DEA Mathématiques appliquées

Extraction de caricatures de bâtiments sur des modèles numériques d'élévation

Mathias Ortner
Encadrement : Xavier Descombes et Josiane Zerubia
Projet Ariana - CNRS/INRIA/UNSA

Août 2001

Résumé

L'objectif de ce travail est de détecter des bâtiments sur des Modèles Numériques d'Elévation (MNE).

Pour ce faire, nous introduisons un processus ponctuel dont les points représentent les bâtiments. La densité de ce processus ponctuel se divise en deux parties :

- la première est un modèle a priori utilisant des interactions entre les points pour introduire la connaissance que l'on a de la structure des batiments en zones urbaines,
- la seconde est un terme d'attache aux données pour assurer la cohérence entre les réalisation du processus ponctuel et le Modèle Numérique d'Elévation.

Nous calculons ensuite une estimée de la zone urbaine à partir de cette densité. Nous étudions deux estimateurs : le Maximum de Vraisemblance et un autre estimateur proposé par Rue qui utilise la Δ-distance de Baddeley.

Ces estimées sont calculées en utilisant une simulation de Monte Carlo par Chaine de Markov et en particulier un algorithme de Metropolis Hastings Green. Nous testons cette méthode sur des images de synthèse.

Mots-clés : Processus ponctuels, RJMCMC, extraction de bâtiments, modèle numérique d'élévation.

Abstract

This work aims at detecting buildings on Digital Elevation Models (DEM).
To achieve this goal, we define a point process whose points represent buildings. We split the density of this point process in two parts :

- the first one consists of a prior model using interactions between points to model the knowlege we have of urban areas,
- the second one is a data model to enforce coherence with respect to the Digital Elevation Model.

Thus we have a posterior distibution. We study two estimators : the first one is the Maximum Likehood while the second one has been proposed by Rue using Baddeley's Δ-distance. To caculate these estimators we use a Metropolis-Hastings-Green algorithm. We test this method on synthetic images.

Key-words : Point process, RJMCMC, building extraction, Digital Elevation Model.

Table des matières

Table des figures

Introduction

Cadre de travail

Ce travail de DEA s'est déroulé à l'Institut National de Recherche en Informatique et en Automatique (INRIA) et plus précisément dans le projet Ariana, projet commun CNRS/INRIA/UNSA.

Ce projet a pour centre d'intérêt les problèmes inverses en observation de la Terre et cartographie.

Ce travail de DEA s'inscrit donc un cadre scientifique très particulier mêlant traitement de l'image, informatique et mathématiques.

Le sujet de ce travail va être prolongé en un sujet de thèse avec un soutient de la DGA : ce mémoire sert donc de conclusion au DEA et d'introduction aux travaux futurs.

Objectifs et intérêts pratiques

L'objectif pratique de ce sujet est de développer une méthodologie et des algorithmes permettant de construire des plans de villes en trois dimension à partir d'images satellitaires.

Les applications d'un tel travail sont multiples : de la téléphonie mobile à l'aménagement urbain en passant par la cartographie, nombreuses sont les applications potentielles.

L'intitulé du sujet de mémoire était "extraction de caricatures de bâtiments sur des modèles numériques d'élévation".

L'objectif est donc d'obtenir une carte tridimensionnelle simplifiée de zones urbaines à partir d'un type de données que l'on va maintenant présenter.

Les données

Les données sont issues d'un traitement stéréoscopique. En utilisant certaines techniques connues en traitement de l'image on peut à partir de deux images satellitaires ou aériennes d'une même scène en proposer une reconstruction en trois dimensions.

Dans notre cas, les scènes d'intérêt sont des zones urbaines. Cette reconstruction se présente sous la forme d'une grille représentant le domaine terrestre d'intérêt munie pour chacun de ses points d'une estimation de la hauteur du relief. (La notion de relief est définie par les photographies aériennes ou satellitaires utilisées pour cette reconstruction : ainsi, les arbres, les maisons , les voitures et les lampadaires font partie du relief.) On appelle une telle reconstruction un **modèle numérique d'élévation (MNE)**. Lorsque la scène d'intérêt est un terrain géologique on appelle une telle reconstruction **modèle numérique de terrain (MNT).**

La figure 1 présente un exemple de paire d'images qui peut être utilisée pour un traitement

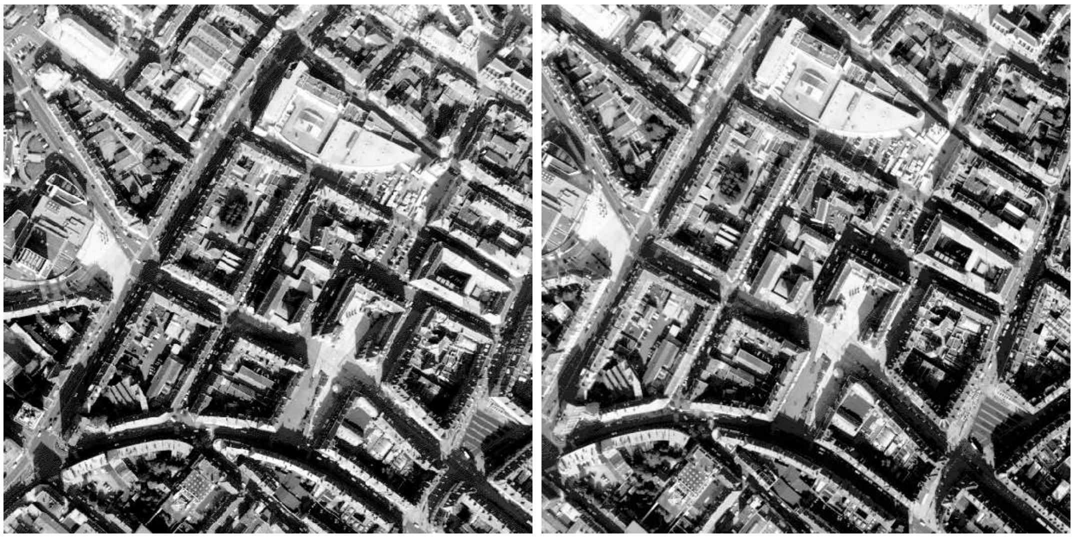

FIG. 1 – *Exemple de paire d'images utilisée en stéréovision pour la construction d'un modèle numérique d'élévation. Il s'agit d'une partie de la ville d'Amiens. ©IGN*

stéréoscopique. La figure 2 montre une partie du modèle numérique d'élévation obtenu par un tel traitement.

Problématique du sujet

Les zones urbaines présentent un intérêt particulier en traitement de l'image, mais nécessitent des images très haute résolution (THR). Ces images peuvent être soit aériennes, soit satellitaires (le satellite commercial IKONOS par exemple fournit des images à 1m de résolution et le futur capteur français des Pléiades devrait fournir des images à 70cm de résolution.)

Les zones urbaines sont des zones ou les objets d'intérêts ne sont plus des surfaces, mais des objets tri-dimensionnels. Ce sont également des zones où les objets sont denses. Ces objets sont de taille réduite, surtout si l'on regarde leur trace au sol.

Enfin les zones urbaines présentent des objets cachés et parfois transparents.

Ces raisons font que les zones urbaines proposent beaucoup de problèmes ouverts en traitement d"image.

Notre problème est donc un problème inverse dont les objets d'intérêt sont particulièrement mal adaptés aux deux dimensions d'une image et aux approches pixeliques.

C'est pourquoi il paraît pertinent de s'affranchir le plus possible de la notion de pixel et d'image 2D pour résoudre ce problème.

Les approches usuelles

On trouvera un récapitulatif des différentes approches classiques pour résoudre ce problème dans [17],[18],[19] et [23]. Ces approches classiques consistent, pour l'essentiel, en des méthodes de segmentation et de fusion de régions.

On distingue toutefois les travaux de M. Roux et H. Maître (voir [7, 8, 13]) qui reconstruisent

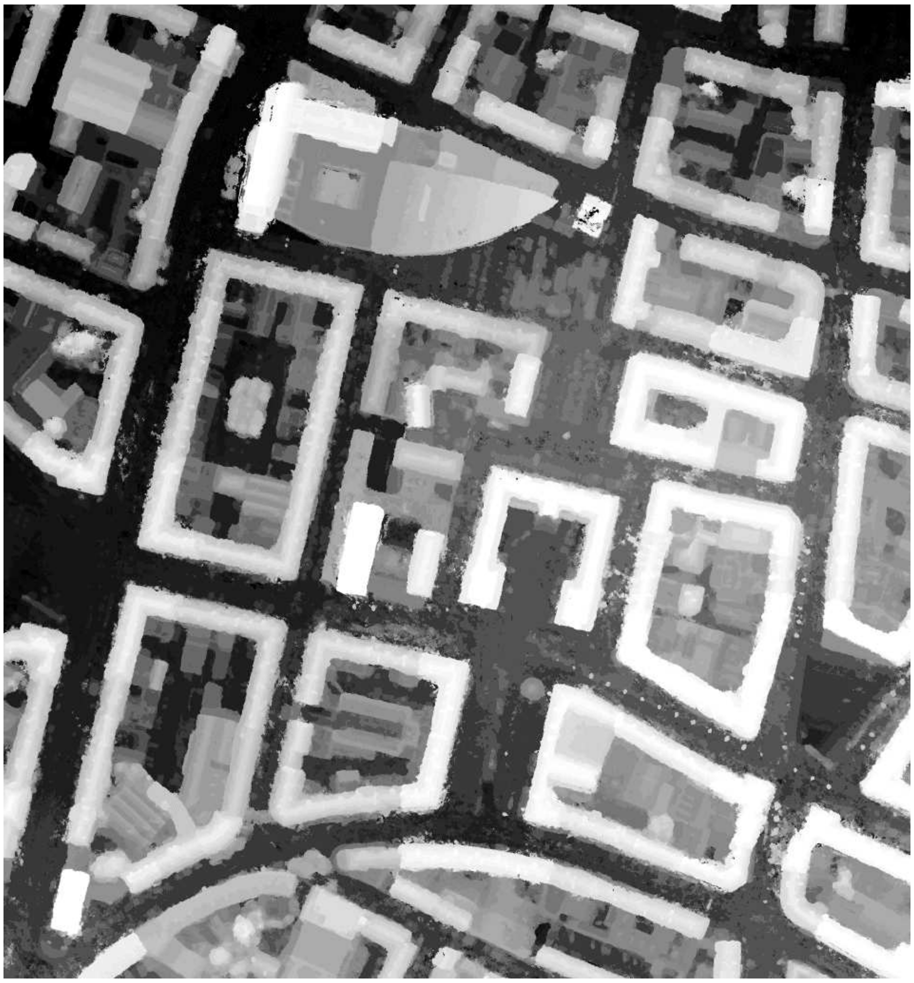

FIG. 2 – *Exemple de modèle numérique d'élévation sur la ville d'Amiens.* ©*IGN*

la scène étape par étape :

1. D'abord, d'une part il font de la stéréovision à n vues et d'autre part de la segmentation d'image par régions. Cela leur donne donc une image de disparité et des cartes de segmentation.
2. Ensuite, ils utilisent ces informations pour construire un Modèle Numérique d'Élévation dense, en vérifiant que les régions segmentées sont planes et en imposant des contraintes géométriques sur le modèle résultat.
   Cette deuxième étape est faite de manière itérative et tient compte de plusieurs facteurs : homogénéité des couleurs, critères géométriques etc...

Notre approche

L'approche que nous avons choisie d'adopter utilise des processus ponctuels. Leur utilisation en traitement de l'image a été proposée par A. J. Baddeley et M.N.M. van Lieshout dans [2] sur un cas d'école très simple. Notre objectif est de pouvoir traiter des images réelles.

Ces processus ponctuels semblent être particulièrement adaptés à notre problème puisque:

– Les objets mis en jeux par ces processus peuvent facilement être nos objets d'intérêt, c'est à dire les batiments.
– L'approche est sur-pixelique : on se détache totalement de l'image pour se rapprocher du modèle de réalité physique.
– Les méthodes mises en jeu sont stochastiques, ce qui donne une panoplie d'outils classiques dans les problèmes inverses.

Chapitre 1

Fondements des processus ponctuels.

Ce chapître a pour but de présenter quelques éléments de la théorie des processus ponctuels, ainsi que quelques exemples illustratifs. Pour plus de détails théoriques, on pourra se référer à [22] ou à [37] d'où sont tirés l'essentiels des éléments présentés ici.

1.1 Premières définitions et notations.

1.1.1 Le contexte mathématique.

On s'intéresse à des ensembles non ordonnés de points d'un espace donné χ, par exemple $\mathbb{R}^2$.

Définition 1 *On appelle* **configuration** *un ensemble dénombrable, non ordonné de points de χ:*

$$\mathbf{x} = \{x_1, \ldots, x_n, \ldots\}$$

Cette première définition peut poser des problèmes : en effet un point peut être répété dans une configuration. Par exemple :

$$\{1,3,4,3\} \qquad \text{et} \qquad \{\pi,3.5,\pi,\pi\}$$

sont des configurations dans $\mathbb{R}$.

On veut considérer des configurations aléatoires. Pour ce faire, il faut ajouter des contraintes sur χ. On choisit de considérer un espace χ muni d'une métrique d tel que (χ,d) soit complet et séparable. Cette métrique permet de définir une topologie et une σ-algèbre borélienne.

En pratique, χ sera souvent un sous-ensemble compact de $\mathbb{R}^d$ muni de la distance euclidienne.

Définition 2 *On dit d'une configuration $\mathbf{x} \subseteq \chi$ qu'elle est* **localement finie** *si elle place un nombre fini de points dans tout borelien borné $A \subseteq \chi$.*

La famille de toutes les configurations localement finies sera notée N^{lf}.

On définit alors la notion de processus ponctuel :

Définition 3 *Un* **processus ponctuel sur** χ *est une application X d'un espace probabilisé $(\Omega,\mathcal{A},\mathbb{P})$ dans N^{lf}, tel que pour tout borélien $A \subseteq \chi$, le nombre $N(A) = N_X(A)$ de points dans A soit une variable aléatoire (finie).*

Ce qui induit une nouvelle définition:

Définition 3bis *Si l'espace χ est borné ou si $N_X(\chi)$ est fini presque surement, le processus ponctuel est dit* **processus ponctuel fini**.

Les réalisations d'un processus ponctuel X sont donc des configurations aléatoires de points tels 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 $(N^{lf}, \mathcal{N}^{lf})$, où $\mathcal{N}^{lf}$ est la plus petite σ-algèbre telle que pour tout borélien borné $A \subseteq \chi$ l'application $\mathbf{x} \to N_{\mathbf{x}}(A)$ soit mesurable.

La mesure de probabilité induite sur $\mathcal{N}^{lf}$ est appelée la **loi** de X .

1.1.2 Loi d'un processus ponctuel.

La loi d'un processus ponctuel X devrait être la mesure image par l'application X de $\mathbb{P}$ sur $\mathcal{N}^{lf}$.

Mais comme $\mathcal{N}^{lf}$ est définie par la mesurablité des applications $\mathbf{x} \to N_{\mathbf{x}}(A)$ pour des boréliens $A \subseteq \chi$, 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 $(N(A_1),\ldots,N(A_m))$ où les A_i sont des boréliens bornés :

Définition 4 *La famille des* **lois en dimensions finie (fidis)** *d'un processus ponctuel X sur un espace métrique (χ,d) complet et séparable est la collection des lois jointes de $(N(A_1),\ldots,N(A_m))$ pour tout vecteur fini $(A_1,\ldots,A_m)$ de boréliens bornés $A_i \subseteq \chi,\quad i = 1,\ldots,m$ de longueur quelconque $m \in \mathbb{N}$.*

L'intérêt de cette définition est justifié par le théorème suivant (voir [6]) :

Théorème 2 *La loi d'un processus ponctuel X sur un espace métrique complet et séparable (χ,d) est entièrement déterminée par ses fidis.*

Donc, deux processus ponctuels partageant les mêmes fidis ont même loi.

1.1.3 Processus ponctuels marqués.

Ce type de processus ponctuels a été décrit par Daley dans [6]

Définition 5 *Soit (χ,d) et $(\mathcal{K},d')$ deux espaces métriques, complets et séparables. Un* **processus ponctuel marqué** *dont les positions sont dans χ 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:

$$\rho((x,k),(y,l)) = \max\{d(x,y),d'(k,l)\}$$

Dans les cas où l'espace des marques $\mathcal{K}$ est fini:

$$\mathcal{K} = \{1,\ldots,M\} \qquad d'(k,l) = |k - l|$$

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.

Par contre, dans le cas où $\mathcal{K}$ est continu, il faut faire plus attention. Par exemple, un processus de poisson sur $\mathbb{R}^3$ (voir 1.2.3) n'est pas un processus ponctuel marqué sur $\mathbb{R}^2$ avec ses marques dans $\mathbb{R}$, puisque

$$N_X(A) = N_Y(A \times \mathbb{R})$$

n'est pas nécessairement fini pour un borélien $A \subseteq \mathbb{R}^2$ quelconque.

Exemple : Un processus ponctuel multi dimensionnel:

$$Y = (X_1, \dots X_M)$$

peut être vu comme un processus ponctuel marqué dont les marques définissent le numéro de la composante. Comme l'espace des marques est fini, le processus $X = \bigcup_{i=1}^{i=M} X_i$ des points non-marqués est bien défini.

1.1.4 Processus ponctuels simples.

On s'interesse 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 $N_{\mathbf{x}}(\{x\}) \in \{0,1\} \quad \forall x \in \mathbf{x}$.

Pour vérifier que $\mathcal{N}_s^{lf}$ est $\mathcal{N}^{lf}$ -mesurable, il suffit de noter que, comme χ est séparable, il peut être recouvert par une union dénombrable de boules ouvertes $B(x_i,(\frac{1}{2})^j)$ de rayon arbitrairement petit. Par conséquent,

$$N_s^{lf} = \bigcup_{j=1}^{\infty} \{\omega \in \Omega : N(B(x_i, \frac{1}{2}^j)) \in \{0,1\}\} \quad \in \mathcal{N}^{lf}$$

Définition 6 *Un processus ponctuel X est dit* **simple** *s'il prend ses valeurs dans $\mathcal{N}_s^{lf}$ presque surement.*

De manière intuitive, un processus ponctuel simulé suivant une loi uniforme (il reste à définir ce que veut dire uniforme) serait simple.

Il est intéressant de travailler avec les processus ponctuels simples puisque, pour connaître la loi d'un tel processus ponctuel, il suffit de connaître les **probabilités vides**:

$$v(A) = \mathbf{P}(N(A = 0))$$

pour une classe suffisamment grande d'ensembles $A \subseteq \chi$.

Théorème 2 *La loi d'un processus ponctuel simple X sur un espace métrique complet et séparable (χ,d) est uniquement déterminée par les probabilités vides des boréliens bornés $A \subseteq \chi$.*

Il faut se référer à [22] pour une démonstration assez technique.

1.2 Processus ponctuels finis.

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 surement 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 pontuels. La notion de densité est en effet difficile à mettre en place pour des processus qui ne sont pas finis.

1.2.1 Construction.

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. Une famille de densités de probabilité symétriques $j_n(x_1,\ldots,x_n),n \in \mathbb{N}$ sur χ^n pour les positions des points.

Le deuxième point implique que l'on suppose que χ soit équipé d'une mesure borélienne $\nu(.)$ de manière à pouvoir définir les densités j_n par rapport à la mesure produit $\nu(.)^n$.
On peut alors construire un processus ponctuel X de la manière suivante :
On considére $N(\chi)$ une variable aléatoire de loi $(p_n)_n$, et on conditionne ensuite par rapport aux évènenements $N(\chi) = n$: on prend $(X_1,\ldots,X_n) \in \chi^n$ un vecteur aléatoire de loi $j_n(.,\ldots,.)$ indépendamment de $N(\chi)$. La condition de symétrie est nécessaire puisqu'une configuration est indifférente à l'ordre dans lequel ses points sont recencés.

Il convient de vérifier toutefois si l'on peut passer légalement 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 N^f comme l'ensemble des configurations finies. Ensuite, appelons $\mathcal{N}^f$ la plus petite σ-algèbre pour laquelle les applications $\mathbf{x} \to N_{\mathbf{x}}(A)$ (A borélien borné) sont mesurables.
On définit ensuite les sous ensembles de N^f suivants :

$$N_n^f = \{\mathbf{x} \in N^f : N_{\mathbf{x}}(\chi) = n\}$$

et, à chacun de ces ensembles, on associe une σ-algèbre $\mathcal{N}_n^f$ en prenant la trace de $\mathcal{N}^f$ sur N_n^f.
On considère maintenant les fonctions $f_n : \chi^n \to N_n^f$ qui à des vecteurs de taille n associent des configurations de n points. Les f_n sont mesurables par rapport à la σ-algèbre des boréliens, et grâce à leur invariance par permutation, mesurable par rapport à la σ-algèbre $\mathcal{B}_s(\chi^n)$ des boréliens symétriques dans χ^n.

Par hypothèse, pour tout n, $j_n(.,\ldots,.)$ est une densité invariante par permutation et est donc $\mathcal{B}_s(\chi^n)$-mesurable.
Considérons la fonction $i_n : N_n^f \to \mathbb{R}$ 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 φ qui à une configuration associent l'un des n-uplets correspondant. (Ces n-uplets diffèrent par des permutations, il y a donc $n!$ φ différentes.)
On a donc : $i_n \circ f_n = j_n$.

Montrons que i_n est mesurable.

Soit $A \subseteq \chi$ un borélien borné. On doit vérifier que $i_n^{-1}(A) \in \mathcal{N}_n^f$. Or f_n est surjective, donc :

$$i_n^{-1}(A) = f_n(f_n^{-1}(i_n^{-1}(A))) = f_n(j_n^{-1}(A)$$

Comme j_n est mesurable par rapport à la σ algèbre des boréliens symétriques, il suffit de montrer que $f_n(B) \in \mathcal{N}_n^f$ pour tout $B \in \mathcal{B}_s(\chi^n)$.
Définissons la σ-algèbre

$$\mathcal{A}_n = \{B \in \mathcal{B}_s(\chi^n) : f_n(B) \in \mathcal{N}_n^f\}.$$

Alors $\mathcal{A}_n$ contient tous les rectangles A^n où A est un borélien borné et, puisque de tels rectangles génèrent $\mathcal{B}_s(\chi^n)$, il suit que $\mathcal{A}_n = \mathcal{B}_s(\chi^n)$.
Par conséquent $f_n(B) \in \mathcal{N}_n^f$ pour tout $B \in \mathcal{B}_s(\chi^n)$, et il y a une correspondance entre les fonctions sur l'espace des configurations N_n^f et les fonctions symétriques sur χ^n.

1.2.2 Processus ponctuel binomial.

Soit χ un compact de $\mathbb{R}^d$ de volume strictement positif $\mu(\chi)$. Un processus ponctuel binomial est défini comme l'union

$$X = \{X_1, \ldots, X_n\}$$

d'un nombre n fixé de points indépendants et uniformément distribués $X_1, \ldots, X_n$.
Comme $\mathbf{P}(X_i = X_j) = 0$ pour tout $i \neq j$, X est simple. De plus, comme $\mathbf{P}(N(\chi) = n) = 1$, le processus binomial est fini avec :

$$p_m = \begin{cases} 0 & \text{si} \quad m \neq n \\ 1 & \text{si} \quad m = n \end{cases}$$

Les points X_i sont uniformément distribués, donc :

$$j_n(x_1, \ldots, x_n) = \left(\frac{1}{\mu(\chi)}\right)^n$$

On le voit, les j_n sont invariantes par permutation.
Le processus binomal doit son nom au fait que pour tout borélien $A \subseteq \chi$,

$$N(A) = \sum_{i=1}^{i=n} \mathbf{1}_A(X_i)$$

suit une loi binomiale de paramètres n et $\mu(A)/\mu(\chi)$.

1.2.3 Processus ponctuel de Poisson.

Les processus ponctuels de Poisson sont les processus les plus élémentaires puisque, physiquement, ils traduisent la notion d'uniformité.

Définition 7

Soit $\nu(.)$ une mesure borélienne sur un espace métrique et séparable (χ,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 χ est appelé un **processus ponctuel de Poisson** *de mesure d'intensité $\nu(.)$ 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 $A_1, \ldots, A_k$, les variables aléatoires $N(A_1). \ldots, N(A_k)$ sont indépendantes.*

Si $\nu(.)$ 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 $\lambda\mu(.)$ où μ est la mesure de Lebesgue et λ un paramètre strictement positif. On appelle alors ce paramètre l'intensité du processus.

La propriété **P2** peut être interpretée 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 (χ,d) de mesure d'intensité $\nu(.)$, et $A \subseteq \chi$ un borélien borné. Alors, conditionellement à $\{N(A) = n\}$, X restreint à A suit la loi d'un processus binomial de points indépendants, ν-uniformément distribués sur A.*

La preuve repose sur le théorème 2:
Soit un borélien $B \subseteq A$. La mesure de probabilité vide de B, sachant qu'il y a n points dans A est donnée par :

$$\nu_A(B) = \mathbf{P}\left(N(B) = 0 \mid N(A) = n\right) = \frac{\mathbf{P}\left(N(B) = 0\ ;\ N(A\backslash B) = n\right)}{\mathbf{P}\left(N(A) = n\right)}$$

Grace à **P2** on sait que $N(A)$ et $N(A\backslash B)$ sont indépendantes. De plus $N(A)$,$N(B)$ et $N(A\backslash B)$ suivent une loi de Poisson d'après **P1**. Donc :

$$\nu_A(B) = \left(\frac{\nu(A\backslash B}{\nu(A)}\right)^n$$

ce qui coincide avec les probabilités vides de B pour n points indépendants et de loi ν-uniforme. Comme B est quelconque, l'application du théorème 2 termine la preuve.

1.2.4 Processus ponctuels finis définis par une densité.

L'objectif de cette partie est de construire des modeles de processus ponctuels par leur densité de probabilité (dérivée de Radon-Nikodym) par rapport à un processus de Poisson de référence.

Cas des processus ponctuels non finis.

Le lemme suivant montre qu'il est difficile de définir des densités pour des processus ponctuels qui ne sont pas finis.

Lemme 1 *Soient X_λ et X_μ deux processus de Poisson homogènes sur $\mathbb{R}^d$, définis sur le même espace probabilisé $(\Omega,\mathcal{A},\mathbb{P})$, de mesure d'intensité respective λ et μ. Si $\lambda \neq \mu$, alors la loi de X_λ n'est pas absolument continue par rapport à la loi de X_μ.*

Preuve : Considérons la familles $(B_n)_n$ des boules fermées centrées en l'origine, de rayon tel que le volume de B_n soit égal à n. Pour tout $\nu > 0$, posons :

$$E_\nu = \left\{\omega \in \Omega : \frac{N_{X_\nu(\omega)}(B_n)}{n} \to \nu\right\}$$

Pour $i \in \{1,2,\ldots\}$, posons $L_i = B_i \setminus B_{i-1}$. Les L_i sont disjoints, et d'après la définition des processus de Poisson, sous la loi de X_λ, les variables aléatoires $N(L_i)$ sont indépendantes, et ont pour loi une loi de Poisson d'espérance λ. D'après la loi forte des grands nombres, on a donc :

$$\frac{1}{n}N(B_n) = \frac{1}{n}\sum_{i=1}^{n} N(L_i) \to \mathbb{E}_{X_\lambda}\left[N(L_1)\right] = \lambda \qquad \text{p.s.}$$

On en conclut que $\mathbf{P}(X_\lambda \in X_\lambda(E_\lambda)) = 1$ et que $\mathbf{P}(X_\mu \in X_\mu(E_\mu)) = 1$. C'est une sorte de loi forte des grands nombres spatiale : lorsque les boules grandissent, les réalisations du processus ponctuel X_ν obéissent de mieux en mieux à la contrainte "le nombre moyen de points par unité de volume" est ν.
Comme $X_\lambda(E_\lambda)$ et $X_\mu(E_\mu)$ sont disjoints (car $\lambda \neq \mu$), on en déduit que la loi de X_λ n'est pas absolument continue par rapport à celle de X_μ.

On se souvient que la définition d'un processus ponctuel fini impose que le nombre total de points dans l'espace considéré soit fini presque surement. Donc, si l'on remplace les processus homogènes ci-dessus par des processus ponctuels finis, le nombre moyen de points par unité de volume ne tend plus vers l'intensité lorsque l'on augmente la taille du volume d'intérêt mais vers zéro. La preuve ci dessus ne s'applique donc plus du tout...

Cas des processus ponctuels finis.

Dans la suite de ce paragraphe, on considère (χ,d) un espace métrique complet et séparable, et $\pi(.)$ la distribution d'un processus de Poisson sur χ de mesure d'intensité **finie** et **non-atomique** $\nu(.)$.
Soit $p : N^f \to [0,\infty[$ une fonction positive, mesurable, à valeur dans l'espace des configurations finies de points, telle que :

$$\int_{N^f} \mathbf{p}(\mathbf{x}) d\pi(\mathbf{x}) = 1. \tag{1.1}$$

Alors $p(.)$ est une densité de probabilité et définit un processus ponctuel X sur χ. Comme le processus de poisson dominant est fini et simple, il en va de même pour X . Pour interpréter et justifier la formule 1.1, 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 ν-mesure de N^f est $\nu(\chi)^n/n!$. Ce $n!$ est nécessaire puisque χ^n est ordonné tandis que N^{lf} 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 point d'un processus définit par sa densité p(.) est donné par la famille $(p_n)_{n\in\mathbb{N}}$:

$$p_n = \frac{e^{-\nu(\chi)}}{n!} \int_\chi \cdots \int_\chi p(\{x_1,\ldots,x_n\}) d\nu(x_1)\cdots d\nu(x_n) \tag{1.2}$$

et que d'autre part, conditionnellement aux évênements $\{N(\chi) = n\}$, les n points aléatoires ont pour densité jointe:

$$j_n(x_1,\ldots,x_n) = \frac{p(\{x_1,\ldots,x_n)\}}{\int_\chi \cdots \int_\chi p(\{x_1,\ldots,x_n\}) d\nu(x_1)\cdots d\nu(x_n)} \tag{1.3}$$

1.3 Propriétés générales et exemples.

1.3.1 Mesure et théorème de Campbell.

Il s'agit ici de définir des outils qui expriment des grandeurs moyennes ou d'ordre supérieur sur les processus ponctuels. Pour cela, le meilleur moyen est d'utiliser le fait que pour tout borélien borné A, $N(A)$ soit une variable aléatoire : les moments de $N(A)$ fournissent des statistiques très utiles.

