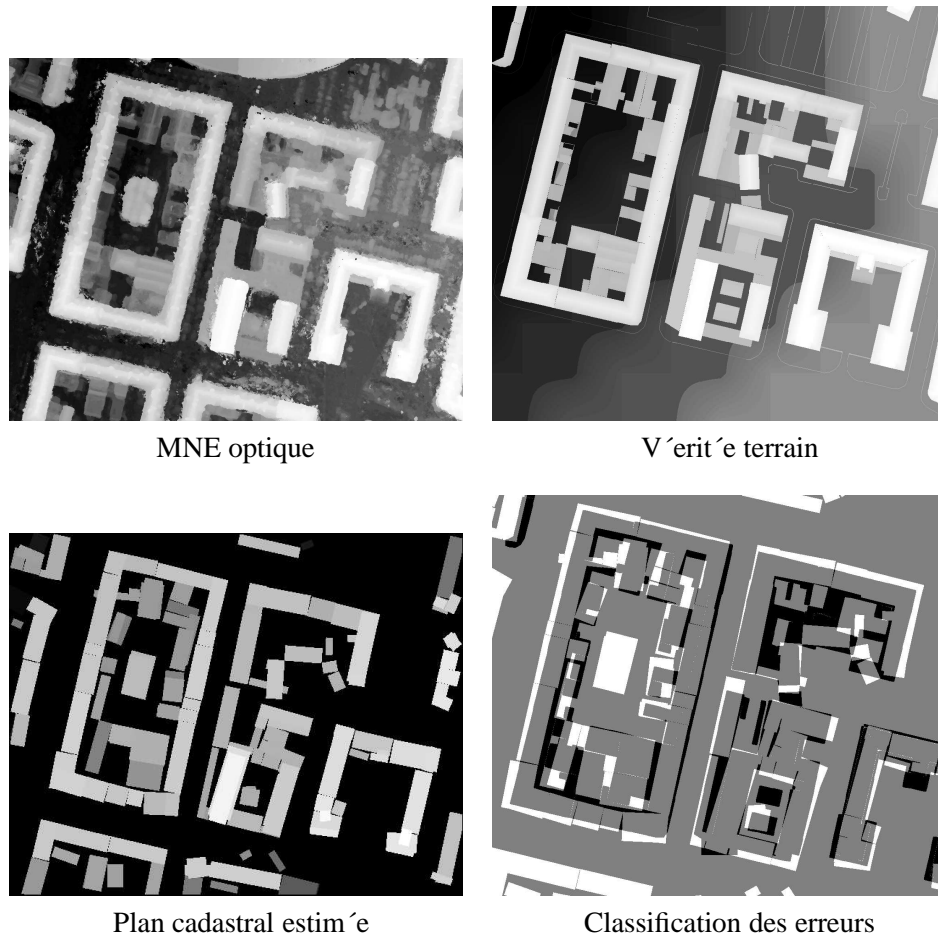


FIG. 3.42 – Résultat sur la première zone d'intérêt à partir du MNE optique.

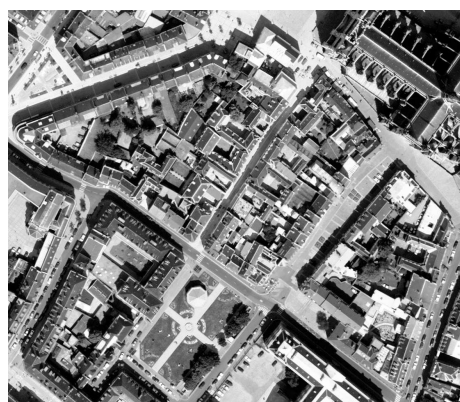
TAB. 3.2 – Taux de mauvaise classification pour chacun des résultats présentés.

Res. 1: données LASER	
Mauvaise détection (noir)	5%
Fausses alarmes (blanc)	11%

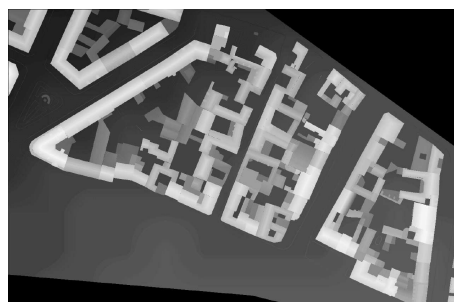
Res. 2 : MNE optique	
Mauvaise détection (noir)	8.9%
Fausses alarmes (blanc)	10.6%

Res. 3 : MNE optique	
Mauvaise détection (noir)	8.01%
Fausses alarmes (blanc)	5.3%

Res. 4 : MNE optique	
Mauvaise détection (noir)	11.4%
Fausses alarmes (blanc)	5.9%



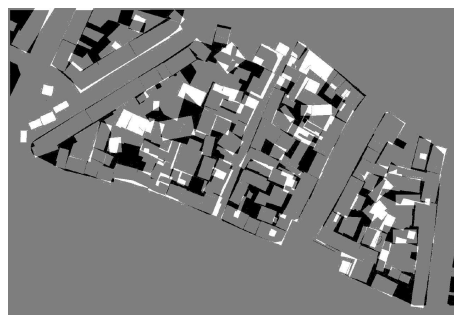
Photographie aérienne (25cm)



Vérité terrain



Plan cadastral estimé



Classification des erreurs

FIG. 3.43 – *Seconde zone d'intérêt : vieille ville d'Amiens.*

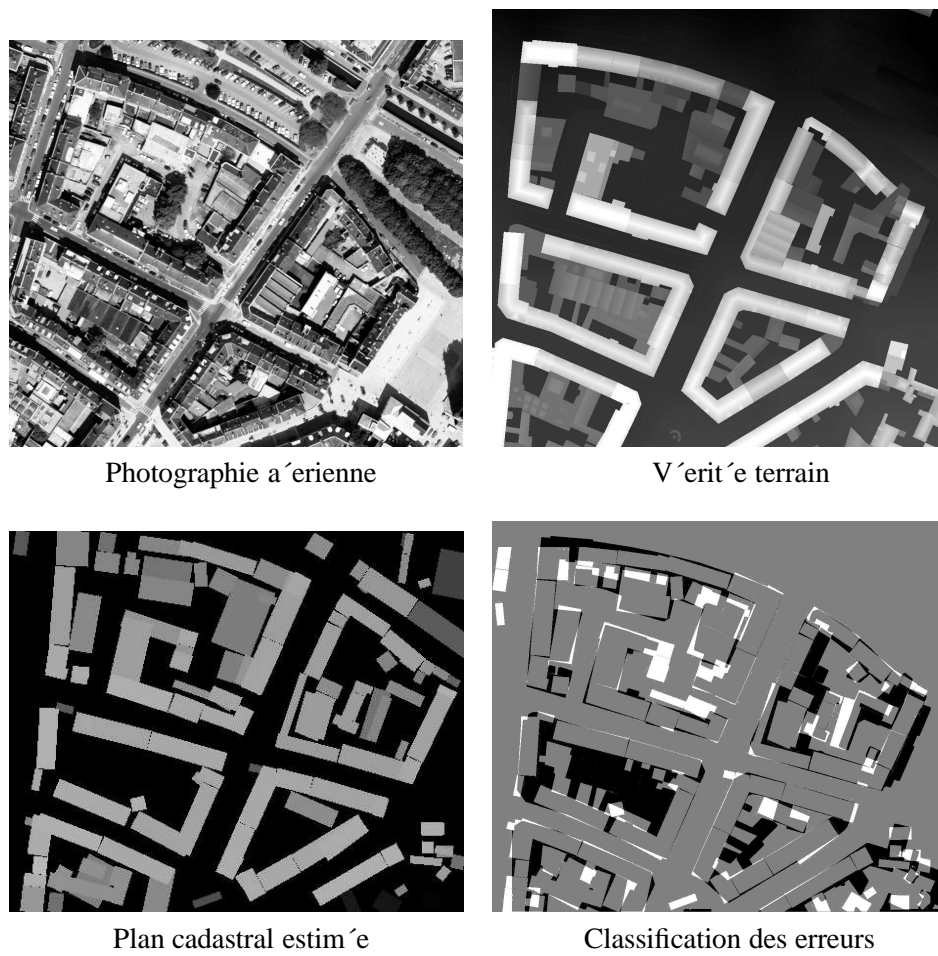


FIG. 3.44 – Troisième zone d'intérêt, Amiens

### 3.4.3 Segmentation

<sup>11</sup> Nous abordons ici le problème de la segmentation. Nous construisons un processus d'objets simples (des triangles équilatéraux) pour lequel nous définissons une densité a priori favorisant les configurations proches d'une partition du plan. L'attache aux données est calculée au niveau de l'objet et non du pixel. Nous pouvons donc espérer une plus grande robustesse au bruit, notamment dans le cas d'un bruit corrélié dû au capteur ou à certains objets parasites (voiture sur une route par exemple). Cette approche est particulièrement adaptée aux données haute et très haute résolution. Nous la testons sur des images aériennes.