Définition 8 *Soit X un processus ponctuel sur un espace métrique complet et séparable (χ,d). Pour tout borélien borné $A \subseteq \chi$ et tout $F \in \mathcal{N}^{lf}$, on définit les grandeurs:*

$$M(A) = \mathbb{E}[N(A)]$$

et:

$$C(A \times F) = \mathbb{E}[N(A).\mathbf{1}\{X \in F\}]$$

Le lemme suivant donne les conditions sous lesquelles les fonctions ensemblistes $M(.)$ et $C(. \times .)$ peuvent être prolongées en mesures sur la σ-algèbre borélienne $\mathcal{B}(\chi)$ de χ et la σ-algèbre produit $\mathcal{B}(\chi) \times \mathcal{N}^{lf}$.

Lemme 2 *Si la fonction $M(.)$ introduite dans la définition 8 est finie sur les boréliens bornés, alors $M(.)$ peut être prolongée de manière unique en une mesure σ-finie sur les boréliens de χ.*
On l'appelle alors **mesure du moment de premier ordre.**

La fonction $C(. \times .)$ peut être prolongée de manière unique en une mesure σ finie sur la σ algèbre produit obtenue à partir de $\mathcal{B}(\chi)$ et N^{lf} .
On l'appelle alors **la mesure de Campbell de premier ordre**.

La mesure du moment de premier ordre peut être exprimée grâce à la mesure de Campbell de la manière suivante :

$$C(A \times N^{lf}) = \mathbb{E}[N(A)] = M(A)$$

pour tout borélien borné A. Par conséquent, $C(.)$ est finie si et seulement si $M(.)$ est finie, c'est à dire si et seulement si $\mathbb{E}[N(\chi)] < \infty$. Les mesures de Campbell d'ordres supérieurs sont définies suivant le même principe, par exemple :

$$C^{(2)}(A \times B \times F) = \mathbb{E}[N(A)N(B).\mathbf{1}\{X \in F\}]$$

Les mesures de Campbell sont importantes lorsque l'on travaille avec des processus ponctuels. Supposons que l'on regarde une grandeur $g(x,\mathbf{X})$ en chaque point x d'une réalisation $\mathbf{x}$ de X . Par exemple $g(x,\mathbf{x})$ peut être la distance entre x et le point le plus proche de la réalisation $\mathbf{x}$. Le théorème suivant permet d'affirmer que l'espérance $\mathbb{E}[\sum_{x\in X} g(x,X)]$ est l'intégrale de $g(.,.)$ contre la mesure de Campbell.
Du point de vue pratique, cela nous est utile pour avoir des critères de convergence pour nos algorithmes.
Une autre utilisation de cette mesure, théorique celle là, concerne la définition de l'intensité conditionnelle d'un processus ponctuel (voir [22]).

Théorème 4 **Formule de Campbell-Mecke**
Soit $g : \chi \times N^{lf} \to \mathbb{R}$ une fonction mesurable. Si g est positive ou alors intégrable par rapport à la mesure de Campbell, on a :

$$\mathbb{E}\left[\sum_{x\in X} g(x,X)\right] = \int_{\chi}\int_{N^{lf}} g(x,X)dC(x,X) \tag{1.4}$$

Si la fonction g ne dépend pas du processus ponctuel X , la formule de Campbell Mecke se réduit à :

$$\mathbb{E}\left[\sum_{x\in X} g(x)\right] = \int_{\chi} g(x)dM(x) \tag{1.5}$$

pourvu que la mesure du moment de premier ordre $M(.)$ existe et soit finie sur les boréliens.

En particulier, pour un processus de Poisson homogène d'intensité $\lambda > 0$,

$$\mathbb{E}\left[\sum_{x\in X} g(x)\right] = \lambda \int_{\chi} g(x)dx$$

1.3.2 Lois conditionnelles.

Il existe deux façons naturelles de conditionner la loi d'un processus ponctuel:

- **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 lapropriété d'invariance de la loi du processus ponctuel par translation. En particulier il permet de définir et d'utiliser des statistiques spatiales (voir [24] et [37]). Dans nos applications, nous ne disposons pas de telles propriétés. Nous nous limitons donc au second type de conditionnement.

Le conditionnement extérieur.

Commençons par une présentation intuitive de la **densité conditionnelle de Papangelou** $\lambda(.;.)$ 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.
Pout définir cette densité conditionnelle plus proprement, il est nécessaire d'introduire la notion de **mesure de Campbell réduite**.

Définition 9 *Soit X un processus ponctuel simple, sur un espace métrique complet et séparable (χ,d). On définit :*

$$C^!(A \times F) = \mathbb{E}\left[\sum_{x\in X\cap A} \mathbf{1}\left\{X \setminus \{x\} \in F\right\}\right]$$

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 σ-finie unique sur la σ-algèbre produit $\mathcal{B}(\chi) \times \mathcal{N}^{lf}$. L'équivalent de la formule de Cambell-Mecke donne :

$$\mathbb{E}\left[\sum_{x\in X} g(x,X \setminus \{x\})\right] = \int_{\chi}\int_{N^{lf}} g(x,\mathbf{x})\ dC^!(x,\mathbf{x}). \tag{1.6}$$

pout toute fonction mesurable $g(.,.)$ qui soit ou bien positive ou bien intégrable.

Supposons que pour un borélien borné fixé $A \in \mathcal{B}(\chi)$, $C^!(A \times .)$ soit absolument continue par rapport à la loi $\mathbf{P}(.)$ de X . Alors,

$$C^!(A \times F) = \int_F \Lambda(A;\mathbf{x})\ d\mathbf{P}(\mathbf{x})$$

pour une fonction mesurable $\Lambda(A;.)$ définie de manière unique à un ensemble **P** négligeable près. De plus, (voir [6]) on peut en trouver une version telle que pour $\mathbf{x}$ fixé, $\Lambda(.;\mathbf{x})$ soit une mesure borélienne localement finie, que l'on appelle **noyau de Papangelou** du premier ordre.
Si $\Lambda(.;\mathbf{x})$ admet une densité $\lambda(.;\mathbf{x})$ par rapport à une mesure de référence $\nu(.)$ sur χ, 1.6 se factorise sous la forme :

$$\mathbb{E}\left[\int_\chi g(x,X)\lambda(x;X)d\nu(x)\right] \tag{1.7}$$

On appelle alors la fonction $\lambda(.;.)$ **l'intensité conditionnelle de Papangelou**.

Le cas où X est dominé par un processus ponctuel de Poisson est particulièrement important :

Théorème 5 *Soit X un processus ponctuel fini définit par une densité $p(\mathbf{x})$ par rapport à un processus ponctuel de Poisson de mesure d'intensité finie et non-atomique $\mu(.)$. Alors X a une intensité conditionnelle de Papangelou :*

$$\lambda(u;(x)) = \frac{p(\mathbf{x}\cup\{u\})}{p(\mathbf{x})} \tag{1.8}$$

pour $u \notin \mathbf{x}$ et $\mathbf{x} \in N^f$.

Preuve : On utilise la caractéristation intégrale donnée par les équations 1.6 et 1.7. Pour cela, considérons $g : \chi \times N^{lf} \to \mathbb{R}$ une fonction positive ou intégrable. En injectant 1.8 dans 1.7, et en utilisant le fait que la loi de X est donnée par une densité $p(.)$, on obtient :

$$\mathbb{E}\left[\int_\chi g(u,X)\lambda(u;X)d\nu(u)\right] =$$

$$\sum_{n=0}^{\infty}\frac{e^{-\nu(\chi)}}{n!}\int_{\chi^{n+1}} g(u,\{x_1,\ldots,x_n\})\frac{p(\{x_1,\ldots,x_n\}\cup\{u\})}{p(\{x_1,\ldots,x_n\})}p(\{x_1,\ldots,x_n\})d\nu(x_1)\cdots d\nu(x_n)d\nu(u) =$$

$$\sum_{n=0}^{\infty}\frac{e^{-\nu(\chi)}}{n!}\frac{1}{n+1}\int_{\chi^{n+1}}\sum_{i=1}^{n+1} g(x_i,\{x_1,\ldots,x_{n+1}\}\setminus\{x_i\})p(\{x_1,\ldots,x_{n+1}\})d\nu(x_1)\ldots d\nu(x_{n+1}) =$$

$$\sum_{n=1}^{\infty}\frac{e^{-\nu(\chi)}}{n!}\int_{\chi^{n}}\sum_{i=1}^{n+1} g(x_i,\{x_1,\ldots,x_n\}\setminus\{x_i\})p(\{x_1,\ldots,x_n\})d\nu(x_1)\cdots d\nu(x_n).$$

Comme cette dernière expression s'identifie à l'espérance dans le membre de gauche de l'équation 1.6, et que $g(.;.)$ est quelconque, on en conclut que $\lambda(.;.)$ satisfait la représentation intégrale et la preuve est complète.

Il faut noter que les étapes de cette démonstration nous resservirons dans la partie 2.4.1 et dans le chapitre 4 de ce manuscrit.

1.3.3 Processus ponctuels de Markov.

La définition intuitive des processus de Markov est assez simple : il s'agit de processus ponctuels finis défini 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 implémentation 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 densitéd sous une forme énergétique en utilisant des potentiels d'interaction entre les points d'une réalisation du processus ponctuel. Pour plus de détails, on pourra se référer à [3],[37] ou encore à [27] et [35]

Nous présentons ici rapidement quelques définitions et propriétés des processus de Markov. Ces définitions sont seulement culturelles, puisque le modèle adopté dans notre application n'est pas markovien.

On considère une relation symétrique et réflexive $\sim$ sur χ. On dit que deux points u et v de χ sont voisins si $u \sim v$. Par exemple, on peut considérer la relation de proximité sur $\chi = \mathbb{R}^2$:

$$u \sim v \Leftrightarrow d(u,v) \leq R$$

pour un réel R quelconque.

Définition 10 *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: (voir [30]).

Définition 11 *Soit* (χ,d) *un espace métrique complet et séparable,* $\nu(.)$ *une mesure borélienne finie non atomique, et* $\pi_\nu(.)$ *la loi d'un processus ponctuel de Poisson de mesure d'intensité* $\nu(.)$.
Soit X *un processus ponctuel sur* χ *définit par sa densité* $p(.)$ *par rapport à* $\pi_\nu(.)$.

Alors X *est un* **processus ponctuel de Markov** *sous la relation symétrique et réflexive* $\sim$ *sur* χ *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 12 *Soit* $\sim$ *une relation de voisinage symétrique et réflexive sur* χ. *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:*

$$\forall u,v \in \mathbf{x} \qquad u \sim v$$

Par convention la configuration vide est une clique également.

On peut ensuite démontrer le théorème suivant (voir [30]), qui est l'équivalent du théorème d'Hammersley-Clifford pour les processus ponctuels :

Théorème 6 *Une densité de processus ponctuel* $p : N^f \to [0,\infty[$ *est markovienne sous une relation de voisinage* $\sim$ *si et seulement si il existe une fonction mesurable* $\phi : N^f \to [0,\infty[$ *telle que :*

$$p(\mathbf{x}) = \prod_{\text{cliques } \mathbf{y} \subseteq \mathbf{x}} \phi(\mathbf{y}) \tag{1.9}$$

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 ϕ de valoir 1 sur les sous ensembles n'étant pas des cliques.

1.3.4 Exemples.

On se place sur $\chi = \mathbb{R}^2$, et on considère un sous ensemble borné $K \subseteq \chi$ muni de la mesure de lebesgue $\lambda(.)$.

On définit des densités par rapport à un processus ponctuel de Poisson sur K, d'intensité $\lambda(.)$. Un tel processus est simple et fini, comme il se doit pour pouvoir définir des densités de processus.

Paramètres d'intensité.

On considére d'abord des processus ayant une densité $h(.)$ de la forme:

$$h(\mathbf{x}) = \alpha.\beta^{n(\mathbf{x})}$$

où $\beta > 0$ et $n(\mathbf{x})$ représente le nombre de points d'une configuration $\mathbf{x}$. Une telle densité est intégrable et α est la constante de normalisation. La constante β est un paramètre qui permet de jouer sur l'intensité du processus. Celle ci vaut alors $\beta\lambda(.)$.

Il est intéressant de remarquer que ce β est un paramètre d'échelle: de manière équivalente on peut changer le borné K d'intérêt en posant K' tel que $\lambda(K') = \beta\lambda(K)$.

Processus de Strauss.

On considére une densité de la forme:

$$p(\mathbf{x}) = \alpha.\beta^{n(\mathbf{x})} \prod_{i<j} g(x_i,x_j) \qquad \mathbf{x} = \{x_1, \ldots, x_{n(\mathbf{x})}\} \tag{1.10}$$

Où la fonction g vérifie:

$$g(x_i,x_j) = \begin{cases} \gamma, & \text{si} \quad d(x_i,x_j) < r \\ 1, & \text{si} \quad d(x_i,x_j) \geq r \end{cases}$$

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:

$$p(\mathbf{x}) = \alpha\beta^{n(\mathbf{x})}\gamma^{s(\mathbf{x})} \tag{1.11}$$

où $s(\mathbf{x})$ représente le nombre de paires de points en relation dans la configuration $\mathbf{x}$. Reste à étudier l'influence de γ:

- Pour $\gamma = 1$ on retrouve un processus ponctuel de Poisson sur notre borélien borné K d'intensité $\beta\lambda(.)$.
- 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 interressait Strauss ([38]) 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.
Il y a deux façons de voir que cette densité n'est pas intégrable:

- La première, proposée par Van Lieshout et Baddeley ([2]), repose sur une optique combinatoire: intuitivement, plus on ajoute de points sur un petit espace et plus la densité augmente vite.
- La deuxième interprétation, proposée par Geyer ([11]), repose sur les familles exponentielles. Elle est trés intéressante puisqu'elle propose un formalisme pour étudier les comportements limites du processus par les familles exponentielles.

Nous reviendrons plus loin sur cet exemple et les problèmes qu'il soulève.

Chapitre 2

Simulation de Processus Ponctuels : méthodes MCMC.

Ce chapitre a pour objectif de présenter les algorithmes utilisés pour simuler des processus ponctuels définis par leur densité. Dans le cadre de notre application, nous avons utilisé un algorithme de Monte Carlo utilisant les chaînes de Markov. Nous commençons donc ce chapitre par des rappels sur les chaînes de Markov et la théorie ergodique. Ces rappels sont essentiellement tirés des livres de C.Robert et G.Casella [31] et de Meyn et Tweedie [25]. Viennent ensuite des discussions plus particulièrement centrées sur le problème de la simulation de processus ponctuels et les problématiques soulevées par ses simulations.

Pour comprendre les enjeux de ce chapi tre, ll faut garder à l'esprit que son objectif final est de proposer des méthodes permettant de simuler des processus ponctuels ayant une densité donnée par rapport à un processus de Poisson.

2.1 Rappels sur les chaînes de Markov : premières définitions.

2.1.1 Notions fondamentales.

Dans toute cette partie, on considère un espace probabilisé $(\Omega,\mathcal{A},\mathbf{P})$ ainsi qu'un espace χ muni de sa tribu $\mathcal{B}(\chi)$. Nous allons construire des variables aléatoires à valeur dans cette espace $(\chi,\mathcal{B}(\chi))$ Commençons par quelques définitions :

Définition | **Noyau de transition**
C'est une fonction K définie sur $\chi \times \mathcal{B}(\chi)$ telle que:
1 $\forall x \in \chi$, *$K(x,.)$ est une mesure de probabilité.*
2 $\forall A \in \mathcal{B}(\chi)$, *$K(.,A)$ est mesurable.*

Cette définition nous permet d'en donner une autre:

Définition | *Une* **chaîne de Markov** *est une suite de variable aléatoires $(X_n)_{n\in\mathbb{N}}$ à valeur dans χ telle que, pour tout entier $k \in \mathbb{N}$ et $A \in \mathcal{B}(\chi)$*

$$\mathbf{P}(x_{k+1} \in A | x_0, \ldots, x_k) = \mathbf{P}(X_{k+1} \in A | x_k) \quad (2.1a)$$
$$= \int_A K(x_k, dx) \quad (2.1b)$$

En général, on restreint l'étude des chaînes de Markov aux chaînes de Markov **homogènes**, c'est à dire aux chaînes de Markov telles que la loi de $(X_{t_1},\dots,X_{t_k})$ sachant x_0 soit la même que la loi de $(X_{t_1-t_0},\dots,X_{t_k-t_0})$ sachant x_0 pour tout k et tout $(k+1)$-uplets $t_0 \leq t_1 \leq \dots \leq t_k$.

Quelques notations.

Quelques rappels sont faits ci-dessous pour comprendre l'utilisation du noyau K, ainsi que pour donner quelques notations usuelles :

$$\mathbf{P}_x(X_1 \in A_1) = K(x,A_1)$$

$$\mathbf{P}_x(X_1 \in A_1,X_2 \in A_2) = \int_{A_1} K(y,A_2)K(x,dy)$$

$$\mathbf{P}_x((X_1,\dots,X_n) \in A_1 \times \dots \times A_n) = \int_{A_1} \dots \int_{A_n} K(y_{n-1},A_n)K(y_{n-2},dy_{n-1})\dots K(x,dy)$$

Voila d'autres notations liées aux transitions en n étapes:

$$K^1(c,A) = K(x,A)$$

$$K^n(x,A) = \int_{\chi} K^{n-1}(y,A)K(x,dy)$$

Pour finir ce paragraphe de notations, il convient d'expliquer les notations suivantes :
- $\mathbf{P}_x(.)$ note la loi de la chaîne $(X_n)_n$ avec $X_0 = x$
- $\mathbf{P}_\mu(.)$ note la loi de la chaîne avec une distribution initale : $X_0 \sim \mu$.
- $\mathbb{E}_\mu[.]$ note l'espérance associée à la loi $\mathbf{P}_\mu(.)$.

Propriétés premières des chaînes de Markov.

On retrouve ici des propriétés ususelles telles que les équations de Chapman-Kolmogorov, la propriété de Markov faible et la propriété de Markov forte. Nous ne nous étendons pas sur ces propriétés usuelles et intuitives. Pour plus de détails on peut se référer à [31]. Robert et Casella définissent à cet endroit de la théorie un outil théorique important pour les démonstrations des propositions et théorèmes qui vont suivre: le noyau résolvant.

Règles d'arrêt et grandeurs associées.

On commence par une définition:

Définition *On appelle* **règle d'arrêt** *toute fonction $\zeta(x_1,\dots,x_n,\dots)$ à valeur dans* $\mathbf{N}$ *telle que les évênements $\{\zeta = n\}$ soient mesurables pour la tribu engendrée par $(X_0,\dots,X_n)$.*

Un exemple important de règle d'arrêt est le suivant: pour un ensemble $A \in \mathcal{B}(\chi)$, on appelle **temps d'arrêt en** A le premier instant n où la chaîne passe par cet ensemble :

$$\tau_A = \inf\{n \geq 1; X_n \in A\} \tag{2.2}$$

par convention, $\tau_A = \infty$ si la chaîne ne passe jamais par A. Il est clair que l'évênement $\{\tau_A = n\}$ ne dépend que des variables aléatoires $(X_0,\dots,X_n)$.

Dans le même esprit, on définit une autre grandeur aléatoire, **le nombre de passage en** A :

$$\eta_A = \sum_{n=1}^{\infty} \mathbf{1}_A(X_n) \tag{2.3}$$

On va voir que ces variables aléatoires sont d'une importance capitale dans l'étude des chaînes de Markov, et en particulier les deux grandeurs associées: $\mathbb{E}_x[\eta_A]$ et $\mathbf{P}_x(\tau_A < \infty)$ qui, intuitivement, représentent le nombre moyen de passages en A de la chaîne de markov et la probabilité d'aller en un temps fini en A.

2.1.2 Irréductibilité.

Dans toute cette partie, φ représente une mesure sur χ.

Définition *On considère (X_n) une chaîne de Markov de noyau de transition K. Cette chaîne est dite φ-**irréductible** si pour tout $A \in \mathcal{B}(\chi)$ tel que $\varphi(A) > 0$ l'une des deux propriétés équivalentes suivantes est vérifiée :*
- *pour tout $x \in \chi$, $\exists n$ tel que $K^n(x,A) > 0$*
- *pour tout $x \in \chi$, $\mathbf{P}_x(\tau_A < \infty) > 0$*

Intuitivement, une chaîne est irréductible si tout ensemble non négligeable au sens de φ est atteignable en un temps fini avec une probabilité non nulle.

Proposition *La chaîne de markov (X_n) est φ-irréductible si et seulement si pour tout $x \in \chi$ et tout $A \in \mathcal{B}(\chi)$ tel que $\varphi(A) > 0$, on a :*

$$\mathbb{E}_x[\eta_A] > 0 \tag{2.4}$$

Une propriété intéressante des chaînes irréductibles sur laquelle nous ne nous étendons pas ici concerne l'existence d'une mesure pour laquelle la chaîne est irréductible, qui domine (au sens de l'absolue continuité) toute mesure pour laquelle la chaîne est irréductible et qui permet d'avoir une forme de réciproque de l'irréductibilité: un ensemble négligeable au sens de cette mesure n'est jamais atteint en un temps fini, sauf peut-être en partant d'un ensemble de points lui-même négligeable.

2.1.3 Atomes et ensembles petits.

Le propos de cette partie est de se donner des outils permettant de travailler sur les chaînes de Markov dite “continues” de façon similaire aux chaînes “discrètes”.

Définition *La chaîne de Markov (X_n) possède un atome $\alpha \in \mathcal{B}(\chi)$ s'il existe une mesure $\nu(.) > 0$ telle que:*

$$K(x,A) = \nu(A) \qquad \forall x \in \alpha \qquad \forall A \in \mathcal{B}(\chi) \tag{2.5}$$

Si (X_n) est φ-irréductible quand $\varphi(\alpha) < 0$, l'atome est dit **accessible**. Cette notion d'atome est intéressante: idéalement on voudrait travailler avec des noyaux “constants par morceaux”. Malheureusement cette constance justement n'est pas très pratique puisqu'on la rencontre rarement dans la nature. D'où la condition de minoration suivante qui est préférée, et qui par suite a donné la notion d'**ensemble petit**:

Définition *Un ensemble C est* **petit** *s'il existe $m \in \mathbb{N}^*$ et une mesure ν_m telle que, pour tout $x \in C$ et tout $A \in \mathcal{B}(\chi)$ on ait :*

$$K^m(x,A) \geq \nu_m(A) > 0 \tag{2.6}$$

Les atomes sont donc un cas particulier d'ensembles petits... et voila le théorème, démontré dans [25] :

Théorème *Soit (X_n) une chaîne de markov φ-irréductible. Pour tout ensemble $A \in \mathcal{B}(\chi)$ tel que $\varphi(A) > 0$, il existe un ensemble petit $C \subseteq A$.*

De plus, χ se décompose en une partition dénombrable d'ensembles petits.

Il apparait toutefois que les atomes sont plus intéressants que les ensembles petits puisque le noyau y est constant. On peut, en utilisant la théorie des temps de renouvellement, montrer que si :

- d'une part la chaîne vérifie la condition (2.6) pour un ensemble C ,
- et que d'autre part on a : $\mathbf{P}_x(\tau_C < \infty)$ qui vaut 1 pour tout $x \in \chi$,

on construire une chaîne compagne à (X_n) possédant un atome (voir [31]). Cette construction utilise les temps de renouvellement :

Définition *Un* **temps de renouvellement** *est une règle d'arrêt τ telle que $(X_\tau,X_{\tau+1},\ldots)$ soit indépendant de $(X_{\tau-1},X_{\tau-2},\ldots)$*

2.1.4 Cycles et apériodicité.

Ces notions sont définies pour des chaînes (X_n) φ-irréductibles. Voila une première définition, plutôt intuitive :

Définition *Les ensembles $A_1,\ldots,A_m$ de $\mathcal{B}(\chi)$ forment un m-cycle de la chaîne (X_n) si :*

$$\left\{\begin{array}{ll} \mathbf{P}(x,A_2) = 1 & \text{pour } x \in A_1 \\ \vdots & \\ \mathbf{P}(x,A_m) = 1 & \text{pour } x \in A_{m-1} \\ \mathbf{P}(x,A_1) = 1 & \text{pour } x \in A_m \end{array}\right. \quad \text{et } \varphi\left(\left[\bigcup_{i=1}^m A_i\right]^c\right) = 0;$$

On en déduit la notion d'âpériodicité :

Définition *Une chaîne φ-irréductible est* **apériodique** *si elle possède un cycle de longueur $m = 1$.*

Voila enfin une définition plus mathématique d'un cycle :

Définition *Une chaîne (X_n) φ-irréductible possède un* **cycle de longeur** *d, s'il existe un ensemble petit C, un entier associé M et une mesure de probabilité ν_m (voir la définition d'un ensemble petit) tel que d soit le* pgcd *de :*

$$\{m \geq 1;\ \exists\, \delta_m \text{ tel que } C \text{ petit } pour\ \nu_m \geq \delta_m\, \nu_M\}$$

Il faut remarquer, dans ces définitions, l'utilisation des ensembles petits qui est faite en substitut des états dans le cas ou l'espace d'état est fini.

2.2 Rappels sur les chaînes de Markov : théorie ergodique.

2.2.1 Transience et récurrence.

Les définitions suivantes sont données pour des chaînes de markov (X_n) φ-irréductibles:

Définitions *Un ensemble $A \in \mathcal{B}(\chi)$ est dit* **récurrent** *si pour tout $x \in \chi$, $\mathbb{E}_x[\eta_A] = \infty$*

L'ensemble $A \in \mathcal{B}(\chi)$ est dit **uniformément transient** *s'il existe une constante M telle que $\mathbb{E}_x[\eta_A] < M$ pour tout $x \in \chi$.*

L'ensemble A est dit **transient** *si on peut le décomposer en une union dénombrable d'ensembles uniformément transients B_i :*

$$A = \bigcup_{i=1}^{\infty} B_i$$

Le théorème suivant est intéressant puisqu'il permet de comprendre l'intérêt de l'existence d'un atome pour une chaîne irréductible. On comprend en effet que pour une chaîne irréductible la propriété d'un atome peut se généraliser à tout l'espace d'état:

Théorème *Soit (X_n) une chaîne de Markov φ irréductible possédant un atome accessible α :*

- **(i)** *si α est récurrent, tout ensemble $A \in \mathcal{B}(\chi)$ tel que $\varphi(A) > 0$ est récurrent.*
- **(ii)** *si α est transient, χ est transient.*

D'où l'extension suivante des notions de transience et de récurrence aux chaines de Markov :

Définitions *Une chaîne de Markov (X_n) est* **récurrente** *si:*

- **(i)** *il existe une mesure φ telle que (X_n) soit φ-irréductible.*
- **(ii)** *pour tout $A \in \mathcal{B}(\chi)$ tel que $\varphi(A) > 0$:*

$$\forall x \in A \qquad \mathbb{E}_x[\eta_A] = \infty$$

La chaîne est **transiente** *si:*

- **(i)** *elle est φ-irréductible*
- **(ii)** *χ est transient.*

Pour finir cette partie, nous donnons le théorème de classification des chaines irréductibles :

Théorème *Une chaîne de Markov φ-irréductible est soit transiente, soit récurrente.*

Une dernière question se pose toutefois: comment distinguer une chaîne transiente d'une chaîne récurrente. La proposition suivante permet de répondre à cette question. Cette proposition est anecdotique , puisque comme on le verra dans la suite de ce chapître, en pratique la simple récurrence n'est pas intérressante à proprement parler.

Proposition *S'il existe un ensemble petit C avec $\varphi(X) > 0$ tel que :*

$$\mathbf{P}_x(\tau_C < \infty) = 1 \qquad \forall x \in C \tag{2.7}$$

la chaîne φ-irréductible est récurrente.

2.2.2 Récurrence au sens de Harris.

L'objectif de ces rappels est de donner des propriétés générales des chaines de Markov. Il se trouve que pour assurer certaines propriétés des algorithmes que nous allons présenter plus loin, la récurrence simple ne suffit pas.

Définitions *Un ensemble $A \in \mathcal{B}(\chi)$ est* **récurrent au sens de Harris** *si:*

$$\forall x \in A \qquad \mathbf{P}_x(\eta_A = \infty) = 1 \tag{2.8}$$

Une chaîne de Markov (X_n) est **récurrente au sens de Harris** *si:*
- **(i)** *il existe une mesure φ telle que (X_n) soit φ irréductible,*
- **(ii)** *pour tout ensemble A tel que $\varphi(A) > 0$, A est récurrent au sens de Harris.*

La différence avec la récurrence classique tient dans le fait que pour obtenir $\mathbb{E}_x[\eta_A] = \infty$ il est suffisant d'avoir $\mathbf{P}_x(\eta_A = \infty) > 0$.
On peut même avoir $\mathbf{P}_x(\eta_A = \infty)$=0 et $\mathbb{E}_x[\eta_A] = \infty$. Cela est du au fait que la condition de récurrence repose sur une espérance. Evidemment, la récurrence au sens de Harris implique la récurrence classique mais la condition de récurrence au sens de Harris est bien plus forte, puisqu'elle impose de repasser une infinité de fois en A avec une probabilité 1.

Voila tout de suite une proposition qui donne une autre caractérisation d'une chaîne récurrente au sens de Harris : Il suffit d'être sûr de repasser par A en un temps fini.

Proposition *Si pour tout $A \in \mathcal{B}(\chi)$, $\mathbf{P}_x(\tau_A < \infty) = 1$ pour tout $x \in A$, alors: $\mathbf{P}_x(\eta_A = \infty) = 1$ pour tout $x \in \chi$ et (X_n) est Harris-récurrente.*

2.2.3 Mesures invariantes.

Les mesures invariantes appartiennent aux objets classiques associés aux chaines de Markov. On les place ici dans un contexte lié aux notions de récurrence.

Définition *Une mesure σ-finie π est dite* **invariante** *pour le noyau de transition $K(.,.)$ si :*

$$\pi(B) = \int_{\chi} K(x,B)\pi(dx) \qquad \forall B \in \mathcal{B}(\chi) \tag{2.9}$$

Si de plus, la mesure invariante est une mesure de probabilité, elle est dite **mesure stationnaire**.
Si la chaîne de Markov (X_n) est φ-irréductible, et si elle possède une mesure stationnaire, on dit que cette chaîne est **positive**.

Proposition *Si la chaîne (X_n) est positive, elle est récurrente.*

Le théorème suivant donne une sorte de réciproque. En fait il assure surtout l'existence et l'unicité d'une mesure invariante pour les chaînes récurrentes. Il faut remarquer cependant qu'il n'assure pas l'intégrabilité de la mesure obtenue. Par conséquent, il n'assure pas que la chaîne soit positive.

Théorème *Si (X_n) est une chaîne récurrente, alors il existe une mesure σ-finie invariante, unique à un facteur multiplicatif près.*

Par extension, on parlera de chaîne **Harris positive** lorsque la chaîne sera Harris-récurrente, et sa mesure invariante intégrable. Il reste un dernier théorème à connaître et dans notre cadre d'application ce thérorème est très important. Il est démontré dans [25].

Théorème 7 *Si une chaîne de Markov (X_n) est φ irréductible pour une mesure $\varphi(.)$ non nulle sur $\mathcal{B}(\chi)$, et si elle posséde une loi stationnaire $\pi(.)$, alors la chaîne est aussi $\pi(.)$ irréductible.*

2.2.4 Ergodicité et convergence.

On arrive aux notions qui nous intéressent plus particulièrement:

Définition *Pour une chaîne Harris-positive, de loi de probabilité invariante $\pi(.)$, on dit d'un atome α qu'il est* **ergodique** *si :*

$$\lim_{n \to +\infty} |K^n(\alpha,\alpha) - \pi(\alpha)| = 0 \tag{2.10}$$

Cette notion d'ergodicité concerne donc la convergence d'une mesure vers une autre. En fait, cette partie a pour objectif d'étudier la convergence éventuelle de la loi d'une chaîne de Markov vers sa loi stationnaire. Pour cela, il convient donc d'abord de définir la norme d'intérêt sur l'espace des mesures. On considére **la norme en variation totale :**

$$\|\mu_1 - \mu_2\|_{\mathrm{VT}} = \sup_A |\mu_1(A) - \mu_2(A)| \tag{2.11}$$

Et voila le théorème important :

Théorème *Si la chaîne de Markov (X_n) est apériodique et positive au sens de Harris, alors :*

$$\lim_{n\to\infty} \| \int_{\chi} K^n(x,.)\mu(dx) - \pi(.)\|_{\mathrm{VT}} = 0 \tag{2.12}$$

pour toute loi initiale $\mu(.)$

Ce thérorème s'applique aussi pour des lois initiales qui sont des diracs. En particulier, on est sûr que quelque soit l'état initial de la chaine, la convergence a bien lieu.

Certains définissent même la récurrence au sens de Harris de cette manière : la convergence a lieu pour n'importe quel état initial au lieu d'être presque sûre. C'est ce théorème qui donne à la récurrence au sens de Harris toute son importance pratique.

Cependant ces résultats de convergence sont donnés sur $K^n(.;.)$. Il se trouve que ce n'est pas directement ce qui nous intéresse.

2.2.5 Théorèmes limites.

On considère les observations $(X_1,\ldots,X_n)$ d'une chaîne de Markov. On examine le comportement limite des sommes partielles :

$$S_n(h) = \frac{1}{n}\sum_{i=1}^{n} h(X_i) \tag{2.13}$$

Le théorème suivant est d'une importance capitale dans le domaine :

Théorème **Théorème ergodique.**
Si (X_n) a une mesure $\pi(.)$ σ-finie invariante, les deux assertions suivantes sont équivalentes :

- **(i)** *Si $f,g \in \mathcal{L}^1(\pi)$ avec $\int g(x)\pi(dx) \neq 0$, alors :*

$$\lim_{n\to+\infty} \frac{S_n(f)}{S_n(g)} = \frac{\int f(x)d\pi(x)}{\int g(x)d\pi(x)} \tag{2.14}$$

- **(ii)** *La chaîne de Markov (X_n) est récurrente au sens de Harris*

Il faut remarquer qu'en général ce qui nous interresse directement c'est le fait que (ii) implique (i)...

Voila ensuite une version du thérorème central limite. Il existe une version sous des conditions plus faibles , mais il se trouve qu'en pratique nous avons la condition de réversibilité présentée ici :

Définition *Une chaîne de Markov (X_n) est* **réversible** *si la loi de $X_{n+1}|X_{n+2} = x$ est la même que la loi de $X_{n+1}|X_n = x$.*

Et :

Théorème *Si (X_n) est apériodique, irréductible et réversible de loi invariante $\pi(.)$ le* **théorème central limite** *s'applique quand:*

$$0 < \gamma_g^2 = \mathbb{E}_\pi\left[g^2(X_0)\right] + 2\sum_{k=1}^{\infty}\mathbb{E}_\pi\left[g(X_0)g(X_k)\right] < \infty \qquad (2.15)$$

On a alors :

$$\frac{1}{\sqrt{N}}\left(\sum_{n=1}^{N}(g(X_n) - \mathbb{E}_\pi\left[g\right])\right) \xrightarrow{\mathcal{L}} \mathcal{N}(0,\gamma_g^2) \qquad (2.16)$$

Voila pour ces quelques rappels sur les chaines de Markov. L'objectif maintenant est de construire des algorithmes permettant de simuler les densités de processus ponctuels qui nous intéressent.

2.3 Metropolis Hastings Green (MHG).

Les méthodes de simulation par chaines 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 chaines de Markov dont la loi stationnaire est la loi objectif. Parmi les méthodes les plus connues, on retient *l'échantilloneur 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.

La partie 2.7 présente d'autres algorithmes qui sont couramment utilisés pour la simulation de processus ponctuels.

2.3.1 Présentation de l'algorithme.

L'algorithme a été introduit par Green en 1995 [15], 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. Pour plus de détails, on pourra se référer à [11] ou à [15].

On se donne:

1. un espace d'état $\mathcal{C}$ et une mesure $\pi(.)$ 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 $Q(.;.)$ 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 $Q(x;.)$ 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 $\pi(.)$ et Q, on a besoin d'une mesure symétrique $\xi(.;.)$ sur $\mathcal{C} \times \mathcal{C}$ telle que ξ domine $\pi(dx)Q(x,dy)$ et qu'il existe une dérivée au sens de Radon-Nikodym:

$$f(x,y) = \frac{\pi(dx)Q(x,dy)}{\xi(dx,dy)} \tag{2.17}$$

On appelle alors **rapport de Green** le rapport:

$$R = \frac{f(y,x)}{f(x,y)} \tag{2.18}$$

L'algorithme se présente alors de la manière suivante :

1. A partir de l'état courant x, on simule $y \sim Q(x,.)$.
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 refusé, on garde x comme nouvel état. Sinon, le nouvel état est donné par y.

2.3.2 Problématique du noyau de transition.

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 ξ, ce noyau n'influe pas du tout sur la mesure objectif $\pi(.)$. 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 :

Faisons deux hypothèses. Supposons d'abord que toutes les mesures soient dominées par une mesure $\lambda(.)$. 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 $h(.)$ est la densité de la mesure objectif $\pi(.)$ sous la mesure dominante λ. 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, et 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 $\pi(.)$. Proposer des états pertinents au sens de π 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, comme cela est expliqué dans [39] et [26] 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.

Le lecteur attentif remarquera toutefois, que dans le cadre de nos hypothèses simplificatrices la condition de réversiblité 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 le cadre général, c'est à dire sans les deux hypothèses trop simplificatrices ci-dessus, Guidici et Roberts proposent des méthodes dans [14] qui permettent d'assurer d'avoir un coefficient de Green à peu près égal à 1. Il semble toutefois qu'il manque encore un cadre mathématique formel pour justifier proprement ces heuristiques. On peut toutefois se réferrer à [4] , [10]pour approfondir ces questions.
Cette problématique revêt une importance toute particulière dans l'adaptation en recuit-simulé de cet algorithme. (Voir paragraphe 2.6).

Reste à voir les manières possibles de fabriquer de bons noyaux de proposition.

2.3.3 Les libertés accordées sur le noyau de proposition.

Commençons par présenter les conditions imposées au noyau:

Conditions imposées au noyau.

L'objectif est de fabriquer une chaîne de Markov apériodique, irréductible, de loi stationnaire $\pi(.)$. On appelle K le noyau de transition de la chaîne de Markov.

L'apériodicité 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 débarasser 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 $\pi(.)$. Pour cela, on impose une condition forte, c'est la condition dite d'"équilibre minutieux" [1]:

$$\int_A \int_B \pi(dx)K(x,dx') = \int_B \int_A \pi(dx')K(x',dx) \tag{2.19}$$

La condition (2.19) est une condition forte. L'algorithme de Green la vérifie naturellement: cette condition se traduit par la condition d'existence de la mesure symétrique ξ. En fait pour assurer cette condition, on peut se permettre quelques libertés.

Mélange de noyaux.

Dans son article [15], Green propose d'utiliser comme noyau de proposition Q un mélange de noyaux $q_m(x,.)$. 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 $q_m(x,dx')$. 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 dans [15] que pour avoir la condition (2.19), il suffit d'avoir la condition suivante pour chacun des noyaux q_m:

$$\begin{array}{c}\pi(dx)q_m(x,dx') \text{ admet une densité finie } f_m(x,x') \\ \text{par rapport à une mesure symétrique } \xi_m \text{ sur } \mathcal{C}\times\mathcal{C}\end{array} \tag{2.20}$$

1. *detailed balance* en anglais.

et de poser comme rapport de Green pour chacun des noyaux:

$$R_m(x,x') = \frac{f_m(x',x)}{f_m(x,x')} \tag{2.21}$$

La loi de Bernoulli qui permet d'accepter ou non la proposition prend alors pour paramètre:

$$\alpha_m(x,x') = \min\left\{1, \frac{f_m(x',x)}{f_m(x,x')}\right\} \tag{2.22}$$

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 $y \sim q_m(x,.)$.
3. Calculer le rapport de Green $R_m(x,x')$.
4. Accepter la proposition avec une probabilité $\alpha_m(x,x')$.

Nous présentons dans le chapître 4 des exemples de noyaux de transformations. Nous verrons alors comment assurer la condition (2.20).

Voila donc le cadre général de l'algorithme de Hasting Metropolis Green. Nous allons nous intéresser maintenant plus particulièrement à son application pour la simulation de processus ponctuels, c'est à dire pour $\mathcal{C} = N^f$.

2.4 MHG pour les processus ponctuels.

Dans [12], Geyer et Moller présentent un algorithme de Metropolis Hastings Green qui permet de simuler facilement des processus ponctuels. 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 "minimum vital" de l'algorithme: 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, ainsi que l'ergodicité géométrique.

2.4.1 Coeur de l'algorithme.

On se place donc sur $\mathcal{C} = N^f(\chi)$.

On considère un processus de Poisson de référence sur χ, simple et fini de loi de probabilité $\mu(.)$ et de mesure d'intensité λ. En fait, on se place sur $S \subseteq \chi$ de mesure $\lambda(S)$ finie. (Par exemple on se place sur un fermé borné de $\mathbb{R}^2$ avec comme mesure d'intensité la mesure de Lebesgue.)

On note donc $\mathcal{C} = N_S^f = \{\mathbf{x} \in N^f \text{ tq } \forall\, x \in \mathbf{x}\ , x \in S\}$

On veut simuler un processus ponctuel de densité $h(.)$ non normalisée par rapport à $\mu(.)$.

On rappelle que l'on note $\mathbf{x} = \{x_1, \ldots, x_{n(\mathbf{x})}\}$. 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 :

$$Q = p_b Q_b + p_d Q_d \tag{2.23}$$

avec les deux noyaux de naissance et de mort :

$$Q_b(\mathbf{x},A) = \int_{u\in S} \mathbf{1}_A(\mathbf{x}\cup u)\frac{\lambda(du)}{\lambda(S)} \tag{2.24}$$

$$Q_d(\mathbf{x},A) = \sum_{u\in\mathbf{x}} \mathbf{1}_A(\mathbf{x}\setminus u)\frac{1}{n(\mathbf{x})} \tag{2.25}$$

Pour prouver la stationnarité de la loi objectif , il faut prouver la condition (2.20). Pour cela, considérons la mesure ξ suivante (A et B sont des boréliens de χ) :

$$\xi(A\times B) = \int_{\mathcal{C}}\int_{u\in S} \mathbf{1}_A(\mathbf{x})\mathbf{1}_B(\mathbf{x}\cup u)\lambda(du)\mu(d\mathbf{x}) + \int_{\mathcal{C}} \mathbf{1}_A(\mathbf{x})\sum_{u\in\mathbf{x}} \mathbf{1}_B(\mathbf{x}\setminus u)\mu(d\mathbf{x}) \tag{2.26}$$

Il faut démontrer que cette mesure est symétrique. Les justifications théoriques des égalités suivantes ont été développées dans la partie 1.3.2:
On prend pour cela : $A_n = A\cap N_n^f$ et on a donc:

$$\begin{aligned}
\xi(A_n\times B_{n-1}) &= \frac{e^{-\lambda(S)}}{n!}\int_{\chi^n}\sum_{u\in S}\mathbf{1}_{A_n}(\mathbf{x})\mathbf{1}_{B_{n-1}}(\mathbf{x}\setminus u)\lambda^n(d\mathbf{x}) && \text{(2.27a)}\\
&= \frac{e^{-\lambda(S)}}{n!}\int_{\chi^n} n\mathbf{1}_{A_n}(\{x_1,\ldots,x_n\})\mathbf{1}_{B_{n-1}}(\{x_1,\ldots,x_n\})\lambda^n(d\mathbf{x}) && \text{(2.27b)}\\
&= \frac{e^{-\lambda(S)}}{(n-1)!}\int_{\chi^{n-1}}\int_S \mathbf{1}_{B_{n-1}}(\mathbf{y})\mathbf{1}_{A_n}(\mathbf{y}\cup u)\lambda^{n-1}(d\mathbf{y})\lambda(du) && \text{(2.27c)}\\
&= \xi(B_{n-1}\times A_n) && \text{(2.27d)}
\end{aligned}$$

Cela suffit pour prouver que ξ 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 cela il est intéressant de remarquer dans la succession d'égalité (2.27) 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 $\pi(.)Q(.;.)$ mesure non-nulle, il est de ξ mesure non nulle.

Il y a donc deux cas à considérer :

1. Si $\mathbf{y} = \mathbf{x}\cup u$, alors : La densité objectif π 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 ξ (cf. (2.26)) quant à elle donne :

   $$\xi(d\mathbf{x},d\mathbf{y}) = \mu(d\mathbf{x})\lambda(du)$$

   Ce qui permet de vérifier l'absolue continuité dans ce cas et d'obtenir la dérivée de Radon-Nikodym:

   $$\boxed{f(\mathbf{x},\mathbf{y}) = p_b\frac{h(\mathbf{x})}{\lambda(S)}} \tag{2.28}$$

2. L'autre cas consiste à considérer $\mathbf{y} = \mathbf{x}\setminus u$. On a alors, d'après la densité objectif et le noyau :

   $$\pi(d\mathbf{x})Q(\mathbf{x},d\mathbf{y}) = h(\mathbf{x})\mu(d\mathbf{x})p_d\frac{1}{n(\mathbf{x})}$$

   La mesure ξ donne:

   $$\xi(d\mathbf{x},d\mathbf{y}) = \mu(d\mathbf{x})$$

   Et on obtient l'absolue continuité et la dérivée :

   $$\boxed{f(\mathbf{x},\mathbf{y}) = p_d\frac{h(\mathbf{x})}{n(\mathbf{y})}} \tag{2.29}$$

Les équations (2.28) et (2.29) définissent donc la dérivée de $\pi(.)Q(.;.)$ par rapport à ξ. 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 : $\mathbf{y}$ peut être obtenu en ajoutant ou en retranchant un point à $\mathbf{x}$.

1. Dans le cas d'une naissance, $\mathbf{y} = \mathbf{x} \cup u$, le coefficient de Green a pour expression :

$$\boxed{R(\mathbf{x},\mathbf{y}) = \frac{f(\mathbf{y},\mathbf{x})}{f(\mathbf{x},\mathbf{y})} = \frac{p_d}{p_b}\frac{h(\mathbf{y})}{h(\mathbf{x})}\frac{\lambda(S)}{n(\mathbf{y})}} \tag{2.30}$$

   Où l'on peut remplacer $n(\mathbf{y})$ par $n(\mathbf{x}) + 1$.

2. Dans le cas d'une mort, $\mathbf{y} = \mathbf{x} \setminus u$, le coefficient de Green a pour expression :

$$\boxed{R(\mathbf{x},\mathbf{y}) = \frac{f(\mathbf{y},\mathbf{x})}{f(\mathbf{x},\mathbf{y})} = \frac{p_b}{p_d}\frac{h(y)}{h(x)}\frac{n(\mathbf{x})}{\lambda(S)}} \tag{2.31}$$

Les coefficients (2.30) et (2.31) permettent d'assurer la condition (2.19). On sait donc qu'un tel algorithme admet pour mesure stationnaire la version normalisée de $\pi(.)$.

Il reste que des conditions sont encore à vérifier: l'irréversibilité et la récurrence au sens de Harris.

L'apériodicité est assurée dès lors que l'on prend $p_b + p_d < 1$ 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.

2.4.2 Conditions de stabilité.

Ces conditions ont été apportées par Ruelle dans [34]. Ce sont des conditions 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 dans le paragraphe 2.4.1.

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 *Un processus ponctuel de densité non-normalisée h par rapport à un processus ponctuel de référence de loi $\mu(.)$ est* **stable au sens de Ruelle** *si il existe un réel $M \geq 1$ tel que :*

$$h(\mathbf{x}) \leq M^{n(\mathbf{x})} \qquad \forall \mathbf{x} \in N^f(\chi) \tag{2.32}$$

Il faut faire quelques commentaires sur cette propriété.
D'abord, un processus ponctuel peut être défini sur tout l'espace et être fini malgré tout (cf. partie 1.2).
Ensuite, 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 $\mathcal{C} = N_S^f$, puisque :

$$\int h(\mathbf{x})\mu(d\mathbf{x}) \leq \int M^{n((x)}\mu(d\mathbf{x}) = \sum_{k=0}^{\infty} \frac{M^k}{k!}\lambda(S)^k = e^{M\lambda S}$$

Par exemple le processus de Strauss avec $\gamma > 1$ ne vérifie pas cette condition.

Voila la seconde condition de stabilité que l'on définit dans le cadre formel de la partie 2.4.1:

Définition *Un processus ponctuel fini de densité non-normalisée $h(.)$ par rapport à un processus de Poisson de loi μ est* **stable** *s'il existe un réel M tel que :*

$$h(\mathbf{x} \cup u) \leq Mh(\mathbf{x}), \text{ pour tout } \mathbf{x} \in \mathcal{C} \text{ et } u \in S. \tag{2.33}$$

Cette condition implique la condition (2.32).
En utilisant les résultats de la partie 1.3.2, on peut reformuler cette condition 2.33 comme une condition de bornitude de l'intensité conditionnelle de Papangelou.

2.4.3 Propriétés.

Les propriétés de l'algorithme présenté dans la partie 2.4.1 tiennent en deux propositions : rappelons toutefois ce que l'on appelle ensemble borné sur $\mathcal{C}$:
Un sous ensemble C de $\mathcal{C}$ est **borné** si il existe $m \in \mathbb{N}$ tel que pour tout $\mathbf{x} \in C$, $n(\mathbf{x}) \leq m$.

Proposition *Si la densité non normalisée du processus ponctuel objectif satisfait la condition 2.33, alors l'algorithme 2.4.1 simule une chaîne de Markov irréductible, et tout ensemble borné de $\mathcal{C}$ est petit.*

Et la dernière propriété manquante est :

Proposition *Si la densité non normalisée $h(.)$ satisfait la condition (2.33), alors l'algorithme 2.4.1 simule une chaîne de Markov Harris-récurrente et géométriquement ergodique.*

Pour les preuves de ces deux propositions, on pourra se référer à [11]. L'algorithme de Metropolis Hastings Green avec des naissances et des morts a donc toutes les propriétés désirées:
- La chaîne de Markov (X_n) produite admet pour mesure stationnaire la mesure objectif $\pi(.)$ connue au facteur de normalisation près.
- La chaîne est irréductible donc $\pi(.)$ irréductible d'après le théorème 7.
- La chaîne est apériodique.
- La chaîne est Harris-récurrente.

Toutes ces propriétés permettent d'assurer la convergence de la chaîne vers la loi-objectif, et ce pour n'importe quelle initialisation. (C'est la contribution de la récurrence au sens de Harris). On vérifie donc le théorème ergodique.
De plus, la deuxième proposition assure une ergodicité géométrique, ce qui donne un résultat sur la vitesse de convergence des mesures en variation totale.

2.5 Discussion sur MHG.

Dans cette partie, on se propose de discuter, de manière assez générale, sur l'algorithme de Hasting Metropolis Green.
Il faut noter qu'une partie importante de cette discussion a déja été menée dans la partie 2.3.2. Rappelons qu'elle concernait l'importance du noyau de proposition Q. Pour l'instant ce noyau de proposition ne contient qu'un type de transformation : la naissance ou la mort. Il est immédiat que l'on peut rajouter d'autres transformations sans changer les propriétés de l'algorithme.

2.5.1 Détection de la convergence.

Une des grandes questions qui se pose dans l'utilisation d'algorithmes de Metropolis Hastings Green concerne la détection pratique de la convergence de la chaine.

C'est une question très délicate. Robert et Casella dans [31] proposent d'utiliser des statistiques du système. On considére un échantillon $(X_n, \ldots, X_n + k)$ de la chaine, et une famille de fonctions réelle $(h_i)_i$ sur $\mathcal{C}$. On regarde alors les estimateurs empiriques :

$$\tilde{h_i}_k = \frac{1}{k+1} \sum_{l=n}^{n+k} h_i(X_l) \tag{2.34}$$

Des méthodes présentées dans [4] et [36] travaillant sur ces estimateurs empiriques permettent "d'intuiter" la convergence. Quoiqu'il en soit ces méthodes restent heuristiques. L'approche la plus immédiate mais aussi la plus dangeureuse consiste à détecter la convergence de ces statistiques. Les publications citées ci-dessus apportent plus : elles donnent des critères de qualité du noyau de la chaine, comme par exemple une grandeur qui estime la "mélangeance " du noyau.

2.5.2 Méthode de réjection rejetée.

Cette méthode, présentée dans [16] permet de réduire l'autocorrélation des réalisations de la chaîne en évitant de refuser une proposition : si une proposition est refusée, Green et Mira proposent de faire une nouvelle proposition en tirant parti de l'information qu'apporte le premier rejet. Leur article montre comment faire une succession de propositions jusqu'a en obtenir une qui soit satisfaisante, et comment assurer la condition d'équilibre minutieux.

2.6 Recuit Simulé.

Il faut se référer à [21] pour avoir une justification de la convergence de l'algorithme de recuit simulé dans le cas de processus ponctuels. On rappelle que l'idée générale du recuit simulé consiste à utiliser un échantilloneur de Metropolis Hastings en utilisant une densité variable avec le temps :

$$h(\mathbf{x})^{\frac{1}{T_t}} \tag{2.35}$$

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 β 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 :

$$\beta\,(h(\mathbf{x}))^{\frac{1}{T_t}} \tag{2.36}$$

2.7 Autres algorithmes utilisés.

2.7.1 Algorithme de naissance-mort.

Ce sont les plus anciens algorithmes utilisés pour simuler des processus ponctuels (voir [27]). Plusieurs études (voir par exemple [20]) ont montré qu'il est préférable d'utiliser un algorithme HGM.

2.7.2 Simulations exactes.

Ces méthodes sont en cours d'exploration. Elles ont été introduites dans [29]. Elles permettent de simuler des chaines de Markov qui atteignent la stationnarité en un temps fini, que l'on peut détecter.

Elles nécessitent cependant d'obéir à des contraintes très fortes sur la densité du processus que l'on veut simuler.

Chapitre 3

Modèle utilisé

3.1 Rappel des objectifs.

On rappelle que l'objectif de ce travail est de détecter des caricatures de bâtiments sur des Modèles Numériques d'Elévation.

3.2 Les données.

Le Modèle Numérique d'Elévation présenté en figure 2 appelle quelques commentaires :

– Les niveaux de gris représentent une réalité physique : l'altitude du point considéré.
– Le bruit n'est pas un bruit classique en traitement de l'image : il s'agit plus d'un bruit géométrique : (il y a des lampadaires, des voitures etc...)
– Il faut également rappeler que dans les zones urbaines, la densité des bâtiments pose des problèmes : si sur une image, un bâtiment particulièrement élevé masque un autre bâtiment, la reconstruction de ce dernier est impossible! Une telle zone masquée s'appelle une zone d'occlusion.
– Les bâtiments obéissent à certaines contraintes : en particulier, il sont alignés dans le sens de la longueur.

3.2.1 Approche sur-pixèlique.

Les méthodes que nous mettons en place ici sont des approches par objets : on ne cherche pas à attribuer à chaque pixel d'une image une étiquette caractérisant la présence d'un bâtiment, mais à placer directement les bâtiments dans un espace tri ou bi-dimensionnel.

Dans le cadre de la reconnaissance d'objets en zone urbaine, il est possible de s'intéresser à différents types fd'objet : les bâtiments, les routes, les ponts etc... mais aussi à des structures d'objets : une rue, un quartier, etc ...

3.2.2 Utilisation des processus ponctuels.

Les processus ponctuels marqués s'adaptent naturellement à nos objectifs :

Les objets d'intérêts.

On veut reconnaitre des caricatures de bâtiment sur le Modèle Numérique d'Elévation. Cela sous-entend qu'il faut commencer par donner un modèle simple de bâtiments.

Idéalement, on voudrait travailler avec trois dimensions, en utilisant par exemple comme modèle de bâtiment un parallepipède surmonté d'un prisme en guise de toit.

Pour l'instant on se cantonne à deux dimensions, et l'on prend comme modèle de bâtiment un simple rectangle. En fait, ce rectangle est la projection du parallepipède précédent. Au vu du type de données dont on dispose, il parait plus simple d'essayer de reconnaitre d'abord la trace au sol des bâtiments.

Pour ce faire on considère un pavé de $\chi = \mathbb{R}^2$, par exemple :$S = [0,X_{\max}] \times [0,Y_{\max}]$. Les points du processus sont donc des coordonnées bi-dimensionnelles, contenues dans ce pavé.
Ces points représentent les centres de la trace au sol des bâtiments. On ajoute ensuite un espace de marques $\mathcal{K}$, de façon à pouvoir décrire des rectangles. Les marques que l'on considère sont au nombre de trois :

- une longeur $L \in [L_{\min},L_{\max}]$.
- une largeur $l \in [l_{\min},l_{\max}]$
- un angle d'orientation: $\theta \in [0,\pi]$

Le processus ponctuel de référence.

Après avoir défini l'espace d'intérêt $S \times \mathcal{K}$, il faut définir le processus ponctuel de référence. Rappelons que ce processus est un processus ponctuel marqué. Il doit donc être à la fois processus ponctuel sur $S \times \mathcal{K}$ et sur S uniquement (cf chapitre 1).

Le processus ponctuel de référence que l'on considère est un processus ponctuel marqué simple et fini de Poisson sur $S \times \mathcal{K}$ avec comme mesure d'intensité la mesure produit de la mesure de Lebesgue λ sur $\mathbb{R}^2$ par la loi de probabilité des marques sur $\mathcal{K}$: $P_m(.)$. On appelle cette mesure d'intensité : $\nu(.) = (\lambda \times P_m)(.)$.

On prend volontairement $P_m(.)$ différent de la mesure de Lebesgue, de manière à ne pas confondre toutes les mesures de Lebesgue dans les démonstrations ultérieures, même si en pratique $\mathbf{P}_m$ sera uniforme. En fait, puisqu'il aura toujours le paramètre β, cela revient à ne pas poser de contrainte sur la mesure sur $\mathcal{K}$ autre que l'intégrabilité.

Pour finir, on note $\mu(.)$ la loi du processus ponctuel de référence.

Densité du processus d'intérêt.

L'objectif de la suite va être de construire une densité de processus qui réponde à certains critères. Ces critères sont de trois ordres :

- La densité doit répondre à des conditions mathématiques pour que le processus puisse être simulable par un algorithme de Metropolis Hastings Green. En particulier elle doit vérifier la condition de stabilité (2.33).