**Le modèle-** Nous considérons une famille exponentielle de densités non normalisées par rapport à la mesure de Poisson, définie par :

$$h_{\theta}(\mathbf{x}, Y) = \exp \langle t(\mathbf{x}, Y), \theta \rangle \quad (3.102)$$

où  $t(\mathbf{x}, Y)$  représente les statistiques canoniques :

$$(n(\mathbf{x}), -U_1(\mathbf{x}), -U_2(\mathbf{x}), -U_3(\mathbf{x}), -U_4(\mathbf{x}, Y), -U_5(\mathbf{x}, Y))^t \quad (3.103)$$

avec  $\theta = (\ln \beta, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5)^t$  le vecteur de paramètre. En exprimant le produit scalaire  $\langle t(\mathbf{x}, Y), \theta \rangle$ , il vient :

$$h_{\theta}(\mathbf{x}, Y) = \beta^{n(\mathbf{x})} e^{-\gamma_1 U_1(\mathbf{x}) - \gamma_2 U_2(\mathbf{x}) - \gamma_3 U_3(\mathbf{x}) - \gamma_4 U_4(\mathbf{x}, Y) - \gamma_5 U_5(\mathbf{x}, Y)} \quad (3.104)$$

où  $n(\mathbf{x})$  est le nombre d'objets dans la configuration  $\mathbf{x}$ , et les  $U_i$  sont des fonctions de potentiel. Le modèle a priori est constitué de trois types de potentiels modélisant chacun une propriété d'esirée de la configuration recherchée :

- $U_1(\mathbf{x})$  **est un potentiel répulsif.** Le but de ce potentiel est d'éviter le recouvrement entre les objets. Il est défini sur des paires d'objets comme un processus ponctuel de Markov au sens de Ripley-Kelly avec comme voisinage la relation d'intersection ( $\sim$ ) :

$$U_1(\mathbf{x}) = \sum_{\{\eta, \xi\} \subset \mathbf{x}: \eta \sim \xi} \phi_1(\eta, \xi) \quad (3.105)$$

où :

$$\phi_1(\eta, \xi) = \left( \frac{s(\eta, \xi)}{s_R} \right)^4 \text{ et } s(\eta, \xi) = \frac{|Z(\eta) \cap Z(\xi)|}{\min(|Z(\eta)|, |Z(\xi)|)}. \quad (3.106)$$

$Z(\eta)$  est la projection de l'objet  $\eta$  sur l'image  $Y$ ,  $|Z(\eta)|$  la surface de cette projection et  $s_R$  un seuil donné.

- $U_2(\mathbf{x})$  **est un potentiel de surface.** Le but de ce potentiel est de favoriser les grands triangles. Il est défini, sur chaque objet, comme suit :

$$U_2(\mathbf{x}) = \sum_{\eta \in \mathbf{x}} \phi_2(\eta) \text{ où } \phi_2(\eta) = -\frac{|Z(\eta)| - s_{min}}{s_{max} - s_{min}}. \quad (3.107)$$

$s_{min}$  et  $s_{max}$  sont respectivement les surfaces minimale et maximale qu'un objet de la configuration peut prendre.

- $U_3(\mathbf{x})$  **est un potentiel d'orientation.** Le but est de favoriser les configurations pour lesquelles deux triangles voisins sont en position tête-bêche. Il est défini sur des paires de triangles comme un processus ponctuel de Markov au sens de Ripley-Kelly avec la relation de voisinage ( $\sim$ ) :

$$U_3(\mathbf{x}) = \sum_{\{\eta, \xi\} \subset \mathbf{x}: \eta \sim \xi} \phi_3(\eta, \xi) \quad (3.108)$$

---

11. Nous écrivons ici le travail de thèse de S. Drot, boursier CIFRE IGN/INRIA, bibliographie : [41]

où :

$$\phi_3(\eta, \xi) = \frac{2\sqrt{3}}{3} \left| \sin(|\theta_\eta - \theta_\xi| - \frac{\pi}{3}) \right|. \quad (3.109)$$

$\theta_\eta$  est l'orientation de l'objet  $\eta$  et  $|a|$  la valeur absolue de  $a$ .

La figure 3.45 montre une simulation du modèle a priori ainsi défini. Nous constatons que nous sommes proches d'une partition de l'image.

A ce modèle sont ajoutés deux termes d'attache aux données permettant de localiser les triangles sur les zones homogènes sans franchir les contours. Ils sont définis sur chaque objet et dépendent de l'image à segmenter  $Y$  :

- $U_4(\mathbf{x}, Y)$  **est un potentiel d'homogénéité.** Le but de ce potentiel est de placer les objets de la configuration sur les zones homogènes de l'image considérée. Le triangle est découpé en quatre régions de même surface. Un test de Kolmogorv-Smirnov est calculé pour chaque paire de ces sous-régions. Nous considérons alors la valeur maximale de ces différents tests pour imposer la similarité des distributions dans chacune des sous-régions. Pour finir, cette valeur maximale est seuillée par une fonction sigmoïde.
- $U_5(\mathbf{x}, Y)$  **est un potentiel de gradient.** Le but de ce potentiel est de coller les objets le long des contours. Pour cela, nous considérons le maximum de la moyenne des gradients le long de chaque arête du triangle considérée. Les triangles aboutissant à une forte valeur de ce terme sont favorisés.

**Résultats-** Les mouvements définissant le noyau de proposition de l'algorithme RJMCMC utilisés sont les mêmes que pour l'application précédente, c'est-à-dire, naissance-mort, translation, rotation et dilatation. Par un recuit simulé (2000000 transitions), nous obtenons une première segmentation. Il s'agit en réalité d'une sur-segmentation car une même zone de l'image peut-être recouverte par plusieurs triangles. En outre, il nous faut gérer les intersections résiduelles entre triangles ainsi que les petits espaces non recouverts par des triangles. Nous effectuons donc un post-traitement qui se déroule en deux étapes. Nous considérons la sur-segmentation définie par les triangles et les zones d'intersection entre plusieurs triangles. A cette sur-segmentation est attaché un graphe dont les noeuds sont définis par les régions et les arêtes par la relation de connexité. A chaque arête est associée un coût de fusion défini par un écart entre les distributions des niveaux de gris des régions associées aux noeuds reliant l'arête considérée. Les deux régions correspondant à l'arête de plus faible coût sont fusionnées, si celui-ci est inférieur à un seuil prédéfini. L'algorithme est itéré jusqu'à ce que tous les coûts soient supérieurs à ce seuil. A ce niveau, nous obtenons une segmentation avec quelques lacunes aux endroits non recouverts par les triangles. Nous considérons alors un modèle de Potts défini sur les sites non étiquetés. Les classes du modèle et leurs paramètres sont appris sur les régions déjà étiquetées. La figure 3.47 montre le résultat final obtenu à partir de l'image de la figure 3.46.