- Elle doit aussi répondre à des critères informatiques : l'algorithme proposé est assez long, et l'est d'autant plus que les rapports de densité $h(\mathbf{y})/h(\mathbf{x})$ sont complexes à calculer.

- Elle doit répondre à nos objectifs : on veut reconnaitre des bâtiments sur une image de données que constitue le modèle d'élévation tout en tenant compte de contraintes connues sur l'agencement des bâtiments : deux bâtiments ne se recouvrent pas, les bâtiments ont des chances d'être alignés etc...

Pour répondre à ces critères, on prend une densité de la forme :

$$\pi(d\mathbf{x}) = \alpha\, \beta^{n(\mathbf{x})}\, h_d(\mathbf{x})\, h_p(\mathbf{x})\, \mu(d\mathbf{x}) \tag{3.1}$$

où h_d représente la partie de la densité qui tient compte de l'image de données tandis que h_p sert à introduire les contraintes sur les objets.
Les paramètres α et β sont respectivement la constante de normalisation que l'on n'a pas besoin de connaitre et le paramètre d'échelle qui sert à “adapter” l'intensité du processus de référence.

3.2.3 Analyse non bayésienne.

Evidemment, les termes h_d et h_p de la densité (3.1) rappellent les termes a priori et de loi conditionnelle de l'analyse bayésienne. Malheureusement dans notre cas, on ne peut appliquer une analyse bayésienne à proprement parler. En effet il faudrait proposer un modèle a priori h_p qui propose des réalisations $\mathbf{x}$ du processus ponctuel. Ce modéle à priori introduirait par exemple les contraintes de non-intersection ou d'alignement des bâtiments.

Il faudrait ensuite un modèle de bruit défini par la densité conditionnelle : $\mathbf{y} \to h_d(\mathbf{y}|\mathbf{x})$. A partir de $\mathbf{x}$ on propose un modèle pour générer $\mathbf{y}$. On inverserait ensuite ce modèle pour définir une vraisemblance :$\mathbf{x} \to h_d(\mathbf{y}|\mathbf{x})$ qui permette pour un $\mathbf{y}$ donné de trouver le "meilleur $\mathbf{x}$"

Cela pose des problèmes dans notre cas :
- D'abord notre objet d'intérêt $\mathbf{x}$ est un objet défini dans un cadre continu, tandis que les données sont discrètes.
- Ensuite et surtout, le modèle de bruit ne serait pas simple puisqu'il s'agit d'un bruit géométrique. Pas question donc d'espérer pouvoir faire une inversion pour obtenir la vraisemblance de manière explicite.

Rue dans [32] s'en sort malgré tout parce que le bruit qu'il considère est un bruit de convolution, et non un bruit géomètrique. L'exemple qu'il utilise est d'ailleurs très intéressant puisqu'il réussit à faire de l'inférence bayésienne sur des processus ponctuels avec des données discrètes que sont les images.

Pour notre cas, il est bien plus simple d'introduire une "vraisemblance fictive" : c'est la fonction $h_d(\mathbf{x})$. Elle permet de savoir si une configuration $\mathbf{x}$ de bâtiments est probable ou non à partir du modèle numérique d'élévation. Nous verrons plus loin comment nous avons défini cette vraisemblance fictive.

L'avantage de procéder ainsi est que l'on peut définir un modèle que l'on appelle malgré tout "modèle a priori" et "un modèle d'attache aux données" qui n'est pas la vraisemblance. La difficulté amenée par l'utilisation d'une vraisemblance fabriquée empiriquement tient dans la pondération des deux influences h_d et h_p dans la densité.

Il reste un dernier problème à règler : le choix de l'estimateur.

3.2.4 Le choix de l'estimateur.

On a défini les objets d'intérêt et justifié le type de densité utilisée. De manière idéale, si l'on construit correctement le modèle à priori et le terme d'attache au données , on obtient donc une densité de probabilité dont les réalisations les plus probables correspondent à des interprétations très plausibles du modèle numérique d'élévation.

Se pose alors un problème classique : comment à partir de ces réalisations très probables obtenir l'interprétation du modèle numérique d'élévation qui correspond le mieux à nos objectifs.

Le maximum a posteriori.

La méthode la plus simple est de prendre comme estimateur le Maximum a Posteriori (cf, [9], [20] et [36]). Du point de vue algorithmique, il "suffit" de faire un recuit simulé sur la densité proposée pour obtenir le Maximum a Posteriori. En fait, comme on ne peut jamais faire un vrai recuit simulé, on obtient juste une réalisation très probable.
Dans notre cadre non-bayésien, cela est pertinent dans le sens où nous allons définir une densité complexe par plusieurs termes. Cependant, cet estimateur pose deux problèmes :
- le premier problème est un problème classique de choix de mode. Il peut paraitre plus pertinent de retenir l'information contenue dans deux modes plutôt que de choisir arbitrairement l'un de ces deux modes. Nous développerons cette idée plus en détail plus loin.

- le deuxième probléme vient de la mesure de référence $\mu(.)$ que l'on utilise :
  Le paramètre β, qui sert à dimensionner l'intensité du processus peut être considéré comme une partie de cette mesure de référence. Ce paramètre influe fortement sur le Maximum a Posteriori, puique c'est lui qui influe sur l'espérance du nombre de points de cette mesure de référence.
  Or on ne connait pas du tout le nombre d'objets que l'on veut détecter !! La solution dans ce cas est d'augmenter l'influence du terme d'attache aux données, ce qui pose quelques problèmes de dynamique.

Les alternatives possibles.

Il n'y en a pas beaucoup dans le cadre d'utilisation de processus ponctuels. Celle qui semble être la plus efficace est celle proposée par Rue dans [32] et [33] à partir des travaux de Baddeley [1].
Elle repose sur la définition d'une distance entre deux images binaires. Cette approche est particulièrement adaptée à notre problème. Nous la présentons dans la partie 3.5.2.

3.3 Le modèle a priori.

3.3.1 Ses enjeux.

Normalement, le terme d'attache aux données devrait suffire pour détecter proprement les objets en question. Il y a trois raisons majeures qui sous tendent l'idée d'ajouter un modèle a priori :

- la première vient d'une idée de pertinence : on pondère les configurations détectées par le terme d'attache aux données par la connaissance que l'on à a priori de la structure d'une ville. Il n'est pas du tout pertinent, par exemple de proposer une configuration où des maisons s'intersectent. En fait on va bien plus loin : on désire faire intervenir dans cette densité a priori au minimum deux contraintes :
  - une de non-intersection entre les rectangles représentant les maisons
  - une d'alignement : dans une zone urbaine, les maisons sont souvent alignées avec d'autres maisons.

  A ces contraintes s'ajoutent d'autres plus générales : on peut émettre des postulats sur l'homogénéité des quartiers par exemple : si deux maisons sont proches, il est probable qu'elles se ressemblent et qu'elles aient même largeur et même longueur etc...
- la deuxième raison est une extrapolation de la première : si l'on règle bien l'influence du terme a priori et celle du terme d'attache aux données, on doit voir apparaitre un phénomène de régularisation.
- enfin, il reste le problème de la dynamique: le modèle a priori favorise l'exploration rapide de modes.

3.3.2 Les approches classiques et leurs limites.

Les processus de Gibbs.

L'idée est d'introduire des relations binaires entre les points d'une configuration $\mathbf{x}$. Par exemple, on peut définir la relation $\sim$ d'intersection:

Pour $\mathcal{C} = N^f_{S\times\mathcal{K}}$:
On prend $\mathbf{x} \in \mathcal{C}$ et on dit de x_i et $x_j \in \mathbf{x}$ qu'ils sont en relation si les deux rectangles s'intersectent.
On note alors : $x_i \sim x_j$.

Pour l'instant, on rejoint tout à fait le cadre des processus ponctuels de Markov et on retrouve des notions abordées dans l'exemple du processus de Strauss. La seule différence tient au fait que le processus ponctuel est marqué et que la relation dépend des marques.

On définit alors la densité suivante :

$$h_p(\mathbf{x}) = \exp\left(\sum_{\substack{x_i,x_j\in\mathbf{x}\\ x_i\sim x_j\\ i<j}} U_{\text{inters}}(x_i,x_j)\right)$$

Il faut faire quelques remarques :

1. L'expression de la densité est une expression énergétique, qui rappelle l'énergie d'interaction entre deux particules dans le cadre de la physique statistique: U_{inters} est appelé **potentiel d'interaction**.
2. Si l'on pose $U_{\text{inters}}(x_i,x_j) = U_{\text{inters}}$, c'est à dire un potentiel constant, l'expression de la densité devient:
$$\exp\left(U_{\text{inters}}\ \text{card}\left\{x_i \in \mathbf{x}\ \exists\ x_j\ i<j\ x_i \sim x_j\right\}\right)$$
On peut aussi reformuler cette densité sous la forme d'Hammersley-Clifford (cf paragraphe 1.9) puisque ce type de processus appartient à la classe des processus de Markov.

3. Si ce potentiel constant est négatif, on pénalise les intersections. S'il est positif, on les favorise.

4. Il s'agit d'une forme de processus de Strauss, puisqu'il suffit de poser : $U_{\text{inters}}(x_i,x_j) = \log(\gamma)$.

Constructions des potentiels.

Faisons d'autres remarques plus avancées :

le cas $U_{\text{inters}} > 0$ pose problème, puisque l'on sait qu'il n'est pas intégrable par analogie avec le processus de Strauss. Heureusement dans le cas de l'intersection, nous cherchons un cadre répulsif, mais pour définir des interactions attractives, nous allons rencontrer des problèmes.

Il est avantageux d'utiliser un potentiel non constant: il est clair que le cas où deux maisons qui s'intersectent beaucoup nous plait beaucoup moins que le cas ou les deux maisons s'intersectent un peu.
Or pour la dynamique de l'algorithme, il est important que cette différence soit prise en compte. D'où l'idée d'utiliser un potentiel variable, comme par exemple :

$$U_{\text{inters}}(x_i,x_j) = U_{\text{int}}\frac{\mathcal{A}(x_i,x_j)}{\mathcal{A}_{\max}}$$

où $\mathcal{A}(x_i,x_j)$ représente l'aire commune aux deux rectangles, et $\mathcal{A}_{\max}$ représente l'aire maximale, soit $l_{\max}L_{\max}$. La normalisation intervient pour que U_{int} soit sans dimension : il règle alors l'importance de la relation d'intersection par rapport à d'éventuelles autres relations binaires entre les maisons.

On peut aller plus loin encore. Imaginons que l'on remplace la relation d'intersection par une relation de proximité $\asymp$. On peut faire intervenir un potentiel sur les objets mis en relation au sens de cette relation. Posons :

$$\tilde{d}(x_i,x_j) = \frac{\left(l(x_i)-l(x_j)\right)^2 + \left(L(x_i)-L(x_j)\right)^2}{\left(l_{\max}-l_{\min}\right)^2 + \left(L_{\max}-L_{\min}\right)^2}$$

$\tilde{d}$ représente une mesure de ressemblance entre deux maisons. C'est une grandeur qui appartient à [0,1].
Introduison une fonction $f : [0,1] \to [-1,1]$ strictement décroissante, telle que $f(0) = 1$ et $f(1) = -1$. Considérons alors le potentiel mixte suivant :

$$U_{\text{proxi}}(x_i,x_j) = U_{\text{prx}}f(\tilde{d}(x_i,x_j))$$

Ce potentiel favorise la ressemblance de deux maisons qui sont proches et pénalise l'hétérogénéité. La fonction f doit être choisie sous le compromis suivant :

- Si elle varie vite, on augmente la précision du maximum du mode correspondant.
- Si elle varie doucement, on augmente la garantie de parcourir tous les cas possibles, donc de parcourir tous les modes.

Si on fait intervenir les deux relations $\sim$ et $\asymp$, on va pouvoir jouer au travers de U_{int} et U_{prx} sur l'importance relative de ces deux relations.

Les limites d'un tel modèle.

La première limite est importante puisqu'elle est d'ordre mathématique : il faut assurer l'intégrabilité du processus.

Prenons le cas de la relation $\sim$ avec un potentiel Uinters > 0 attractif et constant. La première condition de Ruelle (2.32) n'est pas vérifiée puisque, dans le pire des cas, la configuration $\mathbf{x}$ contient des points qui s'intersectent tous deux à deux et que dans ce cas, la densité vaut alors :

$$\beta^{n(\mathbf{x})} h_p(\mathbf{x}) = \beta^{n(\mathbf{x})} U_{\text{inters}}^{\frac{n(\mathbf{x})(n(\mathbf{x})-1)}{2}}$$

Pour rendre un tel processus intégrable, il y a trois méthodes classiques dont deux consistent à limiter le nombre de points :

1. on ajoute un terme **Hard core**. On interdit à deux points d'être trop proches.
   La limitation du nombre de points est alors géométrique.

2. on rajoute directement une contrainte en multipliant cette densité par: $\mathbf{1}(n(\mathbf{x}) \leq n_0)$.

3. on remplace dans l'expression de Hammersey Clifford de cette densité le nombre de paires en relation $s(\mathbf{x})$ par $\min\{s(\mathbf{x}),c\}$ où c est un paramètre limitant. Il faut remarquer qu'alors le processus n'est plus de Markov.

Cela appelle trois réflexions :

1. Il serait mieux de tenir compte du fait que le facteur limitant du nombre de points vient de la mesure du processus de référence, et en particulier de l'intensité $\beta\lambda(.)$. Or cette influence est en $\exp(n(\mathbf{x}) \log \beta)$

2. Il parait difficile d'utiliser une des deux astuces ci-dessus et de pouvoir règler les potentiels de différents types d'intersection de façon fine : les phénomènes combinatoires mis en jeu sont importants.

3. Si l'on veut ajouter des configurations n-aires, le problème va devenir complexe : par exemple, si l'on veut considérer des relations d'alignement entre trois maisons, et que l'on veut les favoriser, comme la combinatoire de ces relations ternaires est d'un ordre strictement supérieur à la combinatoire des relations binaires, les phénomènes ternaires vont effacer l'influence des relations binaires, tout comme les relations binaires effacaient l'influence de l'intensité du processus.

Ces réflexions nous ont conduit à introduire le formalisme et le modèle suivant :

3.3.3 Le formalisme adopté.

Définition des relations.

On rappelle que l'on se place sur $\chi = \mathbb{R}^2$. On fait on se restreint à $S \subseteq \chi$, borné. En l'occurence, on prend $S = [0,X_{\max}] \times [0,Y_{\max}]$.

On ajoute ensuite un espace de marques $\mathcal{K}$, pavé borné de $\mathbb{R}^3$: $\mathcal{K} = [l_{\min},l_{\max}] \times [L_{\min},L_{\max}] \times [0,\pi]$

On considére l'ensemble des configurations finies sur $S \times \mathcal{K}$: $\mathcal{C} = N^f_{S\times\mathcal{K}}$.

On définit ensuite la mesure $\nu(.) = (\lambda \times \mathbf{P}_m)(.)$ sur $S \times \mathcal{K}$ où $\lambda(.)$ est la mesure de Lesbesgue, tandis que $\mathbf{P}_m(.)$ est une mesure de probabilité sur $\mathcal{K}$.

Cette mesure définit l'intensité d'un processus ponctuel marqué X. Ce processus est un processus ponctuel simple, fini de Poisson sur $\mathcal{C} = N^f_{S\times\mathcal{K}}$. Le processus des points de X, est aussi un processus sur N^f_S, d'intensité λ.

Par commodité, dans cette partie, on note U l'espace d'état : c'est à dire $U = S \times \mathcal{K}$.

Définition *On appelle* **relation simple** *une application R_s de $\mathcal{C} = N^f_U$ dans l'ensemble des parties de N^f_U noté $\mathcal{P}(N^f_U)$:*

$$\begin{array}{rlcl} R_s : & N^f_U & \rightarrow & \mathcal{P}(N^f_U) \\ & \mathbf{x} = \{u_1,\ldots,u_{n(\mathbf{x})}\} & \rightarrow & R_s(\mathbf{x}) \end{array} \tag{3.2}$$

telle que :

$$\forall r \in R_s(\mathbf{x}) \qquad r \subseteq \mathbf{x} \tag{3.3}$$

Pour une relation R et une configuration $\mathbf{x}$, on appelle les éléments r de $R(\mathbf{x})$ des **cliques**. Les cliques sont des configurations.

Exemple 1 : On définit R_{s1} de la manière suivante :
Pour $\mathbf{x} = \{u_1,\ldots,u_n\}$:

$$R_{s1}(\mathbf{x}) = \left\{ \{u_i,u_j\} \quad \text{tq.} \quad \left| \begin{array}{l} i \neq j \\ d(u_i,u_j) < d_{\max} \end{array} \right. \right\}$$

Où d est une distance sur U, par exemple la distance enclidienne sur S, et $d_{\max}$ est un réel strictement positif. R_{s1} est un exemple particulier puisque:

$$\forall\, \mathbf{x} \in N^f_U \quad \forall r \in R_{s1}(\mathbf{x}) \qquad \mathrm{card}(r) = 2$$

On aura remarqué que cette relation R_{s1} décrit les paires de points mis en relation au sens de la relation de voisinage de proximité.

Exemple 2 : On définit R_{s2} comme suit : pour $\mathbf{x} = \{u'_1,\ldots,u'_n\}$,

$$R_{s2}(\mathbf{x}) = \left\{ \{u_1,\ldots,u_d\} \subseteq \mathbf{x} \text{ tq } \begin{array}{l} \forall(k,l) \in \{1,\ldots,d\} \\ \exists(i_1,\ldots,i_p) \in \{1,\ldots,d\} \end{array} \left| \begin{array}{c} \{u_k,u_{i_1}\} \in R_{s1}(\mathbf{x}) \\ \{u_{i_1},u_{i_2}\} \in R_{s1}(\mathbf{x}) \\ \vdots \\ \{u_{i_p},u_l\} \in R_{s1}(\mathbf{x}) \end{array} \right. \right\}$$

Et donc si $r \in R_{s2}(\mathbf{x})$, la clique r décrit une des parties connexes de $\mathbf{x}$ au sens de la relation de proximité.

Voila pour les relations simples. Le problème est qu'elles ne suffisent pas pour décrire des interactions d'un type un peu plus complexe comme les alignements. En effet, si l'on considére trois maisons ou trois points alignés, celui du milieu joue un rôle différent des deux autres.
Cette idée de rôle différent est très intéressante puisqu'elle permet de considérer des interactions non-symétriques.

Définition *On appelle* **relation étendue** *R toute application de $\mathcal{C} = N_U^f$ vers $\mathcal{P}\left((N_U^f)^d\right)$ où d est un entier strictement positif :*

$$R: \begin{array}{ccc} N_U^f & \rightarrow & \mathcal{P}\left((N_U^f)^d\right) \\ \mathbf{x} = \{u_1, \ldots, u_{n(\mathbf{x})}\} & \rightarrow & R(\mathbf{x}) \end{array} \tag{3.4}$$

telle que pour tout $r \in R(\mathbf{x})$, on puisse écrire la **clique** *r sous la forme :*

$$r = (r_1, \ldots, r_d) \text{ où, pour tout } i, \quad r_i \subset \mathbf{x} \tag{3.5}$$

Une définition équivalente consiste à remplacer la condition (3.5) par :

$$R(\mathbf{x}) \in R_{s1}(\mathbf{x}) \times \cdots \times R_{sd}(\mathbf{x}) \tag{3.6}$$

Où $R_{s1}, \ldots, R_{sd}$ sont des relations simples. Cette définition n'est pas meilleure mais permet de comprendre les possibilités qu'offrent les relations étendues. Le formalisme ci-dessus peut paraitre très lourd : par la suite, on verra qu'il autorise bien des libertés surtout dans l'écriture de densités de probabilité.

Il faut également remarquer que les relations simples sont un cas particulier de relations étendues ($d = 1$).

Exemple 3 : On définit R_3 de la manière suivante :

$$R_3(\mathbf{x}) = \{(\{u_i\},\{u_j,u_k\}) \text{ tels que } 3.8\} \tag{3.7}$$

Où la condition (3.8) est donnée par :

$$\begin{cases} d(u_i,u_j) < d_{\max} \\ d(u_i,u_k) < d_{\max} \\ \|\overrightarrow{u_i u_j} + \overrightarrow{u_i u_k}\| \leq \epsilon \end{cases} \tag{3.8}$$

Les vecteurs représentent les vecteurs de points dans $S = [0,X_{\max}] \times [0,Y_{\max}]$. Cette condition impose deux conditions de proximité, une condition d'alignement, et une condition d'équidistance. Cette relation est l'une des relations que nous avons mises en oeuvre informatiquement. Un des points interressants de sa définition est que l'on peut remplacer la condition (3.8) par la condition équivalente (3.9) suivante :

$$\begin{cases} \{u_i,u_j\} \in R_{s1}(\mathbf{x}) \\ \{u_i,u_l\} \in R_{s1}(\mathbf{x}) \\ \|\overrightarrow{u_i u_j} + \overrightarrow{u_i u_k}\| \leq \epsilon \end{cases} \tag{3.9}$$

avec R_{s1} la relation simple définie dans l'exemple 1. Du point de vue informatique cela signifie que l'on ne cherche les éléments de la relation R_3 qu'à partir des éléments de la relation précédemment définie, et non à partir de toute la configuration $\mathbf{x}$.

Pour conclure cette définition, il faut remarquer que notre définition des relations étendues propose d rôles différents. Pour une relation étendue, ce nombre d est fixé. Lorsque nous en aurons besoin dans la suite, nous le noterons $\check{d}_R$.

Fonctions immédiates sur les relations.

On note $\mathcal{R}$ l'ensemble des relations simples et étendues considérées. Dans nos exemples, $\mathcal{R}$ vaut donc $\mathcal{R} = \{R_{s1},R_{s2},R_3\}$
Pour commencer, définissons la fonction cardinal : pour $R \in \mathcal{R}$ et $\mathbf{x} \in N_U^f$ on note:

$$p(R,\mathbf{x}) = \text{card}(R(\mathbf{x})) \tag{3.10}$$

Si dans une configuration $\mathbf{x}$ donnée il y a trois alignements, c'est à dire 3 cliques, alors $p(R_3,\mathbf{x}) = 3$.

On continue avec une fonction qui se révelera pratique pour la suite.

Définition *Pour $R \in \mathcal{R}$, $\mathbf{x} \in N_U^f$, on note ϕ la fonction qui a toute clique $r \in R(\mathbf{x})$ associe le sous ensemble de $\mathbf{x}$:*

$$r \in R(\mathbf{x}) \qquad r = (r_1, \ldots, r_{\check{d}_R}) \qquad \phi(r) = r_1 \cup \cdots \cup r_{\check{d}_R} \subseteq \mathbf{x}$$

Cette fonction permet de savoir si un u_i appartient à une clique d'une relation sans se préocuper de son rôle.
Par exemple, dans le cadre de R_3, si $r = (\{u_2\},\{u_3,u_5\})$, $\phi(r) = \{u_2,u_3,u_5\}$. Cela peut paraitre étrange de s'affranchir de la séparation des rôles joués par les points de la configuration dans une relation alors que nous l'avons construite sans cela. En fait, cela permet de définir une notion de voisinage global :

Définition : *On appelle* **voisinage d'une configuration** *la fonction $V : \mathcal{R} \times N_U^f \to N_U^f$ telle que :*

$$V(R,\mathbf{x}) = \{u \in \mathbf{x} \ tq \ \exists\, r \in R(\mathbf{x}) \quad u \in \phi(r)\}$$

On note alors:

$$n_V(R,x) = card(V(R,x)) \in \{0, \ldots, n(\mathbf{x})\}$$

La grandeur n_V représente donc le nombre de points mis en jeu dans au moins une clique d'une configuration. On définit maintenant le voisinage d'un point dans une relation :

Définition *On appelle* **voisinage d'un point** *la fonction $V : \mathcal{R} \times N_U^f \times U \to N_U^f$ telle que : pour toute relation $R \in \mathcal{R}$, toute configuration $\mathbf{x} \in N_U^f$, tout point $u \in \mathbf{x}$,*

$$V(R,\mathbf{x},u) = \left\{ v \in \mathbf{x} \ \ tq \quad \exists\, r \in R(\mathbf{x}) \ \middle| \ \begin{array}{l} v \neq u \\ u \in \phi(r) \\ v \in \phi(r) \end{array} \right\} \subseteq V(R,\mathbf{x}) \qquad (3.11)$$

On note alors :

$$n_V(R,\mathbf{x},u) = card(V(R,\mathbf{x},u)) \qquad (3.12)$$

Autres fonctions sur les relations.

On regarde ensuite une classe particulière de relations

Définition *On note $\mathcal{R}_c$ le sous ensemble de $\mathcal{R}$ des relations telles que pour toute relation $R \in \mathcal{R}_c$ il existe un entier $c \in \mathbb{N}$ tel que :*
pour tout $\mathbf{x} \in N_U^f$, et toute clique $r \in R(\mathbf{x})$:

$$card(\phi(r)) = c \qquad (3.13)$$

Une telle relation est dite de cardinalité constante, et l'entier $c(R)$ est appelé **cardinalité de la relation** *R.*

Par exemple la relation simple R_{s1} est de cardinalité constante et $c(R_{s1}) = 2$. De même $c(R_3) = 3$. Par contre la relation R_{s2} n'est pas de cardinalité constante. L'idée sous jacente est de pouvoir parler des problèmes combinatoires mis en évidence par le processus de Strauss en terme de **cardinalité**.

Exemple : Ecrivons la densité du processus de Strauss en utilisant le cadre formel défini ci-dessus. On obtient :

$$h(\mathbf{x}) = \beta^{n(\mathbf{x})}\gamma^{p(R_{s1},\mathbf{x})} \tag{3.14}$$

où $p(R,\mathbf{x}) = \text{card}(R(\mathbf{x}))$. On l'a vu, une telle densité n'est pas intégrable, pour $\gamma > 1$. L'intensité conditionnelle de Papangelou s'ecrit :

$$\lambda_h(u;\mathbf{x}) = \frac{h(\mathbf{x} \cup u)}{h(\mathbf{x})} = \beta\gamma^{n_V(R_{s1},\mathbf{x},u)} \tag{3.15}$$

Si $\gamma > 1$, dans le pire cas où tous les points sont suffisamment proches, on obtient :

$$\lambda_h(u;\tilde{\mathbf{x}}) = \beta\gamma^{n(\tilde{\mathbf{x}})} \quad \text{et} \quad \begin{aligned} h(\tilde{\mathbf{x}}) &= \beta^{n(\tilde{\mathbf{x}})}\gamma^{C^2_{n(m\tilde{b}fx)}} \\ &= \beta^{n(\tilde{\mathbf{x}})}\gamma^{\frac{n(\tilde{\mathbf{x}})(n(\tilde{\mathbf{x}})-1)}{2}} \end{aligned} \tag{3.16}$$

Considérons maintenant une nouvelle relation simple R_{s3} de cardinalité constante valant $c(R_{s3}) = 3$ définie de la manière suivante : pour $\mathbf{x} = \{u_1, \ldots u_{n(\mathbf{x})}\}$.

$$\{u_i,u_j,u_k\} \in R_{s3}(\mathbf{x}) \quad \text{ssi} \begin{cases} \{u_i,u_j\} \in R_{s1}(\mathbf{x}) \\ \{u_j,u_k\} \in R_{s1}(\mathbf{x}) \\ \{u_k,u_i\} \in R_{s1}(\mathbf{x}) \end{cases} \tag{3.17}$$

Cette relation permet de travailler sur les ensembles de trois points suffisamment proches. Supposons que l'on introduise l'équivalent d'une densité de Gibbs sur cette relation (qui met en jeu des interactions de troisième ordre) et que l'on cherche à la favoriser. On utilise donc un paramètre γ_b et l'on écrit:

$$h_b(\mathbf{x}) = \beta^{n(\mathbf{x})}\gamma_b^{p(R_{s3},\mathbf{x})} \tag{3.18}$$

Pour l'intensité conditionnelle, il vient :

$$\lambda_{h_b}(u;\mathbf{x}) = \beta\gamma_b^{n_V(R_{s3},\mathbf{x}\cup u,u)} \tag{3.19}$$

Considérons à nouveau les configurations $\tilde{\mathbf{x}}$ de pire cas, c'est à dire lorsque tous les points sont proches :

$$\lambda_{h_b}(u;\tilde{\mathbf{x}}) = \beta\gamma_b^{\frac{n(\tilde{\mathbf{x}})(n(\tilde{\mathbf{x}})-1)}{2}} \quad \text{et} \quad \begin{aligned} h_b(\tilde{\mathbf{x}}) &= \beta^{n(\tilde{\mathbf{x}})}\gamma^{C^3_{n(\tilde{\mathbf{x}})}} \\ &= \beta^{n(\tilde{\mathbf{x}})}\gamma^{\frac{n(\tilde{\mathbf{x}})(n(\tilde{\mathbf{x}})-1)(n(\tilde{\mathbf{x}})-2)}{6}} \end{aligned} \tag{3.20}$$

D'une manière générale, les pires cas vont évoluer en $C_n^{c(R)}$ c'est à dire en $n^{c(R)}$. Imaginons maintenant que l'on utilise dans la fabriquation d'une densité les deux relations R_{s1} et R_{s3}. Dans les pires cas, l'intensité conditionnelle de Papangelou est surtout tributaire des termes correspondant aux relations de cardinalité les plus hautes. Même en rendant les densités intégrables (en limitant le nombre de points par exemple) il va être difficile de règler l'importance accordée à une relation de faible cardinalité par rapport à une relation de grande cardinalité.

Pour ces raisons, nous allons proposer un modèle de densité générique qui permette de travailler avec des relations de cardinalité différentes. Pour cela, il est nécessaire d'introduire les fonctions suivantes :

Définition *Soit une relation $R \in \mathcal{R}$. Une* **fonction de qualité** *est une application qui pour une configuration $\mathbf{x} \in N_U^f$ associe à toute clique r de la relation $R(\mathbf{x})$ un réel compris entre* 0 *et* 1*:*

$$\sigma_{(R,\mathbf{x})} : \begin{array}{ccc} R(\mathbf{x}) & \rightarrow & [0,1] \\ r & \rightarrow & \sigma_{(R,\mathbf{x})}(r) \end{array}$$

Ces applications vont nous servir pour introduire des grandeurs qualitatives sur les relations pour améliorer le comportement dynamique de la chaîne de Markov construite, comme cela a été présenté dans la partie 3.3.2 . Nous définissons maintenant trois fonctions :

Définition *On appelle* **intensité d'une relation** *l'application de $\mathcal{R}_c \times N_U^f$ dans $\mathbb{R}$ définie comme suit :*

$$\boxed{I(R,\mathbf{x}) = \frac{c(R)p(R,\mathbf{x})}{n_V(R,\mathbf{x})} = \frac{c(R)\,card(R(\mathbf{x}))}{card(V(R,\mathbf{x}))}} \tag{3.21}$$

Si $n_V(R,\mathbf{x})$ vaut 0, alors $p(R,\mathbf{x})$ vaut également 0. On pose donc dans ce cas : $I(R,\mathbf{x}) = 0$.

De manière intuitive, l'intensité représente le nombre moyen de cliques auxquelles appartient un point, sachant que ce point appartient à au moins une clique.

Définition *On appelle* **diffusion d'une relation** *l'application de $\mathcal{R} \times N_U^F$ dans $[0,1]$, définie de la manière suivante :*

$$\boxed{d(R,\mathbf{x}) = \frac{n_V(R,\mathbf{x})}{n(\mathbf{x})}} \tag{3.22}$$

Si $n(\mathbf{x})$ vaut 0, on pose $d(R,\mathbf{x}) = 0$.

La densité représente le taux de points impliqués dans au moins un élément de la relation.

Définition *On appelle* **qualité** *d'une relation toute application qui s'écrit :*

$$\boxed{\begin{aligned} \mathcal{Q}(R,\mathbf{x}) &= \frac{1}{p(R,\mathbf{x})} \sum_{r \in R(\mathbf{x})} \sigma_{(R,\mathbf{x})}(r) \\ &= \frac{1}{card(R(\mathbf{x}))} \sum_{r \in R(\mathbf{x})} \sigma_{(R,\mathbf{x})}(r) \end{aligned}} \tag{3.23}$$

où les $\sigma_{(R,\mathbf{x})}(.)$ sont des fonctions de qualité.

La qualité d'une relation pour une configuration représente donc la qualité moyenne des cliques de cette configuration.

3.3.4 Le type de modèle a priori adopté.

Maintenant que nous avons présenté le cadre formel et les notations utilisées, on va s'intéresser à la définition de densités de probabilité mettant en jeu des relations. On considére une seule relation $R \in \mathcal{R}_c$. On prend alors pour densité:

$$\boxed{h(\mathbf{x}) = \beta^{n(\mathbf{x})} f_I(R,\mathbf{x}) f_d(R,\mathbf{x}) f_{\mathcal{Q}}(R,\mathbf{x})} \tag{3.24}$$

Où:

$$f_I(R,\mathbf{x}) = \left(\frac{I(R,x) + \epsilon}{\alpha_R + \epsilon}\right)^{\gamma_{I,R}(\alpha_R - I(R,\mathbf{x}))} \tag{3.25a}$$

$$= \exp\left(-\gamma_{I,R}(I(R,\mathbf{x}) - \alpha_R) \log\left(\frac{I(R,\mathbf{X}) + \epsilon}{\alpha_R + \epsilon}\right)\right) \tag{3.25b}$$

Cette fonction appelle plusieurs remarques :

1. $\gamma_{I,R} \geq 0$.

2. $\alpha_R \geq 0$ est le paramètre important : nous allons voir pourquoi ci -dessous.
3. le paramètre $\epsilon > 0$ sert à autoriser les cas où α vaut 0.
4. la fonction:
$$I \rightarrow (I - \alpha) \log \left(\frac{I + \epsilon}{\alpha + \epsilon} \right) \tag{3.26}$$
est toujours positive, sauf en $I = \alpha$ où elle vaut 0. De plus, elle est convexe. Le paramètre α sert donc à placer le minimum de cette fonction. Dans le cadre de la densité ci-dessus, on place donc un maximum. Cela revient à construire un mode de la densité.
Le fait que l'exposant de la densité (3.25b) soit toujours négatif est très utile. Cela revient à toujours rendre répulsif les cas qui ne nous plaisent pas. Par conséquent, l'intégrabilité de la densité est toujours assurée puisque l'on vérifie aisément la première condition de Ruelle (2.32).
5. Le paramètre $\gamma_{R,I}$ permet de jouer sur le compromis exigence/précision: plus γ est élevé, plus on est exigent sur le mode. Plus γ est faible, et moins on pénalise les configurations $\mathbf{x}$ dont l'intensité est éloignée de α.
6. Il faut encore que cette densité satisfasse la deuxième condition de Ruelle 2.33 pour être simulable facilement par un algorithme de Metropolis Hastings Green : pour cela, comme I n'est pas une grandeur bornée, on tronque la fonction I et on la remplace par:
$$\tilde{I} = \begin{cases} I_{\min} & \text{si} \quad I \leq I_{\min} \\ I & \text{si} \quad I \in [I_{\min}, I_{\max}] \\ I_{\max} & \text{si} \quad I \geq I_{\max} \end{cases}$$
Une autre manière d'assurer la deuxième condition de Ruelle aurait été de vérifier que lorsque l'on ajoute un point u à une configuration $\mathbf{x}$ l'intensité ne peut pas "trop" varier. Pour cela, il est plus simple de borner I et d'en déduire que les variations de I données par $|I(R,\mathbf{x} \cup u) - I(R,\mathbf{x})|$ sont donc bornées.

On construit ensuite le même type de fonctions à partir de la diffusion. La différence majeure tient au fait que cette grandeur est bornée et que par conséquent la deuxième condition de Ruelle est automatiquement assurée.

$$f_d(R,\mathbf{x}) = \left(\frac{d(R,\mathbf{x}) + \epsilon}{\delta_R + \epsilon} \right)^{\gamma_{d,R}(\delta_R - d(R,\mathbf{x}))} \tag{3.27}$$

Enfin, il reste à construire la fonction $f_{\mathcal{Q}}$. Ici, le contexte est un peu différent puisque de toute façon l'objectif est d'avoir une qualité proche de 1. Sinon, cela revient à changer les définitions des fonctions de qualité $\sigma_{R,\mathbf{x}}$ sur les cliques :

$$f_{\mathcal{Q}}(R,\mathbf{x}) = \exp\left(-\gamma_{\mathcal{Q},R}(1 - \mathcal{Q}(R,\mathbf{x}))\right) \tag{3.28}$$

Ce qui, à la constante de normalisation près, revient à poser :

$$f_{\mathcal{Q}}(R,\mathbf{x}) = \exp\left(\gamma_{\mathcal{Q},R}\mathcal{Q}(R,\mathbf{x})\right) \tag{3.29}$$

Pour généraliser cela aux cas où il y a plus d'une relation, il suffit de considérer :

$$h_P(\mathbf{x}) = \beta^{n(\mathbf{x})} \prod_{R \in \mathcal{R}} f_I(R,\mathbf{x}) f_d(R,x) f_{\mathcal{Q}}(R,\mathbf{x}) \tag{3.30}$$

Faisons quelques remarques:
1. En mettant l'un des γ à 0, on supprime le terme correspondant.
2. La densité est intégrable et vérifie la deuxième condition de Ruelle, par conséquent on peut la simuler facilement par l'algorithme présenté dans le deuxième chapitre.
3. Il y a deux sortes de paramètres : α et δ d'une part et les γ d'autre part. Cela fait beaucoup de paramètres à règler. Cependant il faut remarquer que les α et δ correspondent à des grandeurs physiques. On peut donc les fixer très facilement, soit parce que l'on a un objectif précis, soit par une évaluation statistique sur des exemples.

4. Les γ peuvent être redécomposés sous une autre forme: pour une relation $R \in \mathcal{R}$:

$$\begin{pmatrix} \gamma_{I,R} \\ \gamma_{d,R} \\ \gamma_{Q,R} \end{pmatrix} = \rho_R \begin{pmatrix} \tau_{I,R} \\ \tau_{d,R} \\ \tau_{Q,R} \end{pmatrix} \quad \text{avec} \quad \tau_{I,R} + \tau_{d,R} + \tau_{Q,R} = 1 \tag{3.31}$$

   Cela permet de régler l'influence d'une relation au travers de ρ_R et au sein de cette relation, de règler séparément l'influence des différentes grandeurs.
5. Toutefois, une des remarques majeures à faire est que le réglage de ces paramètres γ n'est pas forcément nécessaire. Il ne l'est que si les relations considérées empêchent de satisfaire à toutes les contraintes pour des raisons géométriques. En effet les paramètres γ ne changent pas la position des modes.
6. Le modèle présenté ici n'est plus du tout markovien, ce qui ne pose pas de problème particulier.
7. Pour pouvoir simuler ce type de modèle il faut avoir accès aux grandeurs: $n(\mathbf{x})$, $p(R,\mathbf{x})$, $n_V(R,\mathbf{x})$.

3.3.5 Les résultats obtenus.

Exemple du processus de Strauss.

On prend maintenant $X_{\max} = 10$ et $Y_{\max=10}$.

On va simuler l'équivalent d'un processus de Strauss: on considère donc la relation R_{s1} uniquement. On ne considère pas de terme de qualité. Le paramètre de la relation est mis à $d_{\max} = 0.6$.

On fixe β de manière à ce que l'espérance du nombre de point du processus sans les interactions vale 70. Pour les expériences 3 et 4, on a choisit pour valeur de β celle qui correspond à une espérance de nombre de point valant 300 .

On fixe les paramètres α et δ et on effectue un recuit simulé sur la densité proposée:

$$\beta^{n(\mathbf{x})} \left(f_I(R_{s1},\mathbf{x}) f_d(R_{s1},\mathbf{x}) \right)^{\frac{1}{T}} \tag{3.32}$$

β, rappelons le, est un paramètre d'échelle: il sert à régler la mesure de référence contre laquelle on effectue le recuit simulé.

Expérience 1 : On fixe $\alpha = 1$ et $\delta = 1$ avec les γ correspondants valant tous les deux 1. Cela veut dire que l'on veut que tous les points appartiennent à au moins une clique et que pour les points appartenant à au moins une clique le nombre moyen de cliques auxquelles ils appartiennent vale 1. La figure 3.1 montre le résultat du recuit simulé effectué avec une constante de décroissance géométrique valant 0.99999 :
- il y a 69 points dans la configuration présentée.
- cette configuration contient 37 cliques.
- tous les points sauf un appartiennent à au moins une clique.

On voit que les points sont associés par paires, ce qui correspond à l'exigence d'une intensité valant 1. La figure 3.2 montre les résultats en terme d'évaluation de la convergence : les grandeurs d'intérêt sont : le nombre de points des configurations $n(\mathbf{x}_t)$ et le nombre de cliques $p(R_{s1},\mathbf{x}_t)$. Pour effectuer les 25 millions d'itérations, il a fallu 25 minutes sur une `Sun Ultra 10 - 64 M Ram - 250 Mhz`.

Expérience 2 : On fixe $\alpha = 3$ et $d = 0.5$. On veut donc trois cliques par points en moyenne et que la moitié des points seulement appartiennent à une clique. On obtient la configuration de la figure 3.3, tandis que la figure 3.4 montre l'evolution du nombre de points. La figure 3.3 montre

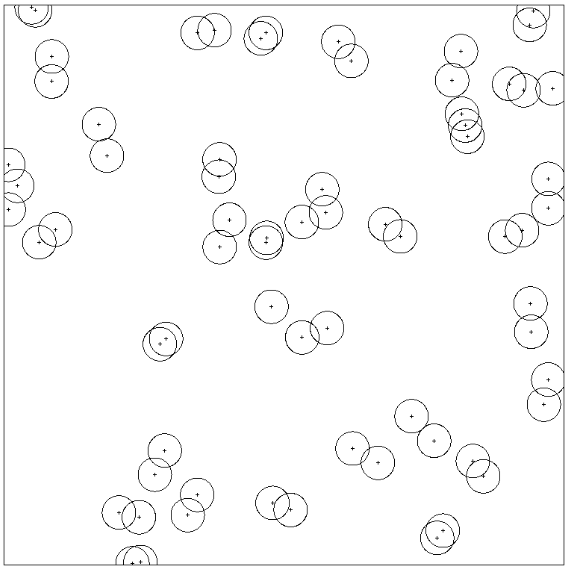

FIG. 3.1 – *Simulation 1 : on considère R_{s1} et on a fixé $\alpha = 1$ et $\delta = 1$.*

un résultat intéressant : d'une part l'exigence de diffusion est respectée. D'autre part la moyenne imposée sur I est respectée mais avec une grande variance. Pour limiter ce genre de phénomènes, il suffirait d'introduire les moments d'ordres supérieurs.

- il y a 83 points dans la configuration présentée.
- cette configuration contient 62 cliques.
- sur les 83 points, 42 appartiennent à au moins une clique.

Processus de strauss et alignements.

Expérience 3 : On utilise à la fois R_{s1} et R_3 : on joue à la fois sur la relation de proximité et la relation d'alignement.

On fixe $\alpha_{R_{s1}} = 1$ et $\alpha_{R_3} = 1$. on fixe également $\delta_{R_{s1}} = 1$ et $\delta_{R_3} = 1$. Pour R_3 on fixe le paramètre $\epsilon_{\max}$ à 0.6.

Une subtilité, toutefois : dans notre définition de R_3 nous avons utilisé R_{s1} comme relation mère de R_3. Nous entendons par là que l'on avait imposé :

$$\begin{cases} \{u_i,u_j\} \in R_{s1}(\mathbf{x}) \\ \{u_i,u_k\} \in R_{s1}(\mathbf{x}) \end{cases}$$

où R_{s1} prenait comme paramètre $d_{\max} = 0.6$. Ici nous avons utilisé une autre relation mère pour R_3 , moins contraignante, de distance seuil : $d'_{\max} = 1$.

Les figures 3.5 et 3.6 montrent les résultats obtenus.

Expérience 4 : On utilise les mêmes paramètres que pour l'expérience 3, mais on change $\epsilon_{\max} = 0.3$ et on ajoute un terme de qualité : la qualité d'une clique $(\{u_i\},\{u_j,u_k\})$ est donnée par :

$$\sigma_{(R_3,\mathbf{x})}\left((\{u_i\},\{u_j,u_k\})\right) = \frac{1}{\epsilon_{\max}^2}\left(\frac{1+\epsilon_{\max}^2}{1+\|\overrightarrow{u_iu_j}+\overrightarrow{u_iu_k}\|^2}-1\right) \tag{3.33}$$

Cette fonction vaut 1 lorsque l'alignement est parfait et 0 lorsqu'on est sur le bord du support imposé par la relation.

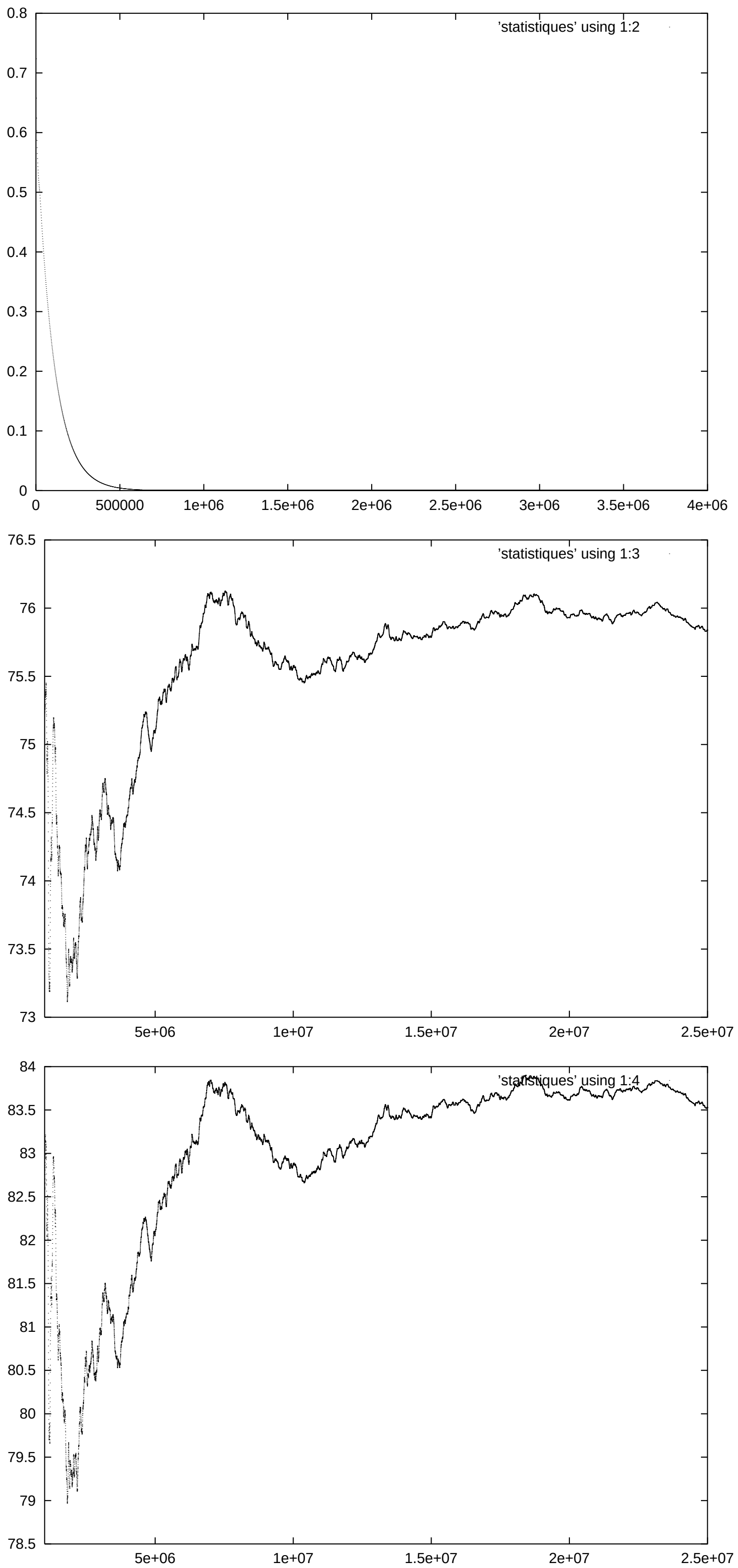

FIG. 3.2 – *Simulation 1 :* R_{s1} *avec* $\alpha = 1$ *et* $\delta = 1$. *Résultats de convergence : de gauche à droite et de haut en bas : évolution de la température du recuit simulé, espérance du nombre de points dans les configurations et espérance du nombre de cliques.*

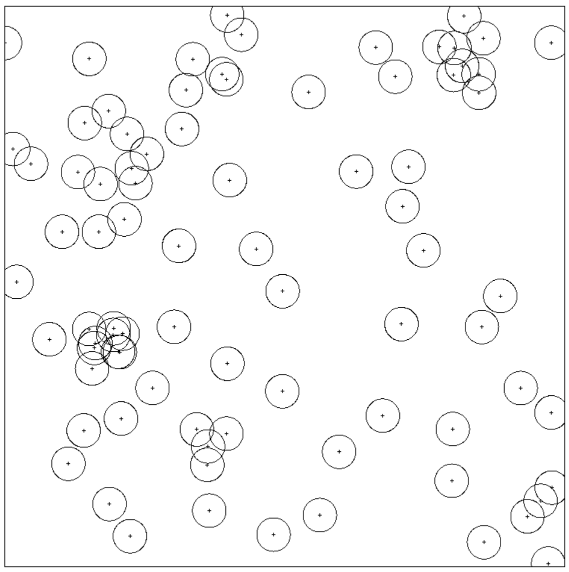

FIG. 3.3 – *Simulation 2 : on considère* R_{s1} *et on a fixé* $\alpha = 3$ *et* $\delta = 0.5$.

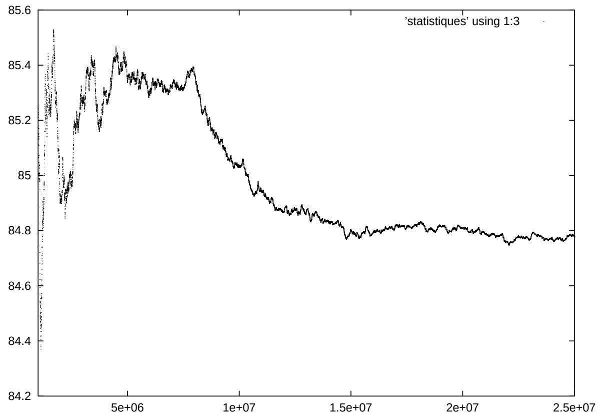

FIG. 3.4 – *Simulation 2 : évolution de la moyenne empirique de* $n(\mathbf{x}_t)$.

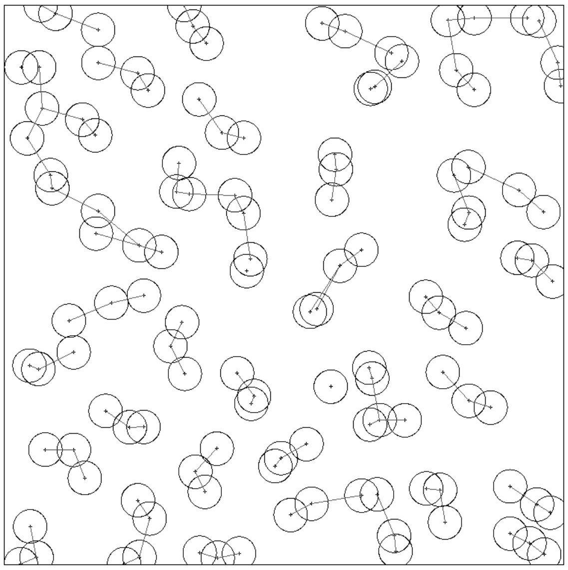

FIG. 3.5 – *Simulation 3 : Recuit simulé. On considère R_{s1} et R_3. On a fixé :* $\alpha_{R_{s1}} = 1$,$\alpha_{R_3} = 1$, $\delta_{R_{s1}} = 1$ *et* $\delta_{R_3} = 1$.

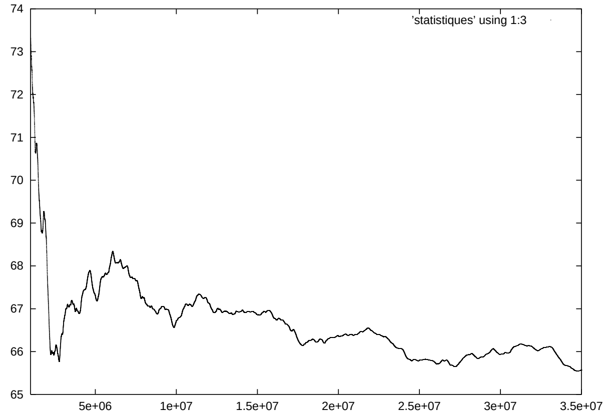

FIG. 3.6 – *Simulation 3 : évolution de la moyenne empirique de* $n(\mathbf{x}_t)$.

Les figures 3.7 et 3.8 montrent les résultats obtenus. On notera que le rôle du paramètre d'intensité est bien respecté.

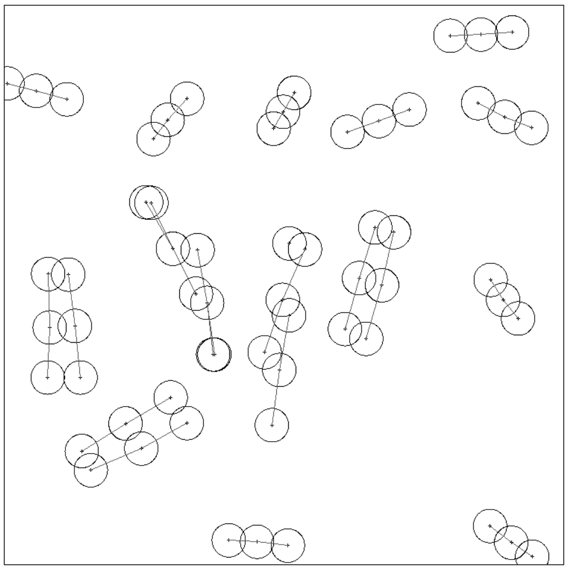

FIG. 3.7 – *Simulation 4 : Recuit simulé. On considère R_{s1} et R_3. On a fixé : $\alpha_{R_{s1}} = 1$,$\alpha_{R_3} = 1$, $\delta_{R_{s1}} = 1$ et $\delta_{R_3} = 1$. On a également ajouté un terme de qualité pour favoriser les bons alignements.*

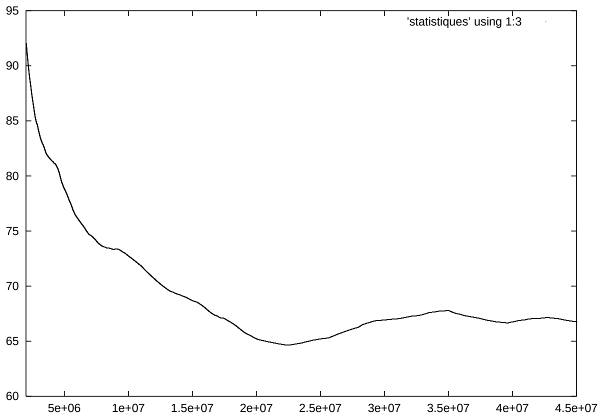

FIG. 3.8 – *Simulation 4 : évolution de la moyenne empirique de $n(\mathbf{x}_t)$.*

3.3.6 Modèle a priori sur les maisons.

Les types de relation considérés.

On considère quatre types de relation :

1. **La relation R_{s0} d'intersection entre les rectangles** : $c(R_s0) = 2$.

   On fixe $\alpha_{R_s0} = 1$ et $\delta_{R_s0} = 0$. On veut éviter les intersections de maisons. On ne met pas de terme de qualité sur cette relation.

2. **La relation R_{s1} de proximité sur les points.**

   On met comme paramètre physique : $d_{\max} = 1.5$. On suppose que la zone urbaine est dense, donc on a décidé de mettre $\alpha_{R_s1} = 3$, tandis que l'on autorise $\delta_{R_{s1}} = 0.75$.
   Cette relation contient un terme de qualité de la même forme que (3.33) : on remplace la norme au carré par $(l(u_i) - l(u_j))^2 + (L(u_i) - L(u_j))^2$ et $\epsilon^2_{\max}$ par $(l_{\max} - l_{\min})^2 + (L_{\max} - L_{\min})^2$. On utilise ensuite la même fonction :

   $$d \to \frac{1}{d^2_{\max}\left(\frac{1+d^2_{\max}}{1+d^2} - 1\right)}. \tag{3.34}$$

   Ce terme de qualité correspond à une mesure d'homogénéïté entre les deux maisons. On cherche à rendre les maisons voisines identiques.

3. **La relation R_{s4} d'alignement dans la longeur.**

   C'est une relation particulière : elle regarde si deux maisons sont alignées dans leur lon-

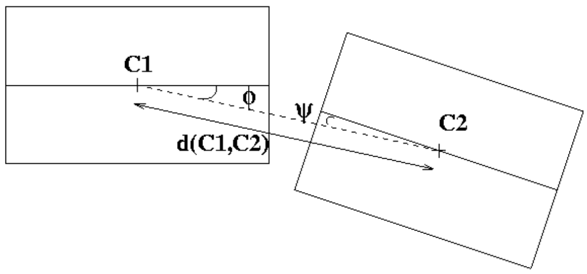

FIG. 3.9 – *Modèle a priori : relation R_{s4} : alignement dans la longueur.*

   geur. La figure 3.9 montre les grandeurs considérées. On impose trois conditions :

   $$\{u_i,u_j\} \in R_{s4}(\mathbf{x}) \quad \text{ssi} \quad \begin{cases} \{u_i,u_j\} \in R_{s1}(\mathbf{x}) \\ |\Delta\varphi(u_i)| \leq \Delta\phi_{\max} \\ |\Delta\varphi(u_j)| \leq \Delta\phi_{\max} \end{cases} \tag{3.35}$$

   Ici, on a pris comme valeur de $\Delta\phi_{\max}$ l'angle $\frac{\pi}{6}$.

   On a fixé $\alpha_{R_{s4}} = 2$ et $\delta_{R_{s4}} = 1$. Cela signifie que l'on veut que tous les points appartiennent à un alignement de deux maisons. On remarquera que ceci est contradictoire avec l'exigence de diffusion de R_{s1} à 0.75 . On a également mis un terme de qualité utilisant la fonction 3.34 et comme mesure d'homogénéïté : $d^2 = \Delta\varphi(u_i)^2 + \Delta\varphi(u_j)^2$. La valeur de $d^2_{\max}$ est alors donnée par : $2\Delta\phi^2_{\max}$. Ceci sert à améliorer les alignements dans la longueur.

4. **La relation R_3 d'alignement de trois points.**

   On a pris pour paramètre physique $\epsilon_{\max} = 0.3$. On a fixé $\alpha_{R_3} = 3$ et $\delta_{R_3} = 1$. Le terme de qualité est le même que celui donné dans l'expérience 4.