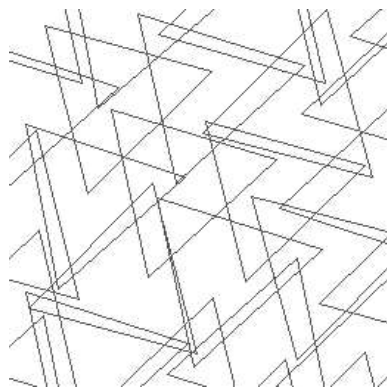


FIG. 3.45 – Réalisation du modèle a priori



FIG. 3.46 – Image aérienne (résolution 50cm)



FIG. 3.47 – Segmentation finale



### 3.5 Conclusion sur les Processus Ponctuels Marqués

Les processus ponctuels marqués permettent donc de dépasser les limites des champs de Markov énoncées en conclusion de la première partie de ce manuscrit. Ils s'interprètent donc comme une extension des champs de Markov à la modélisation par objets. Nous avons montré que leur mise en oeuvre permet de résoudre des problèmes difficiles et fournir des résultats très encourageants. En effet, même s'ils n'ont pas encore atteint le degré de maturité des champs de Markov, des problèmes concrets, aussi bien concernant le ponctuel, le linéique et le surfacique, ont été résolus par cette approche.

La mise en oeuvre et le coût algorithmique de cette approche sont néanmoins nettement plus lourds que pour l'approche markovienne. Cet outil n'a donc pas pour vocation de remplacer les champs de Markov mais simplement d'apporter des solutions lorsque les limites de ces derniers sont atteintes. Le point fort de cette nouvelle approche est la notion d'objet, avec les aspects géométriques que cela comporte. En revanche, lorsque la notion d'objet n'est pas prépondérante dans l'application traitée, l'intérêt des processus ponctuels marqués reste à démontrer. Ainsi, les résultats que nous avons obtenus pour la segmentation d'image ne sont pas à la hauteur de la complexité de l'approche. Une approche fondée sur une partition du plan aurait été plus adaptée.

Il est sans doute trop tôt, pour conclure définitivement sur l'apport des processus ponctuels objets en analyse d'image. Pour obtenir des algorithmes utilisables dans un contexte applicatif pratique, c'est-à-dire dans un cadre industriel, des recherches supplémentaires, concernant notamment l'optimisation et l'estimation, sont nécessaires. Les premiers résultats obtenus nous permettent, cependant, d'être optimistes.

## Chapitre 4

# Conclusion et perspectives

Dans la première partie de ce mémoire, nous avons montré la grande généralité de l'approche markovienne en analyse d'image. Une telle approche se décline suivant trois axes : modélisation, estimation et optimisation. Après avoir écrit quelques contributions à ces trois domaines, nous avons abordé un très grand nombre d'applications. Nous avons pu notamment voir la diversité des images traitées (IRM, angographiques, satellitaires, aériennes, optiques, radar,...) et des thématiques abordées (restauration, segmentation, extraction d'objets,...). Cette approche n'est pas un domaine de recherche clos. En effet, des améliorations peuvent sans aucun doute être apportées par de nouveaux modèles plus performants et plus fidèles aux objets recherchés. En outre, l'efficacité algorithmique peut être augmentée par de nouvelles méthodes d'optimisation et d'estimation. Néanmoins, ce domaine de recherche est suffisamment mûr pour que l'on puisse en entrevoir les limites. Ces limites concernent essentiellement la prise en compte de contraintes géométriques fortes dans les modèles. Or avec le caractère de plus en plus résolu des capteurs, la géométrie devient une information prépondérante pour l'analyse des images. Ces remarques nous ont conduit à étudier les processus ponctuels marqués qui permettent d'allier les atouts des approches stochastiques (robustesse au bruit, prise en compte du savoir de l'expert dans un modèle a priori), avec la prise en compte non seulement de l'information géométrique au niveau de l'objet, mais également de la répartition de ces objets.

Les applications développées à partir d'une modélisation par processus ponctuels marqués répondent bien aux enjeux fixés par la prise en compte de la géométrie des structures recherchées, de la forme des objets (arbres, bâtiments,...) mais aussi de la topologie d'un réseau linéique par exemple (réseau routier). Elles montrent la robustesse de l'approche par rapport au bruit des capteurs, mais aussi par rapport à certains objets parasites (voitures, ...) visible à haute et très haute résolution ou encore aux artefacts issus d'un pré-traitement (imperfection d'un MNE par exemple). En résumé, la définition de critères géométriques apporte, outre une meilleure sensibilité de la détection, une robustesse vis-à-vis des structures corréelées indésirables. Par ailleurs, la diversité des domaines applicatifs abordés montre le pouvoir de modélisation de l'approche. Ces travaux ont pu aboutir car nous disposons des outils théoriques nécessaires. Les algorithmes d'optimisation utilisés pour les champs de Markov sont généralisables aux processus ponctuels marqués tout comme l'algorithmie nécessaire à l'estimation des paramètres.

Les enjeux actuels pour généraliser ce type d'approche concerne l'optimisation, qui induit encore des temps de calcul relativement longs, notamment dans l'optique d'une utilisation commerciale ou industrielle de ce type de modèle. En outre, la réduction des temps de calcul permettra d'aborder de façon pratique l'estimation des paramètres et de fournir ainsi des approches non supervisées. Pour finir, nous comptons aborder la modélisation de scènes complètes avec ce type de processus en considérant une famille d'objets distincts (tronçons routier, bâtiments, arbres,...) pouvant interagir entre eux.



# Bibliographie

- [AG92] R. Azencott et C. Graffigne. Non supervised segmentation using multi-level Markov random fields. In *ICPR*, 1992.
- [Aka74] H. Akaike. A new look at the statistical model identification. *IEEE trans. on Automatic Control*, 19(6):716–723, 1974.
- [aYCK91] U. Grenander, Y. Chou et D.M. Keenan. *Hands - A Pattern Theoretic Study of Biological Shapes*. Springer-Verlag, 1991.
- [BC96] M. Barzohar et D.B. Cooper. Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(2):707–721, 1996.
- [Bes74] J. Besag. Spatial interaction and the statistical analysis of lattice system. *J. of the royal statistical society, series B*, 36:192–236, 1974.
- [Bez82] J.C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, 1982.
- [BGR03] S.P. Brooks, P. Guidici, et G.O. Roberts. Efficient construction of Reversible Jump MCMC proposal distributions (with discussion). *Journal of the Royal Statistical Society, Series B*, 65:3–55, 2003.
- [BL91] C. Bouman et B. Liu. Multiple resolution segmentation of textured images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(2):99–113, 1991.
- [BL93] A. Baddeley et M.N.M. Van Lieshout. Stochastic geometry models in high-level vision. *Statistic and Images, 1, supplement to Journal of Applied Statistics*, 20:231–256, 1993.
- [BP90] A. Baraldi et F. Parmiggiani. Urban area classification by multispectral SPOT images. *IEEE Trans. on Geoscience and Remote Sensing*, 28(4):674–680, 1990.
- [Car87] M.J. Carlotto. Histogram analysis using a scale-space approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(1):121–129, 1987.
- [CHL95] M. Chan, G. T. Herman, et E. Levitan. Bayesian image reconstruction using a high-order interacting Markov random field model. In *8th International Conference on Image Analysis and Processing*, pages 609–614, 1995.
- [CJ83] G.R. Cross et A.K. Jain. Markov random field texture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5(1):25–39, 1983.
- [CJ93] R. Chellappa et A.K. Jain, editors. *Markov Random Fields: Theory and Applications*. Academic Press, 1993.
- [CK01] C. Chalopin et F. Kruggel. Automatic segmentation of focal brain lesions from MR Images. preprint, 2001.
- [Dan00] P. Daniel. *Peut-on extraire le relief à partir d'une seule image?*. Thèse de doctorat, IRIT, univ. Paul Sabatier, 2000. no 3669.