Voila qui définit la physique des relations. Il reste à fixer les paramètres γ.

Les paramètres du modèle.

On travaille sur ces paramètre sous leur forme ρ et τ. Les paramètres ont été reglés par un système d'essai-erreur. Toutefois, cela n'est pas trop difficile, dans le sens où les relations ne sont pas trop concurrentes.

Un seul paramètre est délicat à règler : $\rho_{R_{s0}}$. Ce paramètre correspond à l'importance accordée aux intersections de maisons. Si on lui donne trop d'importance, cela limite la dynamique de la chaîne de Markov produite. L'idéal serait de pouvoir faire intervenir un terme de qualité sur l'aire de la surface commune entre deux maisons. Malheureusement cela prendrait beaucoup de temps en terme de calcul informatique.

Les résultats obtenus

Nous obtenons des résultats intéressants pour le modèle a priori. Cela donne à penser que le type de modèle proposé se comporte très bien dans le cadre de la mise en place de plusieurs interactions complexes entres les points. La figure 3.10 montre un exemple de résultat obtenu par recuit simulé sur ce modèle.

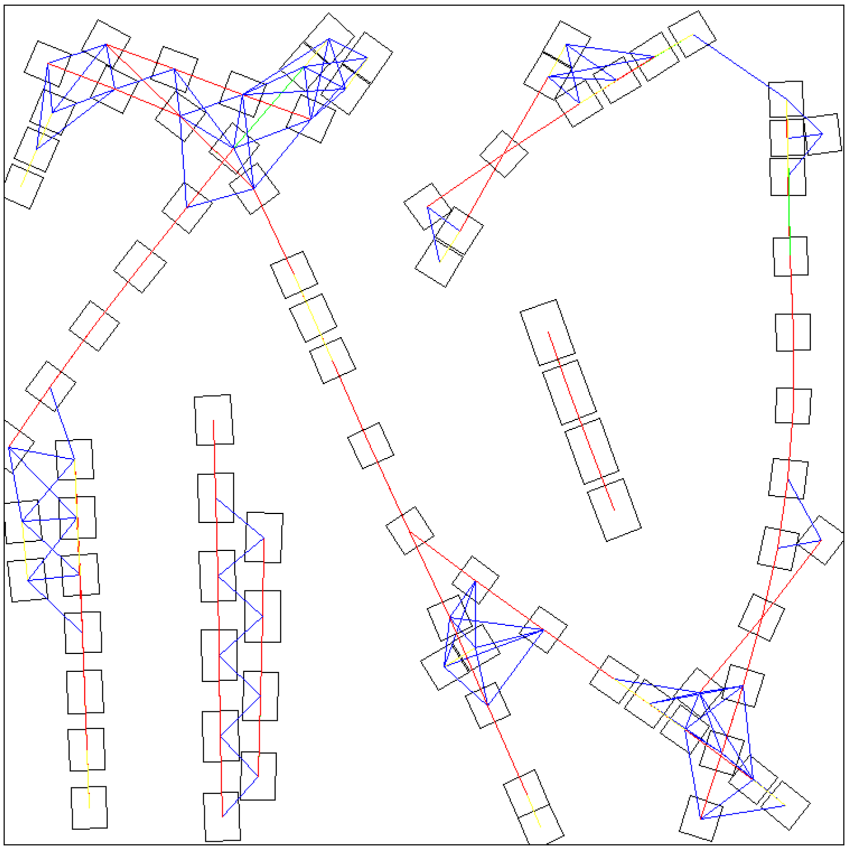

FIG. 3.10 – *Modèle a priori : résultat d'un recuit simulé sur le modèle a priori.*

3.4 Terme d'attache aux données.

C'est la partie la plus sensible du travail présenté dans ce rapport. Il y a sans doute bien des choses à améliorer dans ce contexte. Nous avons travaillé dans un cadre bien plus simple que celui proposé par le sujet. Les images de référence que nous avons prises sont des images binaires.

3.4.1 Restriction à des données binaires.

Cela peut paraitre très restrictif, d'autant plus que les images en question sont très propres. Le but ici est de tester la faisabilité de l'approche.

Génération des images de synthèse.

Pour générer les images de synthèse, nous avons utilisé le modèle a priori en recuit simulé avec un paramètre de température minoré par une grandeur assez élevée. Cela permet d'obtenir des configurations qui vont dans le sens du modèle mais avec un "bruit" assez important. Une remarque toutefois : on a augmenté le paramètre $\rho_{R_{s0}}$ règlant l'importance des intersections, pour que même à faible température on puisse avoir une réalisation qui ait un sens physique.

L'image de référence que nous avons générée est présentée par la figure 3.11.

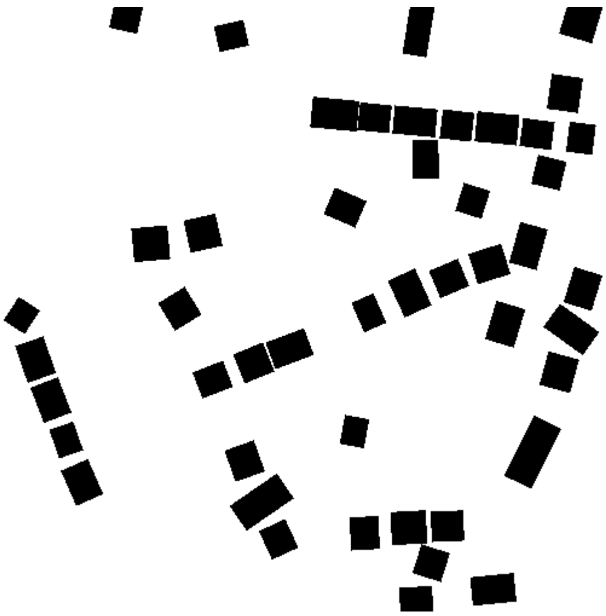

FIG. 3.11 – *Image de donnée de synthèse. Elle est obtenue par un recuit simulé sur le modèle a priori en minorant la température par une valeur élevée.*

3.4.2 Relations d'attache aux données.

Puisque l'outil informatique dont nous disposons (cf Annexes) permet de travailler facilement avec des relations, il nous a paru plus facile et plus propre de faire intervenir ces notions dans la construction de la vraisemblance.

Nous avons donc introduit une relation R_{sD} de cardinalité constante et valant $c(R_{sD}) = 1$. Une relation est avant tout défini par une ou plusieurs conditions, comme nous avons pu le voir dans les exemples de relations précédents. On peut ensuite ajouter un terme de qualité.

Nous avons donc décidé d'introduire la relation R_{sD} sous la forme :

$$\{u_i\} \in R_{sD}(\mathbf{x}) \qquad \text{ssi} \quad \text{test}(u_i) = \text{Vrai} \tag{3.36}$$

Cette fonction test(.) peut être un test sous un modèle statistique. Nous l'avons définie en pratique de manière très simpliste : le test est Vrai si la silhouette de la maison projetée sur l'image de données contient au moins une certaine proportion de pixel noirs. Le problème est que cette étape est l'une des plus longues du point de vue informatique.

Le premier paramètre physique de cette relation d'attache aux données est donc le taux en question. On considère donc une image binaire de données: $\{y_t, t \in T\}$ où T est la grille support des y_t qui appartiennent à $\{0,1\}$. On appelle Silhouette d'un objet u sur cette grille, l'ensemble des t tels que t soit dans l'objet u. On la note $\Lambda(u_i) \subseteq T$ Alors la fonction test que nous avons pris est :

$$\text{test}(u_i) = \text{Vrai} \quad \text{ssi} \qquad \frac{\sum_{t\in\Lambda(u_i)} y_t}{\text{card}\,(\Lambda(u_i))} \geq \eta \tag{3.37}$$

Dans notre cas, nous avons pris $\eta = 0.6$.
Nous avons ajouté un terme de qualité qui tient compte à la fois du taux de pixel noir à l'intérieur de la silhouette, et du taux de pixel blanc sur le pourtour de la silhouette.

3.4.3 Résultats et discussion.

La figure 3.12 montre un exemple de resultat obtenu par recuit simulé sur la vraisemblance ainsi construite. Les résultats sont interessants mais soulèvent quelques questions que nous n'avons pas eu le temps d'élucider :

- Il faut revoir le terme d'attache aux données pour obtenir des résultats plus précis.
- La dynamique de l'algorithme utilisant comme estimateur le Maximum a Posteriori est considérablement handicapée par le fait que pour contrebalancer l'influence de l'intensité il faut donner à l'attache aux données une forte influence.
- A cela s'ajoute le fait que si l'on augmente l'influence de l'attache aux données, il faut également donner de l'influence aux intersections.

En conclusion, il semble que le Maximum a Posteriori ne soit pas le meilleur estimateur pour notre application, tant sur le plan pratique que sur le plan théorique.

3.5 Choix de l'estimateur.

3.5.1 Le MAP.

Le Maximum a Posteriori et ses inconvénients ont été présenté dans les paragraphe précédent. On retiendra qu'il est inadapté à notre problème pour les raisons suivantes :

1. le cadre n'est pas bayesien. Par conséquent la pondération entre l'attache aux données et le modèle a priori joue un rôle majeur sur l'estimateur obtenu : il est alors délicat d'utiliser la partie a priori de la densité pour aider la dynamique de l'algorithme et de vouloir minimiser le nombre de maisons non attachées aux données.
2. la mesure contre laquelle on effectue le recuit simulé est la loi d'un processus de Poisson. Comme, en pratique, le recuit simulé est arrêté pour une certaine température, le paramètre d'intensité de cette mesure joue un rôle important sur l'estimée obtenu.
3. en utilisant un recuit-simulé, on choisit un mode de la densité. Il n'est pas pertinent dans notre cas de choisir entre deux modes. Il est plus intéressant de tenir compte de l'information contenue dans les deux modes.

C'est pourquoi nous nous intéressons à un autre estimateur, proposé par Rue dans [32] suite aux travaux de Baddeley (cf. [1]).

3.5.2 L'estimateur de Rue.

Métrique de Baddeley.

L'ambition de Baddeley dans [1] est de fournir une métrique adaptée aux problèmes inverses en traitement de l'image. En effet, l'estimé donné par le MAP est l'estimé optimale sous le coût suivant :

$$L_{MAP}(\mathbf{x},\hat{\mathbf{x}}) = \begin{cases} 0 & \text{si } \hat{\mathbf{x}} = \mathbf{x} \\ 1 & \text{sinon.} \end{cases} \tag{3.38}$$

Dans les cas ou $\mathbf{x}$ représente une image, Baddeley a introduit une distance entre deux images, pour permettre de fabriquer des coûts mieux adaptés aux problèmes rencontrés en traitement d'image.

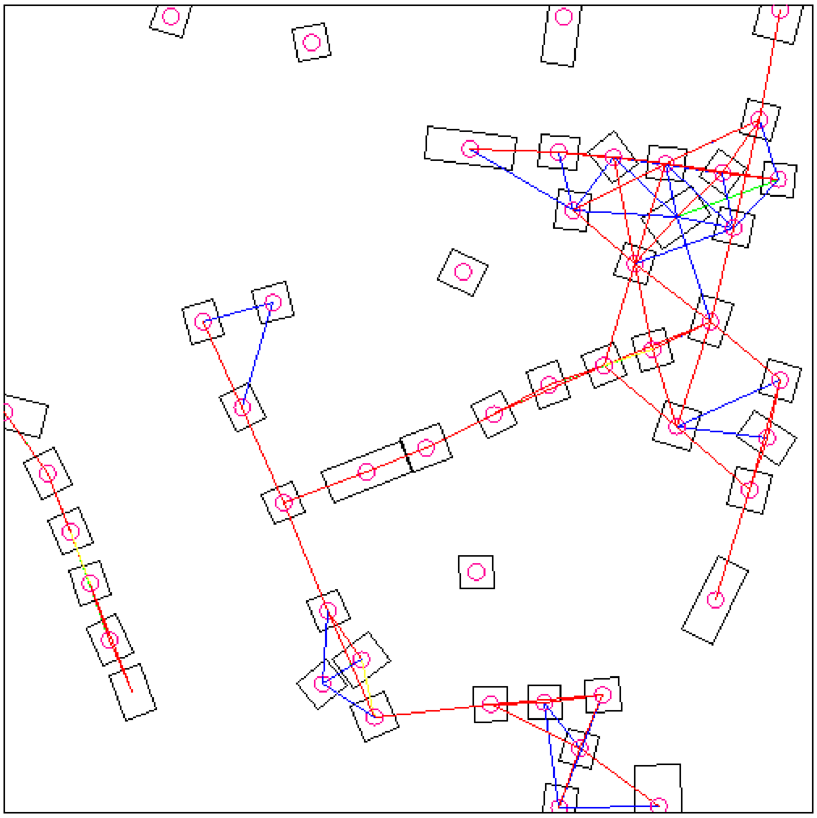

FIG. 3.12 – *Attache aux données : une configuration maximisant la vraisemblance du modèle, obtenue par recuit simulé. Les cercles indiquent les maisons appartenant à une clique d'attache aux données.*

Sa démarche a été de fabriquer une métrique qui vérifie certaines propriétés topologiques.

On considère deux images binaires x et x' : ce sont des applications de S dans $\{0,1\}$. On se donne une grille T de points $t \in S$.

On suppose que S est muni d'une métrique ρ. Le cas usuel est donné par S borné de $\mathbb{R}^2$ muni de la distance euclidienne.

Pour un point s de S, on définit la distance $d(s,x)$ comme étant la ρ-distance minimale entre ce point et le contour de x. Cela veut dire que, pour $s \in S$, on pose :

$$d(s,x) = \min\{\rho(s,s')\ s' \in S \text{ tel que } x(s') = 1\} \tag{3.39}$$

On considère ensuite une fonction concave w de $\mathbb{R}$ dans $\mathbb{R}$ telle que $w(0) = 0$.
La Δ-distance de Baddeley est alors donnée par :

$$\Delta_w^p(x,x') = \left(\frac{1}{\mathrm{card}(T)} \sum_{t\in T} |w(d(t,x)) - w(d(t,x'))|^p\right)^{\frac{1}{p}} \quad , \quad 1 \leq p < \infty \tag{3.40}$$

Il faut noter, que dans notre cas, on passe très facilement d'une réalisation du processus ponctuel aux images binaires. Il suffit de prendre comme image binaire définie par une configuration $\mathbf{x}$ l'union des rectangles.

Estimateur de Rue et MAP.

Pour différentes raisons, Rue pose dans [32] $p = 2$ et $w(d) = \min\{c,d\}$ où c est un paramètre de troncature qui limite l'influence des objets. Il obtient alors une version de la Δ-distance de Baddeley qu'il note : $\Delta_c^2(\mathbf{x},\mathbf{x}')$. Il prend alors comme fonction de coût :

$$L_c(\mathbf{x},\hat{\mathbf{x}}) = \Delta_c^2(\mathbf{x},\hat{\mathbf{x}})^2 \tag{3.41}$$

Ce coût est bien plus intéressant que le coût associé au MAP. Il introduit des différences quantitatives entre des configurations "fausses". On prend comme estimateur :

$$\mathbf{x}_\Delta = \arg\min_{\hat{\mathbf{x}}} \mathbb{E}_{\mathbf{x}|\mathbf{y}}[L_c(\mathbf{x},\hat{\mathbf{x}})]. \tag{3.42}$$

Dans notre cas on remplace $\mathbf{x}|\mathbf{y}$, loi a posteriori du cadre bayésien par la densite proposée dans ce chapître.

Algorithme de Rue.

Il faut calculer ce Δ-estimateur. Pour cela, Rue propose d'effectuer un recuit simulé sur la densité suivante :

$$h(\hat{\mathbf{x}}) = \alpha \exp(-\mathbb{E}_{\mathbf{x}|\mathbf{y}}[L_c(\hat{\mathbf{x}},\mathbf{y})]) \tag{3.43}$$

En fait dans notre cas. cette densité peut s'écrire :

$$h(\hat{\mathbf{x}}) = \alpha' \exp\left(-\frac{1}{|T|}\sum_{t\in T} d_c(t,\hat{\mathbf{x}})(d_c(t,\hat{(x)}) - 2\zeta_t)\right) \tag{3.44}$$

où

$$\zeta_t = \mathbb{E}_{\mathbf{x}|\mathbf{y}}[d_c(t,\mathbf{x})] \tag{3.45}$$

et

$$d_c(t,\mathbf{x}) = \min\{c,d(t,\mathbf{x})\} \tag{3.46}$$

Cette expression donne l'algorithme de Rue qui se décompose en deux étapes :

1. échantilloner $\mathbf{x}|\mathbf{y}$, pour obtenir des estimées des ζ_t. Cela se fait en utilisant l'algorithme de MHG présenté dans le deuxième chapitre.
2. éffectuer un recuit simulé en utilisant ces estimées pour obtenir l'estimée $\mathbf{x}_\Delta$. Pour cela, on utilise le noyau de proposition du premier échantilloneur.

Résultats et commentaires.

La deuxième partie de l'algorithme, qui consiste à minimiser la fonction de coût qui est une distance par rapport à l'image d'espérance donnée par les ζ_t pose de gros problèmes tant pour les temps de calculs que pour la qualité des résultats.

Ces problèmes sont d'ordre dynamique. De façon générale, il apparaît que l'algorithme de recuit simulé est mal adapté à l'optimisation du coût. En effet, ce problème est plutôt convexe. On doit pouvoir trouver un moyen de proposer des mouvements en utilisant cette convexité plutôt que de proposer des mouvements qui vont être rejetés systématiquement.

Ensuite, rendu à un certain stade de l'optimisation, les transformations semblent être très mal adaptées à la Δ-distance. En particulier les points de la grille situés loin des objets semblent avoir trop d'importance parce qu'étant plus nombreux que les points proches. Cet effet doit pouvoir être minimisé en utilisant une autre fonction concave que celle proposée par Rue.
La prochaine étape de ce travail est donc la suivante :

- soit proposer un algorithme pour la deuxième partie mieux adapté à notre problème que le recuit simulé.
- soit travailler sur la métrique, et en particulier sur le fait que nos objets sont assez simples et que par conséquent une grille contenant trop de points peut nuire à la dynamique de l'algorithme.

Quoiqu'il en soit, la méthode proposée par Rue nous semble être la mieux adaptée. La figure 3.13 présente, en effet, une image de l'espérance de la Δ-distance, c'est à dire l'estimation des ζ_t sur une image. On remarquera que l'information contenue dans cette image est intéressante. Changer la métrique ou la fonction de coût reviendrait à changer le type d'information que l'on prend en compte.

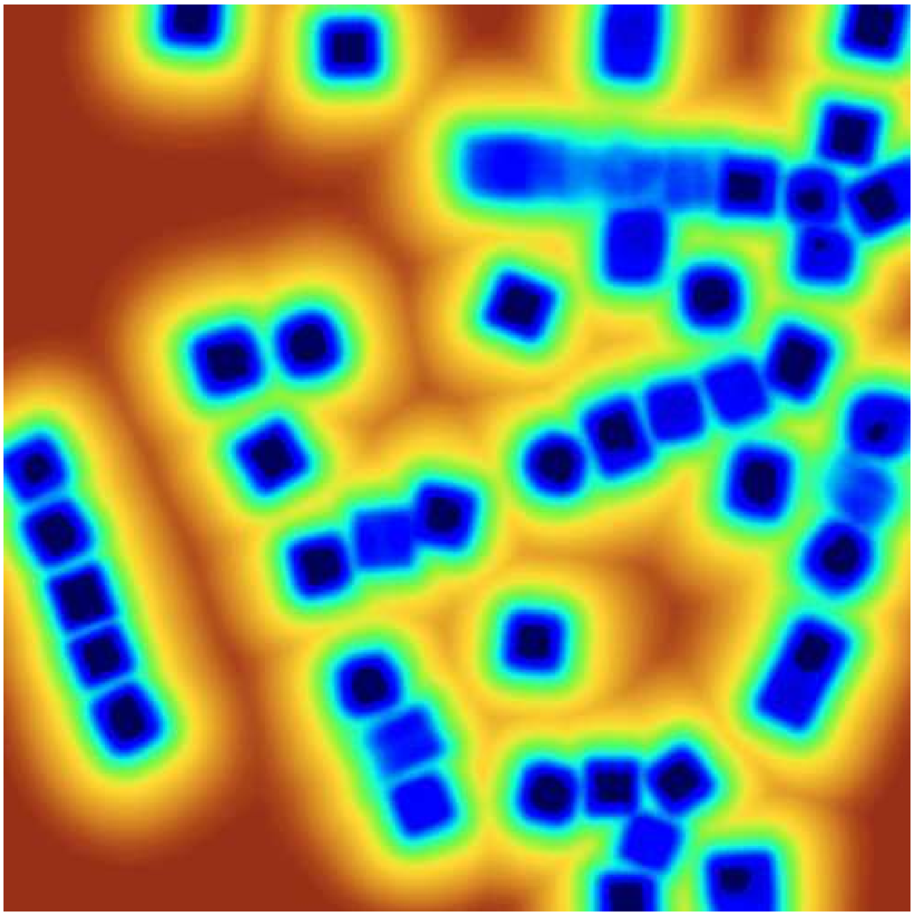

FIG. 3.13 – *Estimateur de Rue : image de l'espérance de la* Δ*-distance. Chaque point de la grille représente donc une estimée d'un* ζ_t.

Chapitre 4

Noyau RJMCMC utilisé.

4.1 Généralités.

4.1.1 Les objectifs.

L'algorithme présenté dans la partie 2.4.1 permet d'explorer l'espace de toutes les configurations possibles et possède les bonnes propriétés mathématiques de convergence. Cependant, comme nous l'avons vu dans la partie 2.3.2, il est important de fabriquer un noyau de proposition qui propose des mouvements correspondant bien à la densité que nous cherchons à simuler et ce pour trois raisons majeures :

- pour accélérer la vitesse de convergence et pour permettre, dans le cas de recuits simulé, d'utiliser des constantes de décroissance géométrique assez faibles.
- pour augmenter la mélangeance de la chaine et parcourir plus de modes de la densité sans s'attarder sur l'un de ces modes.
- pour diminuer l'autocorrélation de la chaine de Markov produite en diminuant le taux de rejet.

On rappelle qu'il manque un cadre formel à toute ces notions intuitives. Cependant, l'expérience parle en faveur de ces heuristiques.

Nous allons proposer ici d'autres mouvements que la simple naissance ou la simple mort. Certains de ces mouvements sont intuitifs : les translations, les dilatations et les rotations. D'autres le sont d'autant moins qu'il est nécessaire d'assurer la condition (2.19) d'équilibre minutieux pour garantir les propriétés de bonne convergence de l'algorithme.

Enfin, tout ce travail est permis dès lors que l'on garde la transformation de naissance ou mort simple puisque certaines propriétés de la chaine de Markov sont garanties par cette transformation.

4.1.2 Formule de Green.

Nous allons présenter ici la formule de Green qu'il démontre dans [15].
Nous n'utilisons pas cette formule dans la suite, puisqu'il est plus simple d'utiliser les mesures directement comme cela a été fait dans les parties 1.3.2 et 2.4.1. Cependant cette formule permet de comprendre les mécanismes mis en jeu par la condition "d'équilibre minutieux" 2.19.

On se place dans le cadre général de l'algorithme de Metropolis Hastings Green, mais dont l'espace d'état est $\bigcup_{d=0}^{N} \mathbb{R}^d$. C'est à dire que l'on veut explorer un état de dimension variable. Cela est utile pour faire de la sélection de modèle statistique. Un exemple courant peut être l'application d'un tel algorithme aux modèles graphiques.
On suppose que l'on veut proposer à partir de $x \in \mathbb{R}^n$ un vecteur $y \in \mathbb{R}^{n+m}$ de dimension supérieure. L'idée est donc de générer un vecteur $u \in \mathbb{R}^m$ suivant une densité $q(.)$ par rapport à la mesure de Lebesgue sur $\mathbb{R}^m$. On applique alors une fonction déterministe ζ de $\mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{n+m}$ pour obtenir $y = \zeta(x,u)$. La condition d'équilibre minutieux impose que cette transformation ζ soit un

difféomorphisme. Green propose en effet de prendre pour mesure d'intérêt la mesure ξ suivante : pour $A \subseteq \mathbb{R}^n$ et $B \subseteq \mathbb{R}^{n+m}$

$$\xi(A \times B) = \xi(B \times A) = \lambda_{n+m}\{(x,u) \quad x \in A \quad \zeta(x,u) \in B\} \tag{4.1}$$

Pour des A et B quelconques, on définit alors:

$$\xi(A \times B) = \xi\{(A \cap \mathbb{R}^n) \times (B \cap \mathbb{R}^{n+m})\} + \xi\{(A \cap \mathbb{R}^{n+m} \times (B \cap \mathbb{R}^n)\} \tag{4.2}$$

La symétrie de la mesure impose la dimension du vecteur u. Si celui-ci ne "complete" pas x, la transformation arrive dans un ensemble (hyperplan ou pire) de mesure nulle.
En écrivant alors la condition d'équilibre, on se rend compte que l'on voudrait appliquer un changement de variable. D'où la nécessité de prendre un difféomorphisme. La formule de Green est alors :

$$R(x,y) = \frac{h(y)}{h(x)} \frac{j_-(y)}{j_+(x)} \frac{1}{q(u)} |J_\zeta(y)| \tag{4.3}$$

Où :
- $j_-(y)$ est la probabilité de choisir le mouvement de retour dans l'état y,
- $j_+(x)$ est la probabilité de choisir le mouvement aller dans l'état x,
- $J_\zeta(y)$ est le jacobien de ζ calculé en $(x,u) = \zeta^{-1}(y)$. On remarquera que sa présence est naturelle puisque l'on mesure un ensemble au travers de la transformation ζ dans l'expression de la mesure (4.1).

Comme les configurations ne sont pas ordonnées, notre cadre d'application est très différent de cet exemple. Cependant nous allons voir que nous retrouvons les notions intuitives qu'elle met en jeu.

Cet exemple est un exemple de "**dimension matching**" : la transformation passe d'un espace à un espace de dimension différente.

Par extension, nous appelons transformations "simples" les transformations que nous avons mises en place qui ne change pas la taille de la configuration et transformations "complexes" les autres.

4.2 Transformations simples.

4.2.1 Cadre général.

On reprend le cadre proposé par Garcin dans [9].

On considére une variable aléatoire Z sur $\Sigma \subseteq \mathbb{R}^k$ où Σ est borné, de loi de probabilité uniforme sur Σ : $d\mathbf{P}_Z(.) = \frac{\lambda(.)}{\lambda(\Sigma)}$
On considère ensuite une fonction ζ:

$$\begin{array}{rccc} \zeta: & U \times \Sigma & \rightarrow & U \\ & (u,z) & \rightarrow & v \end{array} \tag{4.4}$$

Définissons le noyau de manière algorithmique : pour $\mathbf{x} = \{u_1, \ldots, u_{n(\mathbf{x})}\}$:
1. On choisit l'un des objets u_i avec une probabilité uniforme.
2. On génère Z suivant la loi uniforme sur Σ .
3. On remplace le u_i choisi par $\zeta(u_i,Z)$.

Du point de vue mathématique, on écrit un tel noyau sous la forme :

$$Q(\mathbf{x},A) = \sum_{u \in \mathbf{x}} \frac{1}{n(\mathbf{x})} \int_{z \in \Sigma} \mathbf{1}_A(\mathbf{x} \setminus u \cup \zeta(u,z)) \frac{\lambda(dz)}{\lambda(\Sigma)} \tag{4.5}$$

On considère ensuite la mesure ξ définie comme suit :
Pour A et B dans $\mathcal{C} = N_U^f$:

$$\xi(A \times B) = \int_A \sum_{u \in \mathbf{x}} \int_{z \in \Sigma} \mathbf{1}_B(\mathbf{x} \setminus u \cup \zeta(u,z)) \lambda(dz) \mu(d\mathbf{x}) \tag{4.6}$$

avec $\mu(.)$ la mesure du processus ponctuel marqué de référence. On impose ensuite à Σ d'être symétrique, et à ζ de vérifier :

$$v = \zeta(u,z) \Leftrightarrow u = \zeta(v, -z) \tag{4.7}$$

ce qui rend ξ symétrique. On vérifie ensuite que $\pi(d\mathbf{x})Q(\mathbf{x},d\mathbf{y})$ est absolument continue par rapport à ξ et que la dérivée au sens de Radon-Nikodym est donnée par :

$$f(\mathbf{x},\mathbf{y}) = \frac{1}{n(\mathbf{x})}\frac{h(\mathbf{x})}{\lambda(\Sigma)} \tag{4.8}$$

On en déduit le taux de Green :

$$R(\mathbf{x},\mathbf{y}) = \frac{f(\mathbf{y},\mathbf{x})}{f(\mathbf{x},\mathbf{y})} = \frac{h(\mathbf{y})}{h(\mathbf{x})} \tag{4.9}$$

Ce résultat est logique puisqu'il correspond à l'algorithme de Metropolis Hastings usuel. Nous allons voir maintenant quelques unes des transformations de ce type que nous avons implémentées.

4.2.2 Translation.

La translation est la transformation la plus immédiate. On considère l'état $\mathbf{x} = \{u_1, \ldots, u_{n(\mathbf{x})}\}$.

1. On choisit l'un des u_i avec une probabilité uniforme en $\frac{1}{n(\mathbf{x})}$.
2. On génère $z = (z_X, z_Y) \in \Sigma = [-\delta_X; +\delta_X] \times [-\delta_Y; +\delta_Y] \subseteq \mathbb{R}^2$
3. On condidère la fonction ζ telle que :
$$\zeta(.;z) : u = \begin{pmatrix} X \\ Y \\ l \\ L \\ \theta \end{pmatrix} \rightarrow v = \begin{pmatrix} X + z_X \\ Y + z_Y \\ l \\ L \\ \theta \end{pmatrix} \tag{4.10}$$
4. on remplace u_i par $\zeta(u+i,z)$. En fait il se peut que $\zeta(u_i,z) \notin U$. Il y a alors deux approches valables :
   - La première consiste à rejeter ce cas là : si $v \notin u$ alors on met le taux d'acceptance à $\alpha = 0$.
   - La deuxième consiste à mettre dans ζ un "modulo U". Pour cela il faut juste que $\delta_X < \frac{X_{\max}}{2}$ pour pouvoir assurer la condition (4.7) (Cela revient à considérer S comme un tore.).

   Nous avons adopté la seconde solution puisqu'elle évite de rejeter la proposition.

4.2.3 Dilatation.

Le principe est le même que pour la translation. Cette fois ci : $z = (z_l, z_L) \in \Sigma = [-\delta_l; +\delta_l] \times [-\delta_L; +\delta_L]$ et ζ :

$$\zeta(.;z) : u = \begin{pmatrix} X \\ Y \\ l \\ L \\ \theta \end{pmatrix} \rightarrow v = \begin{pmatrix} X \\ Y \\ l + z_l \\ L + z_L \\ \theta \end{pmatrix} \tag{4.11}$$

A nouveau, il convient de faire attention aux cas où $v \notin U$.

4.2.4 Rotation

Le principe est le même que pour la translation. Cette fois ci : $z = \theta \in \Sigma = [-\delta_\theta; +\delta_\theta] \subseteq \mathbb{R}$ et ζ :

$$\zeta(.;z) : u = \begin{pmatrix} X \\ Y \\ l \\ L \\ \theta \end{pmatrix} \rightarrow v = \begin{pmatrix} X \\ Y \\ l \\ L \\ \theta + \delta_\theta \end{pmatrix} \tag{4.12}$$

4.2.5 Paramètres de ces transformations.

Les paramètres de toutes ces transformations élémentaires sont règlés arbitrairement de manière à ce que les transformations soient très locales.

4.3 Transformations complexes.

Nous allons présenter les transformations un peu plus complexes que nous avons mises en place.

4.3.1 Naissance ou mort d'une clique.

Puisque notre densité est construite sur des statistiques concernant les relations, il nous a paru intéressant d'avoir des transformations qui font naitre ou mourir des relations. Nous nous sommes intéressé au cas des relations simples et binaires $c(R_s) = 2$. En particulier, l'objectif était de mettre en place une transformation qui fasse naitre des alignements entre deux maisons dans leur longueur. C'est la relation R_{s4} qui nous intéresse donc.

L'algorithme.

On considère donc une relation simple, binaire $R \in \mathcal{R}$. L'état $\mathbf{x} = \{u_1, \ldots, u_{n(\mathbf{x})}\}$.

On considère l'algorithme suivant :

Naissance d'une clique :

1. On choisit u parmis les u_i avec une loi de probabilité $j_b^{\mathbf{x}}(u_i)$. ($\sum_i j_b^{\mathbf{x}}(ui) = 1$).
2. On génère v tel que $\{u,v\} \in R(\mathbf{x} \cup v)$
3. On calcule le taux de Green correspondant.
4. On accepte ou on rejette $\mathbf{y} = \mathbf{x} \cup v$.

Nous ne précisons pas la distribution j_b. En effet, nous voulons faire apparaitre une clique. Il parait donc pertinent d'être libre sur le choix de cette distribution.

Il est également intéressant de remarquer que la naissance d'une clique présentée ci-dessus peut faire naitre bien plus d'une clique. C'est un élément qu'il va falloir faire intervenir dans l'expression du taux de Green.

Mort d'une clique.

1. On choisit u parmi $\mathbf{x}$ avec une loi de probabilité $j_d^{\mathbf{x}}(u_i)$. On impose une condition sur cette distribution : il faut que l'on soit sûr de choisir un point appartenant à au moins une clique.
2. On calcule le taux de Green correspondant.
3. On accepte ou on rejette $\mathbf{y} = \mathbf{x} \setminus u$.

On va également se donner un outil important pour la suite : l'ensemble $\tilde{V}(R,\mathbf{x})$ qui est une extension logique de la notion de voisinage.

Définition *Pour une configuration* $\mathbf{x} \in N_U^f$ *et une relation* $R \in \mathcal{R}$ *données, on appelle* **voisinage potentiel** *de* $\mathbf{x}$ *le sous ensemble* $\tilde{V}$ *de* U *défini par :*

$$\tilde{V}(R,\mathbf{x}) = \{u \in U \quad tq \qquad u \in V(R,\mathbf{x} \cup u)\} \tag{4.13}$$

Pour une configuration $\mathbf{x}$ donnée, cet ensemble $\tilde{V}$ représente donc tous les points de l'espace d'état U qui appartiendraient à l'une de ses cliques s'ils étaient ajoutés à la configuration . De la même manière on définit $\tilde{V}(R,u)$:

Définition | *On appelle* **voisinage potentiel d'un point** u *le sous ensemble* $\tilde{V}(R,u)$ *de* U *tel que :*

$$\tilde{V}(R,u) = \{v \in U \quad tq \qquad v \in V(R,\{u,v\})\}$$

La mesure ξ.

On considère A et B deux ensembles mesurables tels qu'il existe un entier n avec $A \subseteq N_n^f$ et $B \subseteq N_{n+1}^f$. On définit ensuite la mesure ξ sur ces ensembles par les deux équations suivantes :

$$\xi(A \times B) = \int_{\mathcal{C}} \mathbf{1}_A(\mathbf{x}) \int_U \mathbf{1}_B(\mathbf{x} \cup u)\ \mathbf{1}_{\tilde{V}(R,\mathbf{x})}(u)\ \nu(du)\ \mu(d\mathbf{x})$$

Où $\nu(.) = (\lambda \times \mathbf{P}_m)(.)$,
et pour $A \subseteq N_{n+1}^f$ et $B \subseteq N_n^f$:

$$\begin{aligned}
\xi(A \times B) &= \int_{\mathcal{C}} \mathbf{1}_A(\mathbf{x}) \sum_{u \in \mathbf{x}} \mathbf{1}_{V(R,\mathbf{x})}(u)\ \mathbf{1}_B(\mathbf{x} \setminus u)\ \mu(d\mathbf{x}) \\
&= \int_{\mathcal{C}} \mathbf{1}_A(\mathbf{x}) \sum_{u \in \mathbf{x}} \mathbf{1}_{\tilde{V}(R,\mathbf{x})}(u)\ \mathbf{1}_B(\mathbf{x} \setminus u)\ \mu(d\mathbf{x})
\end{aligned}$$

Il faut montrer que cette mesure ξ est symétrique. On prend donc A et B dans $\mathcal{C} = N_U^f$ et on considère $A_n = A \cap N_n^f$ et $B_{n-1} = B \cap N_{n-1}^f$.
La démonstration suivante est très inspirée des outils de la partie 1.3.2. En particulier on se ramène à des n-uplets sur U^n mesuré par la mesure produit $\nu^n(.)$.
On note $\tilde{\mathbf{x}}$ la configuration $\{u_1,\ldots,u_{n(\tilde{\mathbf{x}})}\}$ associée au n-uplet : $\mathbf{x} = (u_1,\ldots,u_n)$

$$\begin{aligned}
\xi(A_n \times B_{n-1}) &= \frac{e^{-\nu(U)}}{n!} \int_{U^n} \mathbf{1}_{A_n}\ (\tilde{\mathbf{x}}) \sum_{u \in \tilde{\mathbf{x}}} \mathbf{1}_{\tilde{V}(R,\tilde{\mathbf{x}})}(u) \mathbf{1}_{B_{n-1}}(\tilde{\mathbf{x}} \setminus u)\ \ \nu^n(d\mathbf{x}) \\
&= \frac{e^{-\nu(U)}}{n!} \int_{U^n} n\ \mathbf{1}_{A_n}\ (\{u_1,\ldots,u_n\})\ \ \mathbf{1}_{B_{n-1}}\ (\{u_1,\ldots,u_{n-1}\}) \\
&\qquad \mathbf{1}_{\tilde{V}(R,\{u_1,\ldots,u_{n-1}\})}(u_n)\ \ \nu^{n-1}(du_1 \cdots du_{n-1})\ \nu(du_n) \\
&= \frac{e^{-\nu(U)}}{(n-1)!} \int_{U^{n-1}} \int_U \mathbf{1}_{B_{n-1}}(\tilde{\mathbf{y}})\ \mathbf{1}_{A_n}(\tilde{\mathbf{y}} \cup v)\ \mathbf{1}_{\tilde{V}(R,\tilde{\mathbf{y}})}(v)\ \ \nu^{n-1}(d\mathbf{y})\ \nu(dv) \\
&= \frac{e^{-\nu(U)}}{(n-1)!} \int_{U^{n-1}} \mathbf{1}_{B_{n-1}}(\tilde{\mathbf{y}}) \int_U \mathbf{1}_{A_n}(\tilde{\mathbf{y}} \cup v)\ \mathbf{1}_{\tilde{V}(R,\tilde{\mathbf{y}})}(v)\ \ \nu(dv)\ \nu^{n-1}(d\mathbf{y}) \\
&= \xi(B_{n-1} \times A_n)
\end{aligned}$$

Ce qui suffit pour montrer que ξ est symétrique. Il reste donc à prouver l'absolue continuité et à trouver les dérivées de Radon-Nikodym. Commençons par écrire le noyau de la transformation considérée :

Noyau de proposition.

On écrit ce noyau :

$$Q(\mathbf{x},A) = p_b Q_b(\mathbf{x},A) + p_d Q_d(\mathbf{x},A)$$

où Q_b représente la partie du noyau correspondant à la naissance d'un point tandis que Q_d représente la partie correspondant à la mort d'une clique. Développons l'expression de Q_b sous la forme suivante :

$$Q_b(\mathbf{x},A) = \sum_{u \in \mathbf{x}} j_b^{\mathbf{x}}(u)\ q_b^u(\mathbf{x})$$

Cette forme permet d'éxprimer le fait que l'on choisit d'abord un point de $\mathbf{x}$ qui sera élément de la clique construite. Pour pouvoir maintenant exprimer plus en détail ce noyau, il nous faut détailler le moyen de générer une clique.

On considère pour cela une famille de difféomorphismes $(\eta_u)_u$. Pour chaque $u \in U$, la bijection η_u nous permet de générer un point $v = \eta_u(z)$, où z est définit comme précédemment sur Σ, sous ensemble borné de $\mathbb{R}^d$.

On impose comme contrainte que :

$$\eta_u(\Sigma) = \tilde{V}(R,u) \tag{4.14}$$

Nous allons présenter tout de suite le type de difféomorphisme η_u que nous avons utilisé. Pour cela commençons par rappeler la construction de la relation R_{s4} :

$$\{u_i,u_j\} \in R_{s4}(\mathbf{x}) \quad \text{ssi} \quad \begin{cases} d_S(u_i,u_j) \leq d_{\max} \\ |\varphi| \leq \Delta\phi_{\max} \\ |\psi| \leq \Delta\phi_{\max} \end{cases}$$

Nous avons pris pour Σ :

$$\Sigma = [l_{\min},l_{\max}] \times [L_{\min},L_{\max}] \times [-d_{\max 1},d_{\max 1}] \times [-\Delta\phi_{\max}, + \Delta\phi_{\max}] \times [-\Delta\phi_{\max}, + \Delta\phi_{\max}]$$

et $z \in \Sigma$ sécrit :

$$z = \begin{pmatrix} l \\ L \\ d \\ \phi \\ \psi \end{pmatrix}$$

On pose alors :

$$\text{Si } u = \begin{pmatrix} X \\ Y \\ l \\ L \\ \theta \end{pmatrix} \qquad \begin{array}{cccc} \eta_u : & \Sigma & \rightarrow & \tilde{V}(R,u) \\ & z = \begin{pmatrix} l \\ L \\ d \\ \phi \\ \psi \end{pmatrix} & \rightarrow & v = \begin{pmatrix} X + d\cos(\phi+\theta) \\ Y + d\sin(\phi+\theta) \\ l \\ L \\ \theta+\phi+\psi \end{pmatrix} \end{array} \tag{4.15}$$

De par la définition de Σ et de R_{s4}, cette application est surjective. Elle est également injective dès lors que $\Delta\phi_{\max}$ est suffisamment petit.
De manière évidente, les η_u sont donc des difféomorphismes.

On va pouvoir écrire maintenant l'expression du noyau Q_b. Pour cela nous travaillons d'abord sur les $q_b^u(\mathbf{x},A)$ qui sont liés aux η_u. La variable aléatoire Z à valeurs dans Σ a pour loi de probabilité :

$$\mathbf{P}_Z(.) = \frac{\lambda(.)}{\lambda(\Sigma)}$$

où $\lambda(.)$ est la mesure de lebesgue.

$$\begin{aligned} q_b^u(\mathbf{x},A) &= \mathbf{P}_Z(\mathbf{x} \cup \eta_u(Z) \in A) \\ &= \mathbf{P}_Z(\eta_u(Z) \in A_{\mathbf{x}}) \end{aligned}$$

en considérant l'ensemble $A_{\mathbf{x}} \subseteq U$:

$$A_{\mathbf{x}} = \{v \in U \quad \text{tq} \qquad \mathbf{x} \cup v \in A\}$$

Ce qui permet d'écrire :

$$\begin{aligned}
q_b^u(\mathbf{x},A) &= \int_\Sigma \mathbf{1}_{A_\mathbf{x}}(\eta_u(z))\; d\mathbf{P}_Z(z) \\
&= \int_\Sigma \mathbf{1}_{A_\mathbf{x}}(\eta_u(z))\; \frac{\lambda(dz)}{\lambda(\Sigma)} \\
&= \int_{\eta_u(\Sigma)} \mathbf{1}_{A_\mathbf{x}}(v)\; \frac{1}{\lambda(\Sigma)}\; d\lambda \bullet \eta_u(v) \\
&= \int_{\eta_u(\Sigma)} \mathbf{1}_{A_\mathbf{x}}(v)\; \frac{1}{\lambda(\Sigma)}\; |J_{\eta_u^{-1}}(v)|\; \lambda(dv) \\
&= \int_{\tilde{V}(R,u)} \mathbf{1}_{A_\mathbf{x}}(v)\; \frac{1}{\lambda(\Sigma)}\; |J_{\eta_u^{-1}}(v)|\; \lambda(dv)
\end{aligned}$$

On introduit maintenant la fonction Λ_u définie sur U comme suit :

$$\boxed{\Lambda_u(v) = \begin{cases} 0 & \text{si} \quad v \notin \tilde{V}(R,u) = \eta_u(\Sigma) \\ |J_{\eta_u^{-1}}(v)| & \text{sinon} \end{cases}} \tag{4.16}$$

Cela permet donc d'écrire :

$$q_b^u(\mathbf{x},A) = \frac{1}{\lambda(\Sigma)} \int_U \mathbf{1}_{A_\mathbf{x}}(v)\; \Lambda_u(v)\lambda(dv) \tag{4.17}$$

Et par suite :

$$Q_b(\mathbf{x},A) = \frac{1}{\lambda(\Sigma)} \int_U \sum_{u\in\mathbf{x}} j_b^\mathbf{x}(u)\; \mathbf{1}_A(\mathbf{x}\cup v)\; \Lambda_u(v)\lambda(dv) \tag{4.18}$$

La partie du noyau correspondant à la mort s'écrit de façon beaucoup plus simple :

$$Q_d(\mathbf{x},A) = \sum_{u\in\mathbf{x}} j_d^\mathbf{x}(u)\mathbf{1}_A(\mathbf{x}\setminus u) \tag{4.19}$$

On rappelle que nous avons imposé une contrainte sur les $j_d^\mathbf{x}$:

$$\text{Si} \;\; u \notin V(R,\mathbf{x}) \;\; \text{alors} \;\; j_d^\mathbf{x}(u) = 0 \tag{4.20}$$

Dérivées de Radon-Nikodym.

On va chercher l'expression de la dérivée de Radon Nykodym de $\pi(d\mathbf{x})Q(\mathbf{x},d\mathbf{y})$. Pour cela il faut d'abord montrer que cette mesure est absolument continue par rapport à $\xi(.;.)$. Cela est immédiat du fait des contraintes (4.20) et (4.14) imposées au noyau de proposition Q.

Une fois de plus, la dérivée de Radon Nikodym $f(\mathbf{x},\mathbf{y})$ est définie sur des supports disjoints : il faut distinguer les cas où la configuration $\mathbf{y}$ contient un élément de plus que $\mathbf{x}$ des cas où elle en contient un de moins. On considère donc A et B deux ensembles de la σ algèbre construite sur $\mathcal{C} = N_U^f$. On note alors $A_n = A \cap N_n^f$.

On regarde donc ici les "cas de naissance" :

$$\begin{aligned}
\int_{A_n}\int_{B_{n+1}} \pi(d\mathbf{x})\; Q(\mathbf{x},d\mathbf{y}) &= \frac{1}{\lambda(\Sigma)} \int_\mathcal{C} \mathbf{1}_{A_n}(\mathbf{x}) \int_U p_b\; h(\mathbf{x}) \sum_{u\in\mathbf{x}} j_b^\mathbf{x}(u)\mathbf{1}_{B_{n+1}}(\mathbf{x}\cup v)\; \Lambda_u(v)\; \lambda(dv)\mu(d\mathbf{x}) \\
&= \frac{e^{-\nu(U)}}{n!} \frac{1}{\lambda(\Sigma)} \int_{U^n} \mathbf{1}_{A_n}(\tilde{\mathbf{x}}) \int_U p_b\; h(\tilde{\mathbf{x}}) \sum_{u\in\tilde{\mathbf{x}}} j_b^{\tilde{\mathbf{x}}}(u)\; \mathbf{1}_{B_{n+1}}(\tilde{\mathbf{x}})\; \Lambda_u(v) \\
&\qquad \nu^n(du_1,\ldots,du_n)\lambda(dv)
\end{aligned}$$

Or, on a pris comme mesure d'intensité pour le processus ponctuel de référence : $\nu(.) = (\lambda \times P_m)(.)$ avec comme distribution de probabilité sur les marques la loi uniforme :

$$\lambda(dv) = \nu(dv).\lambda(\mathcal{K}) \tag{4.21}$$

Ce qui donne:

$$\int_{A_n} \int_{B_{n+1}} \pi(d\mathbf{x})\; Q(\mathbf{x},d\mathbf{y}) = \frac{e^{-\nu(U)}}{n!} \frac{\lambda(\mathcal{K})}{\lambda(\Sigma)} \int_{U^n} \mathbf{1}_{A_n}(\tilde{\mathbf{x}}) \int_U p_b\; h(\tilde{\mathbf{x}}) \sum_{u \in \tilde{\mathbf{x}}} j_b^{\tilde{\mathbf{x}}}(u)\; \mathbf{1}_{B_{n+1}}(\tilde{\mathbf{x}})\; \Lambda_u(v) \\ \nu^n(du_1,\dots,du_n)\nu(dv)$$

Ce qui nous permet de conclure quant à la dérivée, dans le cas d'une naissance :

$$\boxed{f(\mathbf{x},\mathbf{x} \cup v) = \frac{\lambda(\mathcal{K})}{\lambda(\Sigma)}\; h(\mathbf{x})\; p_b \left(\sum_{u \in \mathbf{x}} j_b^{\mathbf{x}}(u)\; \Lambda_u(v) \right)} \tag{4.22}$$

Cette expression est intéressante : en effet la somme sur les u contient des termes nuls : tous les u tels que $\{u,v\}$ ne soit pas une clique au sens de R sont éliminés.
Finalement, le jacobien usuel est remplacé par une espérance de jacobien.
En faisant naitre une clique, on peut en faire apparaitre plusieurs. Cela signifie que le point que l'on a fait apparaitre aurait pu être généré à partir d'autres triplet (u,η_u,z). Cette espérance de jacobien tient compte de tous les triplets possibles.

De manière plus générale, on doit retrouver ce genre d'expression dès lors que l'on fait un choix sur les u et que l'on utilise ensuite des difféomorphismes η_u dont les ensembles d'arrivée ne sont pas disjoints.

Il faut également noter la complication apportée par l'utilisation de processus ponctuels marqués : ici le terme en λK apparaît parce que l'on utilise $P_m(.)$ au lieu de $\lambda(.)$. Il est malgré tout utile de voir à quel endroit ce choix de loi joue un rôle.

Considérons maintenant le cas de la mort d'une clique.

$$\int_{A_{n+1}} \int_{B_n} \pi(d\mathbf{x})\; Q(\mathbf{x},d\mathbf{y}) = \frac{e^{-\nu(U)}}{(n+1)!} \int_{U^{n+1}} \mathbf{1}_{A_{n+1}}(\tilde{\mathbf{x}})\; p_d\; h(\tilde{\mathbf{x}}) \sum_{u \in \tilde{\mathbf{x}}} j_d^{\tilde{\mathbf{x}}}\; \mathbf{1}_{B_n}(\tilde{\mathbf{x}} \setminus u)\; \nu^{n+1}(du_1,\dots,du_{n+1})$$

Or, de part la condition 4.20, on obtient :

$$\int_{A_{n+1}} \int_{B_n} \pi(d\mathbf{x})\; Q(\mathbf{x},d\mathbf{y}) = \frac{e^{-\nu(U)}}{(n+1)!} \int_{U^{n+1}} \mathbf{1}_{A_{n+1}}(\tilde{\mathbf{x}})\; p_d\; h(\tilde{\mathbf{x}}) \sum_{u \in \tilde{\mathbf{x}}} j_d^{\tilde{\mathbf{x}}}\; \mathbf{1}_{\tilde{V}(R,\tilde{\mathbf{x}})}(u)\mathbf{1}_{B_n}(\tilde{\mathbf{x}} \setminus u)\; \nu^{n+1}(du_1,\dots,du_{n+1})$$

On en déduit :

$$\boxed{f(\mathbf{x},\mathbf{x} \setminus u) = h(\mathbf{x})\; p_d\; j_d^{\mathbf{x}}(u)} \tag{4.23}$$

Coefficients de Green.

En utilisant l'expression de la dérivée f sur ses différents supports, on peut calculer les coefficients de Green dans le cas d'une naissance et dans le cas d'une mort :

$$\boxed{R(\mathbf{x},\mathbf{x} \cup v) = \frac{h(\mathbf{x} \cup v)}{h(\mathbf{x})} \frac{p_d}{p_b} \frac{\lambda(\Sigma)}{\lambda(\mathcal{K})} \frac{j_d^{\mathbf{x} \cup v}(v)}{\sum_{u \in \mathbf{x}} j_b^{\mathbf{x}}\; \Lambda_u(v)}} \tag{4.24}$$

Il faut remarquer que cette expression pose des problèmes informatiques :

- Le numérateur fait intervenir la probabilité de choisir v parmis $\mathbf{x} \cup v$ pour le tuer. Celle ci ne peut être calculée qu'une fois v connu, c'est à dire une fois v généré.

- Le dénominateur demande de parcourir toutes les cliques au sens de R de l'état $\mathbf{x}$. Cela suppose donc que l'on puisse avoir facilement accès à l'ensemble de ces cliques.
- On a volontairement laissé les distributions discrètes de probabilité j_d et j_b quelconques : ces distributions doivent être choisies pour des raisons informatiques et non pour des raisons mathématiques. En particulier en ce qui concerne j_d, puisqu'il y a une condition de nullité sur les $u \in \mathbf{x}$ qui ne sont pas dans au moins une clique.

Dans le cas d'une mort, on obtient :

$$\boxed{R(\mathbf{x},\mathbf{x} \setminus v) = \frac{h(\mathbf{x} \setminus v)}{h(\mathbf{x})} \frac{p_b}{p_d} \frac{\lambda(\mathcal{K})}{\lambda(\Sigma)} \frac{\sum_{u \in \mathbf{x} \setminus v} j_b^{\mathbf{x} \setminus v} \Lambda_u(v)}{j_d^{\mathbf{x}}(v)}} \tag{4.25}$$

Derniers calculs.

En pratique, nous avons pris :

$$j_b^{\mathbf{x}}(u) = \frac{1}{n(\mathbf{x})}$$

Ce qui correspond à une loi uniforme sur les points pour la naissance. Pour la mort, nous avons choisi d'utiliser :

$$j_d^{\mathbf{x}}(u) = \frac{1}{n_V(R,\mathbf{x})}$$

ce qui revient à choisir de manière uniforme l'élément à tuer dans l'ensemble $V(R,\mathbf{x})$. Il nous semble qu'il serait plus pertinent du point de vue informatique de choisir une clique au hasard, puis de choisir ensuite l'un des deux éléments de cette clique. Cela permettrait en outre de tenir compte de la qualité des cliques et de tuer une clique de mauvaise fonction de qualité.

Reste enfin le problème du jacobien ; un rapide calcul montre que dans notre cas :

$$|J_{\eta_u^{-1}}(v)| = \frac{1}{d(u,v)}$$

4.3.2 Naissance ou mort d'un ou plusieurs objets.

Nous allons maintenant nous intéresser plus particulièrement au cas de la transformation dite du "**split and merge**". Pour cette transformation nous allons utiliser directement la formule précédente en se ramenant à une transformation sur des cliques. Les points intéressant de cette transfomation sont les suivants :

- d'abord la construction du noyau de proposition.
- ensuite l'utilisation de la notion de relation simple pour faciliter la mise en oeuvre de cette transformation.

L'objectif de cette transformation est de faire fusionner les maisons qui sont proches et qui se ressemblent.
Evidemment, la tranformation inverse qu'il nous est nécessaire d'implémenter pour assurer la condition 2.19 est la division d'une maison en deux maisons.

Pour cela nous considérons un d'arbre de relations : pour savoir si deux maisons sont fusionnables, nous utilisons :

- La relation R_{s4} d'alignement dans la longueur.
- Une relation supplémentaire, R_{s5}, binaire n'intervenant pas dans la densité du modèle a priori, mais dont les cliques doivent être des cliques de R_{s4} et obéir à un critère d'homogénéité.

De la même manière, on s'intéresse aux maisons divisibles en introduisant une relation R_{s6} dont la cardinalité est $c(R_{s6}) = 1$.

Conditions.

Nous considérons deux maisons u_i et u_j de $U = S \times \mathcal{K}$. Voici la liste des conditions que nous imposons pour considérer que les deux maisons sont fusionnables :

$$\begin{array}{lll}
\text{1) alignement dans la longueur} & \left.\begin{array}{l} d_S(u_i,u_j) \leq d_{\max} \\ |\varphi| \leq \Delta\phi_{\max} \\ |\psi| \leq \Delta\phi_{\max} \end{array}\right\} \Leftrightarrow \{u_i,u_j\} \in R_{s4}(\mathbf{x}) & \\
\text{2) homogénéïté} & |l_i - l_j| \leq \delta_l & \\
\text{3) longueur totale} & L_i + L_j \leq L_{\max} & \\
\text{4) position} & \left| d(u_i,u_j) - \frac{L_i+L_j}{2} \right| \leq \delta_p &
\end{array} \tag{4.26}$$

Il faut juste remarquer que la condition 4 implique une autre condition :

$$d_S(u_i,u_j) \leq \delta_p + L_{\max}$$

Les conditions (4.26) définissent une relation simple binaire :

$$\{u_i,u_j\} \in R_{s5}(\mathbf{x}) \Leftrightarrow \ u_i \text{ et } u_j \text{ vérifient (4.26)} \tag{4.27}$$

Les maisons divisables sont définies par une seule condition. Nous l'utilisons pour construire la relation simple R_{s6} qui est une relation de cardinalité constante valant 1.

$$\{u_i\} \in R_{s6}(\mathbf{x}) \Leftrightarrow L_i > 2L_{\min} \tag{4.28}$$

Transformation de fusion.

Nous commençons par présenter cette transformation puisque c'est elle qui donne les conditions à respecter dans la division :

$$\begin{array}{llll} \tilde{\zeta}: & U \times U & \rightarrow & U \\ & (u,v) & \rightarrow & w \end{array}$$

telle que :

$$\begin{array}{ll}
x_w = \begin{pmatrix} X_w \\ Y_w \end{pmatrix} & = \text{Barycentre}\,((x_u,L_u),(x_v,L_v)) \\
& = \frac{L_u}{L_u+L_v} \begin{pmatrix} X_u \\ Y_u \end{pmatrix} + \frac{L_v}{L_u+L_v} \begin{pmatrix} X_v \\ Y_v \end{pmatrix} \\
\theta_w & = dir(\overrightarrow{x_u - x_v}) \pmod{\pi} \\
l_w & = \frac{l_u+l_v}{2} \\
L_w & = L_u + l_v
\end{array}$$

Il faut remarquer que cette transformation est symétrique en u et v. Elle s'applique donc directement à la clique $\{u,v\}$ et non au couple (u,v).

Transformation de division.

En fait, c'est cette transformation qui est importante. La transformation $\tilde{\zeta}$ sert à définir l'ensemble de départ de la vraie transformation ζ considérée.
Il faut donc que $\tilde{\zeta}$ soit une restriction de ζ^{-1}.
On veut que ζ soit un difféomorphisme, il faut donc que cette transformation soit surjective, d'où l'importance des conditions 4.26. Rappelons également, que pour que ζ puisse être une bijection, il faut utiliser 5 paramètres aléatoires.