- [DE87] H. Derin et H. Elliot. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(1):39–55, 1987.
- [DM96] J.D. Durou et H. Maître. On convergence in the methods of Strat and of Smith for shape from shading. *Int. Journal on Computer Vision*, 17(3):273–289, 1996.
- [Dur93] J.D. Durou. *Reconnaissance du relief à partir de l'éclairément*. Thèse de doctorat, Université Paris Sud, 1993.
- [FCF<sup>+</sup>95] S.D. Forman, J.D. Cohen, M. Fitzgerald, W.F. Eddy, M.A. Mintun, et D.C. Noll. Improved assessment of significant activation in fMRI: Use of a cluster-size threshold. *Magnetic Resonance in Medicine*, 33:636–647, 1995.
- [FHP<sup>+</sup>95] K.J. Friston, A.P. Holmes, J.B. Poline, P.J. Grasby, S.C.R. Williams, R.S.J. Frackowiak, et R. Turner. Analysis of fMRI time-series revisited. *NeuroImage*, 2:45–53, 1995.
- [FJT94] K.J. Friston, P. Jezzard, et R. Turner. Analysis of fMRI time-series. *Human Brain Mapping*, 2:49–78, 1994.
- [FKL<sup>+</sup>98] A. Fischer, T.H. Kolbe, F. Lang, A.B. Cremers, W. Förstner, L. Plüme, et V. Steinhage. Extracting buildings from aerial images using hierarchical aggregation in 2D and 3D. *Computer Vision and Image Understanding*, 72(2):185–203, 1998.
- [FRM99] M. Fradkin, M. Roux, et H. Maître. Building detection from multiple views. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, 1999.
- [FT74] J.H. Friedman et J.W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE trans. On Computers*, 23:881–889, 1974.
- [FTW81] M.A. Fischler, J.M. Tenenbaum, et H.C. Wolf. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. *Computer Graphics and Image Processing*, 15:201–223, 1981.
- [FWF<sup>+</sup>94] K.J. Friston, K.J. Worsley, R.S.J. Frackowiak, J.C. Mazziotta, et A.C. Evans. Assessing the significance of focal activations using their spatial extent. *Human Brain Mapping*, 1:210–220, 1994.
- [GG84] S. Geman et D. Geman. Stochastic relaxation, Gibbs distribution, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- [Gim99] G. Gimel'farb. *Images Textures and Gibbs Random Fields*. Kluwer Academic Publishers, Dordrecht, 1999.
- [GJ96] D. Geman et B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18:1–14, 1996.
- [GM94] C.J. Geyer et J. Moller. Simulation and likelihood inference for spatial point process. *Scandinavian Journal of Statistics series B*, 21:359–373, 1994.
- [GO94] A. Goshtasby et W.D. O'Neil. Curve fitting by a sum of gaussian. *Graphical Models and Image Processing*, 65(4):281–288, 1994.
- [Gou95] 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):274–284, 1995.
- [Gou96] C. Gouinaud. *Traitement d'images satellitaires pour la détection d'agglomérations*. Thèse de doctorat, ENST, 1996.
- [Gou98] 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, *International Forum*

- on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, pages 11–23, 1998.
- [GR92] D. Geman et G. Reynolds. Constrained restoration and recovery of discontinuities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(3):367–383, 1992.
- [Gra87] C. Graffigne. *Experiments in Texture Analysis and Segmentation*. PhD thesis, Brown University, 1987.
- [Gre95] P. Green. Reversible jump MCMC computation and bayesian model determination. *Biometrika*, 1995.
- [GT92] C.J. Geyer et E.E. Thompson. Constrained Monte Carlo maximum likelihood for dependent data. *J. of the royal statistical society, series B*, 54:657–699, 1992.
- [Har79] R.M. Haralick. Statistical and structural approaches to texture. *Proc. of the IEEE*, 67(5):786–804, 1979.
- [Haz00] G.G. Hazel. Multivariate gaussian MRF for multispectral scene segmentation and anomaly detection. *IEEE Trans. on Geoscience and Remote Sensing*, 38(3):1199–1211, 2000.
- [HB89] B.K.P. Horn et M.J. Brooks. *Shape from Shading*. MIT Press, Cambridge, Massachusetts, 1989.
- [HG91] S. Houzelle et G. Giraudon. Data fusion using SPOT and SAR images for bridge and urban area extraction. In *IGARSS*, pages 1455–1458, 1991.
- [HJ87] R. Hoffman et A.K. Jain. Segmentation and classification of range images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(5):608–620, 1987.
- [HK98] S.A. Hojjatoleslami et J. Kittler. Region growing: a new approach. *IEEE Trans. on Image Processing*, 7:1079–1083, 1998.
- [Hor75] B.K.P. Horn. *The Psychology of Computer Vision*, chapter 4:Obtaining shape from shading information, pages 115–155. McGraw-Hill, New York, 1975.
- [Hub85] P.J. Huber. Projection Pursuit. *The annals of Statistics*, 13(2):435–475, 1985.
- [Hug68] G.F. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Trans. on Information Theory*, 14(6):55–63, 1968.
- [JL99] L.O. Jimenez et D.A. Landgrebe. Hyperspectral data analysis and supervised feature reduction via projection pursuit. *IEEE Trans. on Geoscience and Remote Sensing*, 37(6):2653–2667, 1999.
- [KK94] R. Krishnapuram et J.M. Keller. Fuzzy and possibilistic clustering methods for computer vision. In *SPIE, institue series in neural and fuzzy systems*, volume 12, pages 133–159, 1994.
- [Lar99] M. Larsen. Individual tree top position estimation by template voting. In *International Airbone Remote Sensing Conference and Exhibition*, pages 83–90, 1999.
- [Lig85] Th. Liggett. *Interacting particle systems*. Springer-Verlag, 1985.
- [LJ90] J.S. Lee et I. Jurkevich. Coastline detection and tracing in SAR images. *IEEE Trans. on Geoscience and Remote Sensing*, 38(4):662–668, 1990.
- [LJ91] S.P. Liou et R.C. Jain. An approach to three-dimensional image segmentation. *CVGIP: Image Understanding*, 53(3):237–252, 1991.
- [LJC92] Z. Liang, R.J. Jaszczak, et R.E. Coleman. Parameter estimation of finite mixtures using the EM algorithm and information criteria with application to medical image processing. *IEEE Trans. on Nuclear Science*, 39:1126–1133, 1992.
- [LMH94] Z. Liang, J.R. MacFall, et D.P. Harrington. Parameter estimation and tissue segmentation from multispectral MR Images. *IEEE Trans. on Medical Imaging*, 13(3):441–449, 1994.

- [Lor99] A. Lorette. *Analyse de texture par méthodes markoviennes et par morphologie mathématique ; application à l'analyse des zones urbaines sur des images satellitales*. Thèse de doctorat, Université de Nice-Sophia Antipolis, 1999.
- [LR97] M. Larsen et M. Rudemo. Using ray-traced templates to find individual trees in aerial photographs. In J. Parkkinen M. Frydrych and A. Visa, editors, *Scandinavian Conference on Image Analysis*, volume 2, pages 1007–1014, 1997.
- [May99] H. Mayer. Automatic object extraction from aerial imagery - a survey focusing on buildings. *Computer Vision and Image Understanding*, 74(2):138–149, 1999.
- [MBP95] C.R. Meyer, P.H. Bland, et J. Pipes. Retrospective correction of intensity inhomogeneities in MRI. *IEEE Trans. on Medical Imaging*, 14(1):36–41, 1995.
- [MHK<sup>+</sup>96] T.R. Metcalf, H.S. Hudson, T. Kosugi, R.C. Puetter, et R.K. Pina. Fractal pixon image reconstruction for Yohkoh's Hard X-Ray telescope. *ApJ*, 466:585–594, 1996.
- [MLBS97] H. Mayer, I. Laptev, A. Baumgartner, et C. Steger. Automatic road extraction based on multi-scale modeling, context and snakes. *International Archives of Photogrammetry and Remote Sensing*, 32:106–113, 1997.
- [MS67] R.A. Minlos et Ya.G. Sinai. The phenomenon of "phase separation" at low temperatures in some lattice models of a gas i. *Math. USSR-Sb*, 2:337–395, 1967.
- [MS68] R.A. Minlos et Ya.G. Sinai. The phenomenon of "phase separation" at low temperatures in some lattice models of a gas ii. *Trans. Moscow Math. Soc.*, 19:121–196, 1968.
- [Mum96] D. Mumford. Pattern theory: a unifying perspective. In D.Knill and W.Richards, editors, *Perception as Bayesian Inference*. Cambridge Univ. Press, 1996.
- [Mum02] D. Mumford. Pattern theory: The mathematics of perception. In *International Congress of Mathematicians*, volume 1. Higher Educ. Press, 2002.
- [Mus98] S. Mustiere. Généralisation adaptative du linéaire basé sur la détection des empâtements, application au routier. Bulletin d'information de l'IGN, 1998. 69.
- [MZ96] N. Merlet et J. Zerubia. New prospects in line detection by dynamic programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(4):426–431, 1996.
- [NC93] H.H. Nguyen et P. Cohen. Gibbs random fields, fuzzy clustering, and the unsupervised segmentation of textured images. *Computer Vision, Graphics, and Image Processing*, 55(1):1–19, 1993.
- [nPFISK97] W.N. Neueneschwander nad P. Fua, L. Iverson, G. Szekey, et O. Kubler. Ziplock snakes. *Int. Journal on Computer Vision*, 25(3):191–201, 1997.
- [PK92] E. Platen P. Kloeden. *Numerical solution of stochastic differential equations*. Springer-Verlag, 1992.
- [PP93a] R.K. Pina et R.C. Puetter. Bayesian image reconstruction: the pixon and optimal image modeling. In *P.A.S.P.*, volume 105, pages 630–637, 1993.
- [PP93b] R.C. Puetter et R.K. Pina. The pixon and bayesian reconstruction. In *SPIE*, volume 1946, pages 405–416, 1993.
- [RD92] S. Shlosman R.L. Dobrushin, R. Kotecký. *Wulff construction: a global shape from local interaction*. AMS translations series, vol 104, Providence R.I., 1992.
- [Rel02] G. Rellier. *Analyse de textures dans l'espace hyperspectral par des méthodes probabilistes*. Thèse de doctorat, Université de Nice-Sophia Antipolis, 2002.
- [RH99] H. Rue et M. Hurn. Bayesian object identification. *Biometrika*, 3:649–660, 1999.
- [Ris78] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.
- [RK77] B.D. Ripley et F.P. Kelly. Markov point processes. *Journal of the London Mathematical Society*, 15:188–192, 1977.

- [Rob96] C. Robert. *Méthodes de Monte Carlo par chaînes de Markov*. Economica, 1996.
- [SC00] A. Stassopoulou et T. Caelli. Building detection using bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(6):715–733, 2000.
- [Ser89] M.A. Serendero. *Extraction d'informations symboliques en imagerie SPOT : réseaux de communication et agglomérations*. Thèse de doctorat, Université de Nice Sophia Antipolis, 1989.
- [SG90] S.Peleg et G.Ron. Nonlinear multiresolution: A shape-from-shading example. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(12):1206–1210, 1990.
- [SGJM94] A. Srivastava, U. Grenander, G.R. Jensen, et M.I. Miller. Jump-diffusion markov processes on orthogonal groups for object recognition. *Special Issue of the Journal of Statistical Planning and Inference*, 1994.
- [SKM87] D. Stoyan, W.S. Kendall, et J. Mecke. *Stochastic Geometry and its Applications*. John Wiley and Sons, chichester, 1987.
- [SM94] M. Schmitt et J. Mattioli. *Morphologie Mathématique*. Masson, Paris, 1994.
- [SR92] M. Sigelle et R. Ronfard. Modèle de Potts et relaxation d'images de labels par champs de markov. *Traitement du Signal*, 6(9):449–458, 1992.
- [TB98] H. Tjelmeland et J. Besag. Markov random fields with higher-order interactions. *Scandinavian Journal of Statistics*, 25:415–433, 1998.
- [TMM<sup>+</sup>98] F. Tupin, H. Maître, J.F. Mangin, J.M. Nicolas, et E. Pechersky. Detection of linear features in SAR images: application to road network extraction. *IEEE Trans. on Geoscience and Remote Sensing*, 36(2):434–453, 1998.
- [UWS<sup>+</sup>97] J.K. Udupa, L. Wei, S. Samaraseka, Y. Miki, M.A. van Buchem, et R.I. Grossman. Multiple sclerosis lesion quantification using fuzzy-connectedness principles. *IEEE Trans. on Medical Imaging*, 16:598–608, 1997.
- [vL00] M.N.M. van Lieshout. *Markov Point Processes and Their Applications*. Imperial College Press, 2000.
- [vLS01] M.N.M. van Lieshout et R. Stoica. The candy model revisited: Markov properties and inference. Technical Report PNA-R0115, CWI, 2001.
- [WEMN92] K.J. Worsley, A.C. Evans, S. Marret, et P. Neelin. A three-dimensional statistical analysis for CBF activation studies in human brain. *Journal of Cerebral Blood Flow and Metabolism*, 12:900–918, 1992.
- [WF95] K.J. Worsley et K.J. Friston. Analysis of fMRI time-series revisited again. *NeuroImage*, 2:173–181, 1995.
- [WH97] R.C. Wilson et E.R. Hancock. Structural matching by discrete relaxation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(6):634–648, 1997.
- [Wit84] A.P. Witkin. Scale-space filtering: a new approach to multi-scale description. In *Image Understanding*, pages 79–95, 1984.
- [WMCL97] A. Winkler, H. Maître, N. Cambou, et E. Legrand. An original multi-sensor approach to scale-based image analysis for aerial and satellite images. In *ICIP*, pages 234–237, 1997.
- [WP85] G. Wolberg et T. Pavlidis. Restoration of binary images using stochastic relaxation with annealing. *Pattern Recognition Letters*, 3:375–388, 1985.
- [Zad65] L.A. Zadeh. Fuzzy sets. *Information and control*, 8:338–353, 1965.
- [ZC93] A. Zlotnick et P. Carnine. Finding road seeds in aerial images. *Computer Vision, Graphics, and Image Processing*, 57:243–260, 1993.
- [ZGO94] D. Zhang, L. Van Gool, et A. Oosterlinck. Coastline detection from SAR images. In *IGARSS'94*, volume IV, pages 2134–2136, 1994.





# Annexe : Curriculum Vitae détaillé

## Etat civil

Xavier Descombes  
 Né le 27 avril 1966 à Angers (Maine et Loire)  
 Nationalité Française  
 Crista Galli, B2  
 372, route de Saint Jean  
 06600 Antibes  
 Mariée, 2 enfants

## Etudes

Sept. 86-Juil. 89 Elève ingénieur à Télécom Paris - Option : Traitement des Images  
 Oct. 89 - Sept. 90 DEA de mathématiques pures à Paris VI (mention bien)  
 Oct. 90 - Déc. 93 Thèse de doctorat à Télécom Paris  
 (obtenue avec les félicitations du Jury)

## Expérience professionnelle

Oct. 89 - Sept. 90 Scientifique du contingent : développement d'algorithmes de restauration d'images.  
 Janv. 94 - Déc. 94 Assistant de recherche à Télécom Paris  
 Janv. 95 - Déc. 95 "Free research fellow" à l'Université Catholique de Louvain (KUL-Belgique)  
 Janv. 96 - Sept. 96 Post-doctorant à l'INRIA Sophia Antipolis (projet Patis)  
 Oct. 96 - Oct. 97 "Visiting Scientist" à l'Institut Max Planck pour les neurosciences de Leipzig (Allemagne)  
 Depuis Nov. 97 Chargée de recherche à l'INRIA Sophia Antipolis (projet Ariana- CR1 depuis janv. 2000)

## Enseignement

97,98,99	20h/an	Master UNAM-Mexico	Champs de Markov en analyse d'image
depuis 98	9h/an	DEA Astrophys. de l'UNSA	Télé-détection et cartographie
depuis 00	15h/an	5ème année de l'ESINSA	Analyse d'image
depuis 00	17h/an	3ème année de Supaero	Filtrage et Segmentation
2002	17h	DEA Univ. Antilles-Guyanne	Traitement d'image et télé-détection
depuis 03	15h/an	DESS Université Paul Sabatier	Analyse d'image

## Collaborations

### Industrielles

- **Morpho-Système** - *Binarisation d'empreintes digitales par champs de Markov*, contrat avec Teledcom Paris, 8 mois à temps plein en 94.
- **Elf Aquitaine** - *Détection des failles dans les images sismiques*, contrat avec le CNRS, 2 mois à temps plein en 94.

- **CNES** - *Extraction des zones urbaines à partir de simulations SPOT V*, thèse d'A. Lorette (97/99).
- **Alcatel CRC** - *Recalage de réseaux routiers carte/image en vue de la mise à jour*, stage de DEA de C. Hivernat (98).
- **Astrium** - *Classification d'images satellitaires hyperspectrales en milieu rural et péri-urbain*, stage de DEA de O. Pony (00).
- **Alcatel Space** - *Classification de milieux urbains et péri-urbains à partir de données hyperspectrales*, thèse de G. Rellier (bourse région PACA/Alcatel 00/02).
- **BRGM/DGA** - *MAJOR : utilisation des différents capteurs satellitaires pour la mise à jour des indicateurs liés aux problèmes de ressources hydrogéologiques et minières en Afrique*, rapport d'étude commandé par la DGA, collaboration avec le BRGM (00/01).
- **IGN** - *Extraction de bâtiments par processus ponctuels marqués à partir d'un couple stéréo*, stage de DEA de L. Garcin (00).
- **IGN** - *Segmentation d'images d'observation de la Terre par des techniques de géométrie stochastique*, thèse de S. Drot (bourse CIFRE 00/02).
- **BRGM** - *Extraction des réseaux linéiques en télédétection par processus Markov objets*, thèse de C. Lacoste (02/04, bourse Centre INRIA/BRGM).
- **DGA/CNRS** - *Extraction de caricatures de bâtiments sur des Modèles Numériques d'Élévation par processus ponctuels marqués*, thèse de M. Ortner (02/04, bourse DGA/CNRS).
- **Cemagref** - *Reconnaissance assistée par ordinateur d'arbres et de forêts sur images numériques à haute ou très haute résolution*, stage de DEA de G. Perrin dans le cadre d'une action COLORS03.

#### *Académiques*

- **IPIT** - *Physique Statistique Stable et Instable en Traitement d'Image*, depuis 1998, collaboration avec l'IPIT de l'Académie des Sciences de Russie dans le cadre d'un projet conjoint soutenu par l'**Institut Lyapunov**.
- **IRIT** - *Recuit simulé pour le "Shape from Shading"*, responsable d'un projet "jeunes chercheurs" du **CNRS** en 2001/2002 menée en collaboration avec l'équipe TCI de l'IRIT.
- **CWI** - *Processus ponctuels en analyse d'image*, collaboration soutenue par **ERCIM** (01/02).
- **MPI** - *Processus Ponctuels Marqués pour la détection de lésions cérébrales à partir d'IRM*, 2 mois en "sabbatique" à l'**Institut Max Planck** pour les Neurosciences de Leipzig en 2001.
- **IRIT, Univ. de Rome** - *Shape from Shading*, projet **PAI Galilée** (acceptée fin 2003).

#### **Activités d'encadrement ou de co-encadrement :**

##### *Stages d'ingénieur*

- **J.C. Fondeur** - *Segmentation par Fusion de Données*, Telecom Paris, 1993.
- **M. Imberty** - *Simulation de processus objets et application à la segmentation d'image*, Ecole Polytechnique, 1999.
- **S. Pellise** - *Recalage d'images satellitaires optiques et radar RSO*, Ecole Polytechnique, 1999.
- **F. Antoine** - *Extraction de Routes par Analyse Multi-Echelle*, Ecole Polytechnique, 2000

##### *Stages de DEA*

- **S. Drot** - *Modélisation probabiliste de la perception de la régularité dans une forme*, DEA de Mathématiques, Univ. de Metz, 1996.
- **C. Hivernat** - *Extraction, recalage et mise en correspondance de linéiques à partir d'un couple carte/image*, DEA SIC, Univ. Paul Sabatier, Toulouse, 1998.

- **G. Rellier** - *Recalage et déformation d'un réseau routier sur une image SPOT*, DEA STIR, Univ. Brest, 1999.
- **L. Garcin** - *Détection de bâtiments par processus Markov objets*, DEA MVA, ENS Cachan, 2000.
- **M. Rzepka** - *Etude d'un processus ponctuel de Markov en vue de détecter des réseaux routiers*, DEA de Probabilités et Applications, Paris VI, 2000.
- **O. Pony** - *Classification d'images satellitaires hyperspectrales en zone rurale et périurbaine : méthodes de régularisation*, DEA MVA, ENS Cachan, 2000
- **M. Ortner** - *Extraction de caricatures de bâtiments sur des modèles numériques d'élévation*, DEA Mathématiques appliquées, Univ. Paul Sabatier, Toulouse, 2001.
- **F. Cerdat** - *Rendu de scène urbaine à partir d'une description par objets*, DEA Image, Univ. Jean Monnet, Saint-Etienne, 2002
- **G. Perrin** - *Etude du couvert forestier à partir d'un processus objet*, DEA MVA, ENS Cachan, 2003.

#### Thèses

- **A. Lorette** - *Analyse de texture par méthodes markoviennes et par morphologie mathématique : application à l'analyse des zones urbaines sur des images satellitaires*, soutenue le 28 septembre 1999.
- **R. Stoica** - *Processus ponctuels pour l'extraction de réseaux linéiques dans les images satellitaires et aériennes*, soutenue le 9 février 2001.
- **G. Rellier** - *Analyse de textures dans l'espace hyperspectral par des méthodes probabilistes*, soutenue le 8 novembre 2002.
- **S. Drot** - *Segmentation d'images d'observation de la Terre par des techniques de géométrie stochastique*, soutenue le 5 décembre 2002.
- **O. Viveros-Cancino** - *Analyse des zones urbaines par fusion de données en télédétection*, soutenue le 10 juin 2003.
- **M. Ortner** - *Extraction de caricatures de bâtiments sur des Modèles Numériques d'Elévation par processus ponctuels marqués*, soutenance prévue en septembre 2004.
- **C. Lacoste** - *Extraction des réseaux linéiques en télédétection par processus Markov objets*, soutenance prévue en septembre 2004.
- **G. Perrin** - *Etude du couvert forestier à partir de processus ponctuels marqués*, soutenance prévue en octobre 2006.

#### Autres activités

- Responsable du groupe TF4 (*Fusion d'images*) du projet du GdR ISIS "Radar à Synthèse d'Ouverture", soutenu par le PNTS (99-00)
- Relecteurs pour différentes revues (IEEE Trans. on PAMI, IP, Medical Imaging, IJCV,...) et conférences (ICIP, ICASSP, Orasis,...)
- Membre du comité de programme de FSPIPA'99, Budapest, juillet 1999.
- Co-organisateur de la conférence **IPRM** (*Image Processing and Related Mathematical fields*), ayant eu lieu dans les locaux de l'Institut Lyapunov à Moscou en juillet 2002.
- Membre du comité de lecture de la revue *Traitement du Signal*.
- Membre des Jury de thèse de A. Lorette (UNSA), M. Fayolle (LSS), R. Stoica (UNSA), G. Rellier (UNSA), S. Drot (UNSA), O. Viveros-Cancino (UNSA), D. Petit (UPS).

- Membre de la commission d'évaluation COLOR (*COopérations L Ocales de Recherche*) de l'INRIA Sophia-Antipolis de 00 à 02.
- Expert pour le "special fund for research" de l'université d'Anvers, Belgique, 2001.
- Expert pour l'ACI "masses de données", 2003.
- Responsable d'une action COLOR 2003, en collaboration avec le Cemagref.

**Publications**

Chapitres de livres :	2
Revue internationale avec comité de lecture :	15
Revue Nationale avec comité de lecture :	6
Conférences internationales :	29
Conférences nationales :	13

## Liste des Publications

- [1] L. Aurdal, X. Descombes, H. Maître, I. Bloch, C. Adamsbaum, et G. Kalifa. Fully automated analysis of adrenoleukodystrophy from dual echo MR-images. In *CAR'95*, pages 35–40, 1995. Berlin.
- [2] N. Baghdadi, A. Bourguignon, X. Descombes, C. Parent, J.F. Desprat, et C. King. MAJOR : utilisation des différents capteurs satellitaires pour la mise à jour des indicateurs liés aux problèmes de ressources hydrogéologiques et minières en Afrique. Rapport Technique RP-51299, BRGM, 2001.
- [3] F. Cerdat, X. Descombes, et J. Zerubia. Urban scene rendering using object description. In *IGARSS'03*, 2003. Toulouse.
- [4] A. Cruzil, X. Descombes, et J.D. Durou. Multiresolution approach for shape from shading coupling deterministic and stochastic optimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(11):1416–1421, 2003.
- [5] X. Descombes. Modélisation de la vision humaine. ENST- Dpt Image, 1989. Mémoire de fin d'étude.
- [6] X. Descombes. Diffusion, processus de Markov et Martingales. Master's thesis, DEA de Mathématiques Pures - Paris VI, 1990.
- [7] X. Descombes. *Champs markoviens en analyse d'images*. Thèse de doctorat, Telecom Paris (signal et images), 1993.
- [8] X. Descombes. A fission and fusion markovian approach for multi-channel segmentation. In *IGARSS'95*, pages 124–127, 1995. Firenze.
- [9] X. Descombes. Application of stochastic techniques in image processing for automatic tissue classification in MRI and blood vessel restoration in MRA. Technical Report KUL/ESAT/MI2/9603, Laboratory for Medical Imaging Research (ESAT-Radiology), K.U.Leuven, Belgium, 1996.
- [10] X. Descombes. A dense class of Markov random fields and associated parameter estimation. *Journal of Visual Communication and Image Representation*, 8(3):299–316, 1997.
- [11] X. Descombes, S. Drot, M. Imbert, H. Le Men, et J. Zerubia. Segmentation d'image haute résolution par processus Markov objet. Cemagref Edition, 2001. Séminaire Télédétection à très haute résolution spatiale et analyse d'image. Montpellier, 14 déc. 1999.
- [12] X. Descombes, J.D. Durou, et D. Petit. Recuit simulé pour le "shape from shading". In *GRETSI'01*, 2001. Toulouse.
- [13] X. Descombes et Y. Goussard. *Approche bayésienne pour les problèmes inverses*, chapitre "Problèmes non supervisés". Number 8 in *Traité IC2*. Hermès Science Publications, 2001.
- [14] X. Descombes, C. Hivernat, S. Randriamasy, et J. Zerubia. Graph-matching model using gibbsian modeling: application to map-SPOT image roads network. In *SPIE conference on Mathematical Modeling, Bayesian Estimation and Inverse Problems*, volume 3816, pages 2–10, 1999. Denver.
- [15] X. Descombes et F. Kruggel. A Markov Pixon Information approach for low level image description. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(6):482–494, 1999.

- [16] X. Descombes, F. Kruggel, et Y. von Cramon. fMRI signal restoration using an edge preserving spatio-temporal Markov random field. *NeuroImage*, 8:340–349, 1998.
- [17] X. Descombes, F. Kruggel, et Y. von Cramon. Spatio-temporal fMRI analysis using Markov random fields. *IEEE Trans. on Medical Imaging*, 17(6):1028–1039, 1998.
- [18] X. Descombes, F. Kruggel, G. Wollny, et H.J. Gertz. An object based approach for detecting small brain lesions: application to Virchow-Robin spaces. *IEEE Trans. on Medical Imaging*, to appear.
- [19] X. Descombes, JF. Mangin, E. Pechersky, et M. Sigelle. Fine structure preserving Markov model for image processing. In *SCIA'95*, pages 349–356, 1995. Uppsala.
- [20] X. Descombes, M. Moctezuma, H. Maître, et J.P. Rudant. Coastline detection by a markovian segmentation of SAR images. *Signal Processing*, 55:123–132, 1996.
- [21] X. Descombes, R. Morris, et J. Zerubia. Quelques améliorations la segmentation d'images bayésienne. première partie: modélisation. *Traitement du Signal*, 14(4):373–382, 1997.
- [22] X. Descombes, R. Morris, et J. Zerubia. Quelques améliorations la segmentation d'images bayésienne. deuxième partie: classification. *Traitement du Signal*, 14(4):383–393, 1997.
- [23] X. Descombes, R. Morris, J. Zerubia, et M. Berthod. Maximum likelihood estimation of markovian prior parameters using Markov chain Monte Carlo. In *EMMCVPR*, 1997. Venice.
- [24] X. Descombes, R. Morris, J. Zerubia, et M. Berthod. Estimation of markov random field prior parameters using markov chain monte carlo maximum likelihood. *IEEE Trans. on Image Processing*, 8(7):954–963, 1999.
- [25] X. Descombes et E. Pechersky. Isotropic properties of some multi-body interaction models. In D. Chetverikov and T. Sziranyi, editors, *FSPIPA*, pages 133–144. Osterreichische Computer Gesellschaft, 1999. Budapest.
- [26] X. Descombes et E. Pechersky. Metropolis vs Kawasaki dynamic for image segmentation based on gibbs models. In E.R. Hancock et M. Pelillo, editors, *EMMCVPR*, Lecture Note in Computer Science 1654, pages 99–114. Springer, 1999. York.
- [27] X. Descombes et E. Pechersky. Droplet shapes for a class of models in  $Z_2$  at zero temperature. *Journal of Statistical Physics*, 111(1):129–169, 2003.
- [28] X. Descombes et F. Preteux. Potentiels canoniques et modèles markoviens. In *GRETSI'91*, volume II, pages 821–824, 1991. Juan-les-pins.
- [29] X. Descombes et F. Preteux. Topology and parameter estimation in MRF modeling. In *SPIE, Neural and Stochastic Methods in Image and Signal Processing II*, volume 2032, 1993. San Diego.
- [30] X. Descombes et F. Preteux. Les phi-modèles : caractérisation de processus markoviens gaussiens perturbés. In *AFCEC-RFIA'94*, pages 75–86, 1994. Paris.
- [31] X. Descombes, F. Preteux, et M. Sigelle. Modèles markoviens en analyse d'images : définition des potentiels canoniques, des cliques et estimation des paramètres. Journées thématiques du GDR/CNRS, 1991.
- [32] X. Descombes et J.M. Salvador. Extraction et classification des réseaux de craquelures dans les oeuvres picturales. ENST- Dpt Image, 1989. Mémoire de Stage.
- [33] X. Descombes, M. Sigelle, et F. Preteux. Application de la renormalisation à l'analyse de textures markoviennes gaussiennes. In *GRETSI'93*, volume I, pages 21–25, 1993. Juan-les-pins.
- [34] X. Descombes, M. Sigelle, et F. Preteux. GMRF parameter estimation in a non-stationary framework by a renormalization technique: Application to remote sensing imaging. *IEEE Trans. on Image Processing*, 8(4):490–503, 1999.
- [35] X. Descombes, R. Stoica, L. Garcin, et J. Zerubia. A RJMCMC algorithm for object processes in image processing. *Monte Carlo Methods and Applications*, 7(1-2):149–156, 2001.

- [36] X. Descombes, R. Stoica, et J. Zerubia. Two Markov point processes for simulating line networks. In *ICIP'99*, 1999. Kobe.
- [37] X. Descombes, M.N.M. van Lieshout, R. Stoica, et J. Zerubia. Parameter estimation by a Markov Chain Monte Carlo technique for the Candy-model. In *IEEE Workshop on Statistical Signal Processing*, pages 22–25, 2001. Singapore.
- [38] X. Descombes et J. Zerubia. Marked point processes in image analysis. *IEEE Signal Processing Magazine*, vol. 19, no. 5, pp.77-84, 2002.
- [39] X. Descombes et J. Zerubia. Marked point processes in image analysis. *ERCIM News*, No. 50, pp. 24-25, 2002.
- [40] S. Drot, X. Descombes, H. Le Men, et J. Zerubia. Object point processes for image segmentation. In *ICPR02*, 2002. Quebec City.
- [41] S. Drot, X. Descombes, H. Le Men, et J. Zerubia. Remotely sensed image segmentation using an object point process. In *IGARSS'03*, 2003. Toulouse.
- [42] L. Garcin, X. Descombes, J. Zerubia, et H. Le Men. Building extraction using a Markov point process. In *ICIP'01*, 2001. Thessalonik.
- [43] C. Hivernat, X. Descombes, S. Randriamasy, et J. Zerubia. Mise en correspondance et recalage de graphes : application aux réseaux routiers extraits d'un couple carte/image. *Traitement du Signal*, 17(1):21–32, 2000.
- [44] C. Hivernat, S. Randriamasy, X. Descombes, et J. Zerubia. Qualification automatique des résultats d'une mise en correspondance de réseaux routiers en vue de la mise à jour cartographique. In *ISPRS Working Group II/6 Workshop on: "3D Geospatial Data Production: Meeting Application Requirements"*, 153, pages 91–93. Bulletin de la SFPT, 1999. Paris.
- [45] F. Kruggel, C. Chalopin, X. Descombes, et V. Kovalev. Segmentation of pathological features in MRI brain datasets. In IEEE Press, editor, *ICONIP'02*, pages 2673–2677, 2002. Piscataway, NJ.
- [46] F. Kruggel, X. Descombes, et Y. von Cramon. Die vorerarbeitung von fMRI-daten. In *Bildverarbeitung fr die Medizin*, 1998. Universittsklinikum der RWTH Aachen.
- [47] F. Kruggel, X. Descombes, et Y. von Cramon. Preprocessing of fMR datasets. In *Workshop on Biomedical Image Analysis*, pages 211–220. IEEE Computer Press, 1998. Los Alamitos.
- [48] F. Kruggel, Y. von Cramon, et X. Descombes. Comparison of filtering methods for fMRI datasets. *NeuroImage*, 10:530–543, 1999.
- [49] C. Lacoste, X. Descombes, et J. Zerubia. Road network extraction in remote sensing by markov object processes. In *ICIP'03*, 2003. Barcelona.
- [50] C. Lacoste, X. Descombes, J. Zerubia, et N. Baghdadi. Extraction automatique des réseaux linéiques à partir d'images satellitaires et aériennes par processus Markov objet. *Bulletin de la SFPT*, 170(2):13–22, 2003.
- [51] C. Lacoste, X. Descombes, J. Zerubia, et N. Baghdadi. Extraction de réseaux linéiques à partir d'images satellitaires par processus Markov objet. In *GRETSI'03*, 2003. Paris.
- [52] F.M. Letourneau, A. Lorette, J.P. Rudant, et X. Descombes. Utilisation d'une séquence d'images ERS et d'une méthode d'extraction automatique des zones urbaines pour le suivi de l'extension de la ville de Macapa (Amapa, Brésil). *Bulletin de la SFPT*, 156:2–11, 1999.
- [53] A. Lorette, X. Descombes, et J. Zerubia. Texture analysis through Markov random fields: Urban areas extractions. In *ICIP'99*, 1999. Kobe.
- [54] A. Lorette, X. Descombes, et J. Zerubia. Fully unsupervised fuzzy clustering with entropy criterion. In *ICPR'00*, volume III, pages 998–1001, 2000. Barcelone.
- [55] A. Lorette, X. Descombes, et J. Zerubia. Modélisation markovienne multi-directionnelle: Application à l'extraction des zones urbaines. In *RFIA'00*, volume III, pages 17–26, 2000. Paris.



- [56] A. Lorette, X. Descombes, et J. Zerubia. Texture analysis through a markovian modelling and fuzzy classification: Application to urban area extraction from satellite images. *Int. Journal on Computer Vision*, 36(3):219–234, 2000.
- [57] R. Morris, X. Descombes, et J. Zerubia. Fully bayesian image segmentation - an engineering perspective. In *ICIP'97*, 1997.
- [58] R.D. Morris, X. Descombes, et J. Zerubia. The Ising/Potts model is not well suited to segmentation tasks. In *IEEE Digital Signal Processing Workshop*, 1996. Norway.
- [59] M. Ortner, X. Descombes, et J. Zerubia. Building extraction from digital elevation models. In *ICASSP'03*, volume III, 2003. Hong-Kong.
- [60] M. Ortner, X. Descombes, et J. Zerubia. Un nouveau modèle pour l'extraction de caricatures de bâtiments sur des modèles numériques d'élévation. In *TAIMA'03*, 2003. Hammamet.
- [61] G. Palubinskas, X. Descombes, et F. Kruggel. An unsupervised clustering method using the entropy minimization. In *ICPR'98*, 1998. Brisbane.
- [62] O. Pony, U. Polverini, L. Gautret, J. Zerubia, et X. Descombes. Classification d'image satellitaire superspectrale en zone rurale et périurbaine. In *GRETSI'01*, 2001. Toulouse.
- [63] F. Preteux et X. Descombes. Synthèse et analyse de textures par coopération de processus multi-échelles. In *AFCEP-RFIA'91*, volume II, pages 1015–1026, 1991. Lyon-Villeurbanne.
- [64] G. Rellier, X. Descombes, F. Falzon, et J. Zerubia. Classification de textures hyperspectrales fondée sur un modèle markovien et une technique de poursuite de projection. *Traitement du Signal*, 20(1):25–42, 2003.
- [65] G. Rellier, X. Descombes, et J. Zerubia. Deformation of a cartographic road network on a SPOT satellite image. In *ICIP'00*, volume II, pages 736–739, 2000. Vancouver.
- [66] G. Rellier, X. Descombes, et J. Zerubia. Local registration and deformation of a road cartographic database on a SPOT satellite image. *Pattern Recognition*, 35:2213–2221, 2002.
- [67] G. Rellier, X. Descombes, J. Zerubia, et F. Falzon. La poursuite de projection pour la classification d'images hyperspectrales texturées. In *ORASIS'01*, pages 435–444, 2001. Cahors.
- [68] G. Rellier, X. Descombes, J. Zerubia, et F. Falzon. Un modèle markovien gaussien pour l'analyse de texture hyperspectrale en milieu urbain. In *GRETSI'01*, 2001. Toulouse.
- [69] G. Rellier, X. Descombes, J. Zerubia, et F. Falzon. A Gauss-Markov model for hyperspectral texture analysis of urban areas. In *ICPRO2*, 2002. Quebec City.
- [70] R. Stoica, X. Descombes, et J. Zerubia. Road extraction in remote sensed images using a stochastic geometry framework. In *MaxEnt'00*, 2000. Gif Sur Yvette.
- [71] R. Stoica, X. Descombes, et J. Zerubia. A Gibbs point process for road extraction from remotely sensed images. *Int. Journal on Computer Vision*, to appear.
- [72] R.S. Stoica, M.N.M. van Lieshout, X. Descombes, et J. Zerubia. *Spatial Statistics through Applications*, chapter “An application of marked point processes to the extraction of linear networks from images”. WIT Press, 2002.
- [73] F. Tupin, E. Trouvée, X. Descombes, J.M. Nicolas, et H. Maître. Improving IFSAR phase unwrapping by early detection of non interferometric features. In Giorgio Franceschetti; Christopher J. Oliver; Franco S. Rubertone; Shahram Tajbakhsh, editor, *SPIE, Microwave Sensing and Synthetic Aperture Radar*, volume 2958, pages 250–261, 1996.
- [74] D. Vandermeulen, X. Descombes, P. Suetens, et G. Marchal. Unsupervised regularized classification of multi-spectral MRI. In K.H.Hohne and R.Kikinis, editors, *VBC'96*. Springer Verlag, 1996.
- [75] O. Viveros, X. Descombes, et J. Zerubia. Apport de l'imagerie radar pour l'extraction des zones urbaines. In *ORASIS'01*, pages 405–414, 2001. Cahors.

- [76] O. Viveros-Cancino, X. Descombes, N. Baghdadi, et J. Zerubia. Fusion of radiometry and textural information for SIRC image classification. In *ICIP02*, 2002. Rochester.