On considère $w \in U$, tel que $L_w \geq 2L_{\min}$. On va générer u et v :

$$\begin{cases} L_u & = L_{\min} + z_1(L_w - 2L_{\min}) \\ L_v = L_w - L_u & = L_{\min} + (1 - z_1)(L_w - 2L_{\min}) \end{cases} \quad z_1 \sim \mathcal{U}([0,1])$$

$$\begin{cases} l_u & = l_w + z_2 \\ l_v = 2l_w - l_u & = l_w - z_2 \end{cases} \quad z_2 \sim \mathcal{U}([-\delta_l,\delta_l])$$

$$\begin{cases} \theta_u & = \theta_w + z_3 \\ \theta_v & = \theta_w + z_4 \end{cases} \quad z_3,z_4 \sim \mathcal{U}([-\Delta\phi_{\max}, + \Delta\phi_{\max}])$$

Il reste à fixer les points x_u et x_v. La condition de barycentre donne :

$$\begin{cases} \overrightarrow{x_u x_w} = \dfrac{L_u}{L_u + l_v} \overrightarrow{x_u x_v} \\ \overrightarrow{x_w x_v} = \dfrac{L_v}{L_u + L_v} \overrightarrow{x_u x_v} \end{cases}$$

On va donc générer le vecteur $\overrightarrow{x_u x_v}$. Or sa direction est connue. Donc seule sa mesure algébrique $\overline{x_u x_v}$ nous intéresse , et on prend :

$$\overline{x_u x_v} = L_u + L_v + z_5 \qquad z_5 \sim \mathcal{U}([-\delta_p, + \delta_p])$$

Ce qui donne :

$$\begin{cases} X_u & = X_w - (L_{\min} + z_1(L_w - 2L_{\min})) \ \dfrac{L_w + z_5}{L_w} \cos(\theta_w) \\ Y_u & = Y_w - (L_{\min} + z_1(L_w - 2L_{\min})) \ \dfrac{L_w + z_5}{L_w} \sin(\theta_w) \end{cases}$$

$$\begin{cases} X_v & = X_w + (L_{\min} + (1 - z_1)(L_w - 2L_{\min})) \ \dfrac{L_w + z_5}{L_w} \cos(\theta_w) \\ Y_u & = Y_w + (L_{\min} + (1 - z_1)(L_w - 2L_{\min})) \ \dfrac{L_w + z_5}{L_w} \sin(\theta_w) \end{cases}$$

Il se trouve que le jacobien de la transformation $\zeta : (w,x) \rightarrow (u,v)$ est assez compliqué à calculer directement.

Deuxième transformation de division.

On procède donc à une série de changements de variables et on génére z' conditionnellement à w.
On pose :

$$\begin{aligned} z'_1 & = L_{\min} + z_1(L_w - 2L_min) \\ z'_2 & = z_2 \\ z'_3 & = \theta_w + z_3 \\ z'_4 & = \theta_w + z_4 \\ z'_5 & = \frac{L_w + z_5}{L_w} \end{aligned}$$

Il se trouve qu'ici les changements de variables sont linéaires. Par conséquent, l'uniformité de Z se transporte sur Z'. D'une manière plus générale, même si ces changements de variables n'étaient pas linéaires, il suffirait de poser Z' uniforme, et de prendre pour la loi de Z la loi image par la transformation inverse, avec comme contrainte sur la loi de Z' que la loi de Z ainsi obtenue ait le même support que précédemment.

$$\begin{aligned} & z'_1 \sim \mathcal{U}([L_{\min},L_w - L_{\min}]) \\ & z'_2 \sim \mathcal{U}([-\delta_l,\delta_l]) \\ & z'_3 \sim \mathcal{U}([\theta_w - \Delta\phi_{\max},\theta_w + \Delta\phi_{\max}]) \\ & z'_4 \sim \mathcal{U}([\theta_w - \Delta\phi_{\max},\theta_w + \Delta\phi_{\max}]) \\ & z'_5 \sim \mathcal{U}\left([\frac{L_w - \delta_p}{L_w}, \frac{L_w + \delta_p}{L_w}]\right) \end{aligned}$$

Ce qui revient à définir Σ' et à prendre Z' de loi uniforme sur Σ'. La nouvelle transformation est donc donnée par :

$$\begin{cases} L_u &= z'_1 \\ L_v &= L_w - z'_1 \end{cases}$$

$$\begin{cases} l_u &= l_w + z'_2 \\ l_v &= l_w - z'_2 \end{cases}$$

$$\begin{cases} \theta_u &= z'_3 \\ \theta_v &= z'_4 \end{cases}$$

$$\begin{cases} X_u &= X_w - z'_1\ z'_5\ \cos\theta_w \\ Y_u &= Y_w - z'_1\ z'_5\ \sin\theta_w \end{cases}$$

$$\begin{cases} X_v &= X_w + (L_w - z'_1)\ z'_5\ \cos\theta_w \\ Y_v &= Y_w + (L_w - z'_1)\ z'_5\ \sin\theta_w \end{cases}$$

Ce qui donne pour le jacobien :

$$|J_{\zeta^{-1}}| = \frac{1}{z'_5\ L_w^2} = \frac{L_u + L_v}{d_S(u,v)} \tag{4.29}$$

Taux de Green.

On écrit le noyau sous la forme :

$$Q(\mathbf{x},A) = p_s Q_s(\mathbf{x},A) + p_m Q_m(\mathbf{x},A)$$

On choisit donc d'abord si l'on effectue une fusion ou une division.
Dans le cas d'une **division :**

1. On choisit un objet w parmi les objets divisables. Cela revient à choisir une clique au sens de la relation R_{s6}. On note la distribution de probabilité correspondante : $j_s^{\mathbf{x}}(w)$,
2. On génère z' de loi uniforme sur Σ'.
3. On calcule $(u,v) = \zeta(w,z')$,
4. On calcule le taux de Green,
5. On accepte ou on rejette $\mathbf{y} = (\mathbf{x} \setminus w) \cup u \cup v$.

Dans le cas d'une **fusion :**

1. On choisit une clique $\{u,v\}$ au sens de la relation R_{s5},
2. On calcule $w = \tilde{\zeta}(\{u,v\})$,
3. On calcule le taux de Green,
4. On accepte ou on rejette $\mathbf{y} = (\mathbf{x} \setminus (u \cup v)) \cup w$.

Pour calculer le taux de Green il faut faire intervenir la probabilité de choisir la clique $\{u,v\}$ dans $R_{s5}(\mathbf{x})$. On note cette probabilité $j_m^{\mathbf{x}}(\{u,v\})$. Le premier taux de Green, dans le cas d'une division, est alors donné par l'équation suivante, en notant $\mathbf{y} = \mathbf{x} \setminus w \cup u \cup v$

$$\boxed{R_{split}(\mathbf{x},\mathbf{y}) = \frac{h(\mathbf{y})}{h(\mathbf{x})} \frac{p_m}{p_s} \frac{\lambda(\Sigma'_w)}{\lambda(\mathcal{K})} \frac{j_m^{\mathbf{y}}(\{u,v\})}{j_s^{\mathbf{x}}(w)} \frac{d_S(u,v)}{L_u + L_v}} \tag{4.30}$$

On pose maintenant $\mathbf{y} = \mathbf{x} \setminus (u \cup v) \cup w$:

$$\boxed{R_{merge}(\mathbf{x},\mathbf{y}) = \frac{h(\mathbf{y})}{h(\mathbf{x})} \frac{p_s}{p_m} \frac{\lambda(\mathcal{K})}{\lambda(\Sigma'_w)} \frac{j_s^{\mathbf{y}}(w)}{j_m^{\mathbf{x}}(\{u,v\})} \frac{L_u + L_v}{d_S(u,v)}} \tag{4.31}$$

Derniers calculs.

En pratique, nous avons pris:

$$j_s^{\mathbf{x}}(w) = \frac{1}{n_V(R_{s6},\mathbf{x})}\ \mathbf{1}_V(R_{s6},\mathbf{x})(w) \tag{4.32}$$

Par contre pour la distribution de probabilité pour la fusion, nous avons utilisé l'algorithme suivant :

1. On choisit u uniformément parmi $V(R_{s5},\mathbf{x})$,
2. On choisit v uniformément parmi $V(R_{s5},\mathbf{x},u)$,

Par conséquent, pour $\{u,v\} \in R_{s5}(\mathbf{x})$:

$$\begin{aligned}
j_m^{\mathbf{x}}(\{u,v\}) &= \mathbf{P}((u,v) \text{ ou } (v,u)) \\
&= \mathbf{P}(v|u)\mathbf{P}(u) + \mathbf{P}(u|v)\mathbf{P}(v) \\
&= \frac{1}{n_V(R_{s5},\mathbf{x},u)}\frac{1}{n_V(R_{s5},\mathbf{x})} + \frac{1}{n_V(R_{s5},\mathbf{x},v)}\frac{1}{n_V(R_{s5},\mathbf{x})} \\
&= \frac{1}{n_V(R_{s5},\mathbf{x})}\left(\frac{1}{n_V(R_{s5},\mathbf{x},u)} + \frac{1}{n_V(R_{s5},\mathbf{x},v)}\right)
\end{aligned}$$

On remarquera qu'une fois de plus, il est essentiel que le programme informatique donne accès aux grandeurs nécessaires au calcul du taux de Green.

Conclusions et perspectives.

Commençons par résumer ce que nous avons fait :

- nous avons introduit un modèle de densité de processus ponctuel permettant de considérer des interactions très variées et très diverses.
- ce modèle permet d'introduire plusieurs types d'interactions en même temps et de jouer sur les importances relatives de ces interactions assez facilement.
- nous avons écrit les transformations du noyau de proposition qui correspondent à ce modèle.
- enfin, et ce ne fut pas la part la plus mince de travail, nous avons mis en oeuvre ce modèle et ces transformations en C++.

Continuons avec ce qui n'a pas été fait :

Les problèmes que ce mémoire laisse ouverts et auxquels il va falloir s'attaquer pendant la thèse sont nombreux:

- d'abord revoir l'estimateur de Rue et en particulier l'algorithme de Recuit Simulé qui permet de minimiser la fonction de coût L. Cet algorithme n'est pas du tout adapté à notre problème car il propose trop de mouvements qui ne sont pas accepté alors que le problème est “presque convexe”.
- revoir éventuellement la métrique d'intérêt : celle proposée par Rue est beaucoup trop couteuse en temps de calcul dans notre cas.
- adapter le terme d'attache aux données au niveaux de gris et obtenir des résultats avec le maximum a posteriori sur des données réelles.
- ajouter les coordonnées tri-dimensionnelles.

Voila donc pour les perspectives.
Quant aux ultimes conclusions de ce mémoire :

- le domaine d'application est plaisant, même si les temps de calculs sont parfois très longs.
- la double implication du sujet à la fois probabiliste et informatique est vraiment très attrayante, puisqu'elle permet de jongler entre les énigmes informatiques et des objets mathématiques séduisants.
- l'ouverture du sujet en fait un sujet de thèse très intéressant : il reste encore beaucoup de choses à dire, à faire et à construire.

Bibliographie

[1] A.J. Baddeley *Errors in binary images and a L^p version of the Hausdorff metric* Nie. Arch. c. Wisk. 10 : 157-183, 1992.

[2] A. J. Baddeley and M.N.M. van Lieshout *Stochastic geometry models in high-level vision*, in Statistics and Images, Volume 1, K.V. Mardia and G.K. Kanji (Eds.) *Advances in Applied Statistics*, a supplement to *Journal of Applied Statistics* 20 : 231-256. Abingdon : Carfax, 1993.

[3] O.E. Banorff-Nielsen, W.S Kendall, M.N.M. van Lieshout (eds) *Stochastic Geometry Likehood and computation* Chapman and Hall, 1999.

[4] S. P. Brooks *Quantitative Convergence Assessment for Markov Chain Monte Carlo via Cusums* Statistics and computing, 8 pp. 267-274,1998

[5] S.P. Brooks, P, Guidici and G.O. Roberts *Efficient construction of reversible jump MCMC proposal distributions.* Research Report University of Cambridge, December 2000.

[6] D. Daley and V. Jones. *an introduction to the theory of point processes.* Springer, New York, 1988.

[7] M. Fradkin, M. Roux, and H. Maître *Building detection form multiple views.* In ISPRS Conference on Automatic Extraction of GIS Objects form Digital Imagery. 1999.

[8] M. Fradkin, M. Roux, and H. Maître and U. Leloglu *Surface reconstruction from multiple aerial images in dense urban areas.* In CVPR 1999.

[9] L. Garcin, X. Descombes, J. Zerubia and H. Le Men *Building Detection by Markov Object processes and a MCMC algorithm.* Inria Research report No 4206, 2001.

[10] A. Gelman, G.O. Roberts and W.R. Gilks *Efficient Metropolis Jumping Rule.* In J.M. Bernardo, J.O. Berger, A. P. Dawid and A.F. M. Smith (eds), *Bayesian Statistics 5*, New York: Oxford University Press, 1995.

[11] C. J. Geyer *Likehood Inference for Spatial point Processes.* In O.E. Banorff-Nielsen, W.S Kendall, M.N.M. van Lieshout (eds) *Stochastic Geometry Likehood and computation* Chapman and Hall, 1999.

[12] C.J. Geyer and J. Moller *Simulation and likehood inference for spatial point process*, Scandinavian Journal of Statistics, Series B, 21,359-373,1994.

[13] S. Girard, P. Guerin, H. Maître and M. Roux *Building detection from high resolution colour images.* Symp. on Remote Sensing Barcelona, 1998.

[14] P. Guidici and G.O. Roberts *On the automatic choice of reversible Jumps.* Research Report University di Pavia, June 1998.

[15] P.J. Green *Reversible jump markov chain Monte-Carlo computation and Bayesian model determination.* Biometrika, 57, 97-109. 1995.

[16] P.J. Green and A. Mira *Delayed rejection in reversible Jump Metropolis Hastings* Mathematics Research Report No. S-01-99, University of Bristol, 1999.

[17] A. Gruen, O. Kuebler, and P. Agouris (eds) *Automatic Extraction of Man-made Objects from Aerial and Space Image.* Birkhäuser Verlag, Basel 1995.

[18] A. Gruen , O. Kuebler, and P. Agouris, (eds) *Automatic Extraction of Man-made Objects from Aerial and Space Image (II)* .Birkhäuser Verlag, Basel 1997.

[19] A. Gruen and R. Navatia (eds). *Special Issue on " Automatic Building Extraction from Aerial Images"* Computer Vision and Image Understanding (CVIU) 72(2), 1998

[20] M. Imberty and X. Descombes *Simulation de processus objets : Etude de faisabilité pour une application à la segmentation d'images.* INRIA Research Report No 3881, 2000.

[21] P.J.M. van Laarhoven and E.H.L. Aarts, *Simulated Annealing: Theory and Applications*, Reidel, Dordrecht, 1987.

[22] M. N. M. van Lieshout *Markov Point Processes and their Applications* Imperial College Press, London, 2000

[23] H. Mayer *Automatic object extraction from aerial imagery- a survey focusing on buildings* Computer Vision and Image Understanding (CVIU), 74(2):138-149,1999

[24] K. R. Mecke and D. Stoyan (eds) *Statistical Physics and Spatial Patterns.* Springer 2000.

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

[26] P.H. Peskun *Optimum Monte Carlo sampling using Markov Chains.* Biometrika 60, 607-612. 1973

[27] C.J. Preston. *Random Fields.* Springer, Berlin 1976

[28] C.J. Preston. *Spatial birth-and-death processes* Bulletin of the International Statistical Institute, 46:371-391,1977

[29] J. G. Propp and D.B. Wilson. *Exact sampling with coupled markov chains and applications to statistical mechanics.* Random Structures and Algorithms, 9:223-252,1996.

[30] B.D. Ripley and F. P. Kelly. *Markov point processes.* Journal of the London Mathematical Society, 15:188-192, 1977.

[31] C. Robert and G. Casella *Monte Carlo Statistical Methods*, Springer-Verlag, New York, 1999.

[32] H. Rue and M. Hurn *Bayesian object identification* Biometrika, 3, pp. 649-660, 1999

[33] H. Rue and A. R. Syverseen *Bayesian Object Recognition with Baddeley's Delta Loss* Adv. Appl. Prob, 30, pp 64-84, 1998.

[34] D. Ruelle *Statistical mechanics.* Benjamin, New York, 1969

[35] D. Ruelle *Superstable Interactions in Classical Statistical Mechanics.* Commun. math. Phys. , Springer-Verlag, 18, pp. 127-159,1970.

[36] R. Stoica *Processus ponctuels pour l'extraction de réseaux linéïques dans les images satellitaires et aériennes.* Thèse de doctorat, Université de Nice Sophia-Antipolis, février 2001.

[37] D. Stoyan, W.S. Kendall, and J. Mecke. *Stochastic geometry and its applications.* Wiley, Chistester, second edition, 1995.

[38] D.J. Strauss *A model for clustering.* Biometrika, 62:467-475, 1975

[39] L. Tierney *A note on Metropolis Hasting for general state spaces.* Annals of Applied Probability, 8, 1-9, 1998.

Chapitre 5

Annexe A : Les algorithmes

Les algorithmes ont été codés en C++.

Codage de l'état

Premier niveau

Le premier point est de coder l'état. L'état consiste en une configuration $\mathbf{x}$ d'objets u_i. L'état est donc simplement un tableau d'objets, de taille dynamique.

Les relations

Le modèle suppose que l'on code efficacement des relations sur les objets.

Une première approche consiste à mettre à jour dans la mémoire les voisinages de chaque objet. Cette approche n'est d'une part pas efficace du point de vue informatique, et d'autre part elle n'est pas du tout adaptée aux relations étendues qui peuvent contenir plus de deux objets. Cela est dû au fait qu'en axant le codage des interactions sur les objets, on s'autorise à coder une clique plusieurs fois. Il est bien plus intéressant de faire en sorte que **chaque clique soit un objet unique dans la mémoire.**

L'idée est de garder en mémoires les ensembles $R(\mathbf{x})$, c'est à dire les cliques au sens de chaque relation.

A chaque objet $\mathbf{x}$, on associe donc une liste de liens $L(\mathbf{x})$ vers les cliques dont il est membre. Les éléments d'une liste sont accessibles directement grace à l'emplacement e de l'élément: $L(u_i)(e)$ donne l'élément situé en e de la liste. En l'occurence, on obtient le lien vers la clique située en e dans la liste.

Chacune de ces cliques contient la liste des liens vers les objets u_i qui la composent et pour chacun de ces objets l'emplacement e de la clique dans la liste des cliques auxquelles l'objet appartient:

On note $\tilde{u_i}$ un lien vers l'objet u_i (informatiquement c'est un pointeur). Une clique r va se coder sous la forme :

$$r = ((\tilde{u_i},e(u_i)),\ldots,(\tilde{u_j},e(u_j))$$

où les $e(u_i)$ vérifient :

$$L(u_i)\,(e(u_i)) = \tilde{r}$$

Cela permet de tuer un objet très facilement.

Supposons que l'on veuille tuer l'objet u. L'algorithme qui découle du codage ci-dessus est le suivant :

- On parcourt les cliques r auxquelles u appartient en parcourant la liste $L(u)$ qui nous donne les liens vers ces cliques,
- Pour chacune de ces cliques, on parcourt les liens $\tilde{u_i}$ vers les éléments u_i de r . L'un de ces u_i est u,
- Pour chacun de ces u_i, on supprime dans $L(u_i)$ le lien situé à l'emplacement $e(u_i)$ codé dans r.

De cette façon le temps de suppression d'une clique est linéaire par rapport à la taille de la clique et constant par rapport aux autres paramètres.

Le temps de suppression d'un objet est donc linéaire par rapport aux nombres de cliques auxquelles il appartient pour des relations de cardinalité constante.

Pour ajouter un objet, c'est simple : il suffit de détecter les cliques auxquelles il appartient et d'enregistrer dans chacunes de ces cliques en même temps que le lien vers un objet qui la compose, l'emplacement où cette clique est recensée dans la liste d'objet.

Le temps d'ajout d'un objet est donc lui aussi linéaire par rapport aux nombres de cliques auxquelles cet objet appartient.

L'autre avantage majeur de cette structure de données concerne le fait qu'il nous donne accès facilement aux données nécessaires pour les calculs de densité du modèle.

On remarquera que les emplacements e ne sont pas des numéros, mais des pointeurs sur les cellules d'une liste chainée.

Arborescence de relations

Lorsque l'on ajoute un objet u on va créer informatiquement toutes les cliques de tous les types de relations que l'on considère.

On peut accélérer les algorithmes en construisant astucieusement les relations considérées.
Prenons l'exemple de R_3. Cette relation est construite sur R_{s1}. On dit que R_3 est une fille de R_{s1}.
Lorsque l'on ajoute un objet u à l'état :

1. on parcourt tous les objets u_i pour mettre à jour $R_{s1}(\mathbf{x})$. On y ajoute toutes les cliques dont u est membre.
2. pour construire $R_3(\mathbf{x})$, il suffit de parcourir les cliques de $R_{s1}(\mathbf{x})$ qui sont désormais à jour et dont u est membre. Or ces cliques sont directement accessibles par la liste $L_{s1}(u)$. Il ne reste ensuite plus qu'à tester parmi ces cliques lesquelles donnent lieux à des cliques de R_3.

Il peut même être judicieux de créer des relations qui n'interviennent pas dans le modèle mais dont l'existence est justifiée par l'informatique.

On remarquera qu'il est plus long d'ajouter un objet que de le supprimer puisqu'il faut détecter les cliques auxquelles il appartient.

Cette remarque conduit à l'astuce suivante :

Noyau de proposition

Le point essentiel de l'algorithme MHG est la loi de Bernoulli. En effet, on propose d'abord une transformation, et on décide ensuite de l'accepter ou non.

Or une transformation se décompose essentiellement en une mort d'abord, et une naissance ensuite. Pour certaines transformations l'une de ces deux étapes n'a pas lieu, mais cela ne change rien à la problématique présentée ici.

Si elle est acceptée tout va bien. Sinon il faut tuer l'objet nouveau-né et ressusciter l'ancien.

Comme il est plus coûteux en temps de calcul de détecter des cliques que de les supprimer, il faut éviter lorsque l'on ressuscite un objet tué d'avoir à refaire un travail qui a déjà été fait par le passé .

Pour cette raison, nous avons codé une mort fictive. Lorsque l'on propose une transformation, comme la translation d'un objet u par exemple, on fait semblant de tuer u et on fait naitre u'. Pour cela, il suffit de mettre u à coté du tableau codant l'état et de rendre transparentes les cliques auxquelles il appartient puisque celles-ci ne doivent plus intervenir.

La transparence est donc juste un attribut que l'on ajoute aux cliques : pour détecter une clique fille, on vérifie que les cliques mères ne sont pas transparentes.
Si la proposition est acceptée, on tue u pour de bon. Sinon on réintègre u dans le tableau et on rend opaques les cliques auxquelles il appartient. Il ne reste alors qu'a tuer u'.

En résumé :

1. On choisit une transformation.
2. Cette transformation choisit l'objet u à transformer : c'est à dire qu'elle propose de remplacer u par u'.
3. On tue u de manière fictive. Entre autre, on rend transparentes les cliques contenant u.
4. On ajoute u', en mettant à jour ses cliques et en ne tenant compte que des cliques opaques.
5. Ces deux mises à jour permettent d'avoir accés aux grandeurs permettant de calculer le taux de Green $R(\mathbf{x},\mathbf{y})$.
6. On simule la loi de Bernoulli et suivant le résultat :

   on accepte la transformation : on tue alors u définitivement et on efface les cliques dont il est membre.

   on refuse la transformation : on tue u' et on efface les cliques dont u' est membre, on ressuscite u et on rend opaques les cliques dont u est membre.

Chapitre 6

Annexe B : Autres résultats

Détection de convergence

Pour détecter la convergence de la chaine de Markov produite, on ne peut se contenter de regarder l'évolution de l'espérance du nombre de points dans les configurations. On utilise également le nombre de cliques pour toutes les relations détectées.

Or parmi ces relations, les plus intéressantes sont celles qui n'interviennent pas directement dans le modèle.

Les exemples suivants expliquent ce propos.

Exemple : expérience 5

On lance une expérience selon le protocole présenté dans le paragraphe 3.3.5.

On prend donc pour β la valeur correspondant à un processus de Poisson dont l'espérance du nombre de points vaut 300 . On considère trois relations :

1. Une relation mère $R_{s1'}$ construite sur le même principe que R_{s1} avec un paramètre $d'_{\max}$ valant 1. Cela signifie que les cliques de $R_{s1'}$ contiennent les paires de points situés à une distance inférieure à 1.
2. La relation R_{s1}, dont le paramètre de distance vaut $d_{\max} = 0.6$.
3. La relation d'alignement R_3, dont le paramètre est fixé à 0.6, construite en utilisant la relation mère $R_{s1'}$. Cette relation, de même que dans le paragraphe 3.3.5, est munie d'un terme de qualité.

L'expérience 5 fait intervenir R_{s1} et R_3 dans le modèle. On fixe :

- pour R_{s1} : $\alpha = 1$ et $\delta = 0$ ce qui signifie que si l'on considère les disques de rayon 0.3 centrés en les points du processus, on ne veut pas que ces disques s'intersectent,
- pour R_3 : $\alpha = 1$ et $\delta = 1$ ce qui signifie que l'on veut que tous les points appartiennent à un seul alignement.

Ce modèle ne fait pas intervenir la relation $R_{s1'}$ directement. On se propose donc d'utiliser $p(R_{s1'})_t$ pour détecter la convergence. Les figures 6.1, 6.2, 6.3 et , 6.4 montrent les résultats obtenus.

- La figure 6.1 montre l'image obtenue. Le résultat n'est pas surprenant.
- La figure 6.2 montre l'évolution de la moyenne empirique de $n(\mathbf{x}_t)$.
- La figure 6.3 montre l'évolution de la moyenne empirique de $p(R_3,\mathbf{x}_t)$. Cette évolution est fortement liée à celle de $n(\mathbf{x}_t)$ du fait des contraintes imposées par le modèle en terme d'intensité et de diffusion.
- La figure 6.4 montre l'évolution de la moyenne empirique $p(R_{s1'}.\mathbf{x}_t)$. Cette évolution semble plus "libre" que celle de $p(R_3,\mathbf{x}_t)$.

Il paraît donc important d'utiliser $p(R_{s1'}, \mathbf{x}_t)$ pour détecter la convergence.

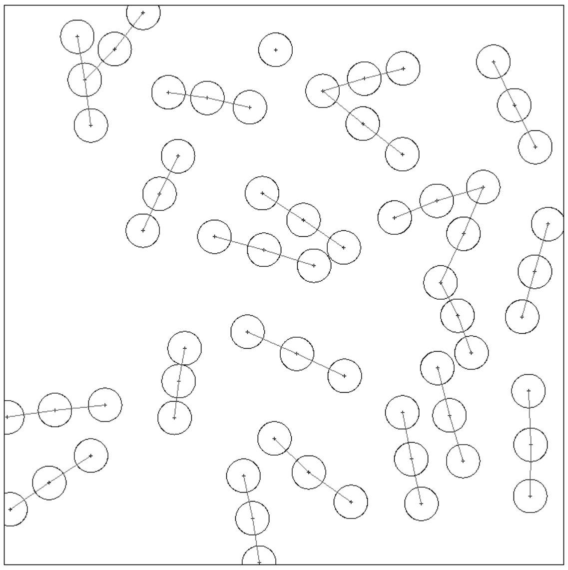

FIG. 6.1 – *Simulation 5 : Image de résultat.*

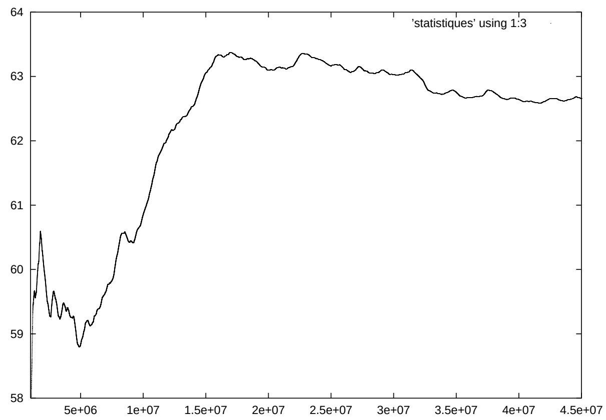

FIG. 6.2 – *Simulation 5 : Evolution de la moyenne empirique de* $n(\mathbf{x}_t)$.

Exemple 2 : Expérience 6

On ne fait intervenir dans le modèle que la relation R_{s1}. Les deux autres, c'est à dire R_3 et $R_{s1'}$ ne vont nous servir qu'à détecter la convergence de la chaine.

On fixe $\alpha_{R_{s1}} = 2$ et $\delta_{R_{s1}} = 1$.

On fixe β correspondant à une espérance du nombre de points valant 100.

Les figures 6.5, 6.6, 6.8, 6.7 et 6.9 montrent les résultats obtenus.

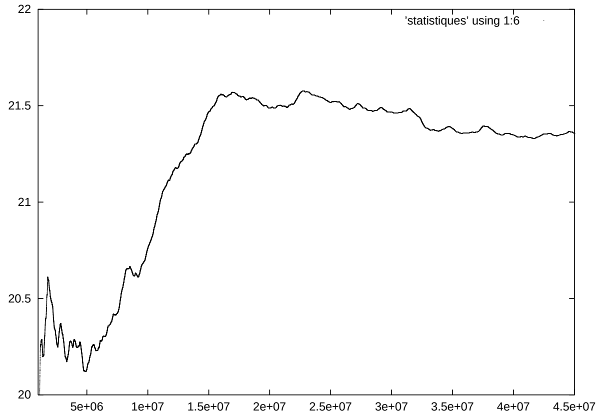

FIG. 6.3 – *Simulation 5 : Evolution de la moyenne empirique de* $p(R_3,\mathbf{x}_t)$.

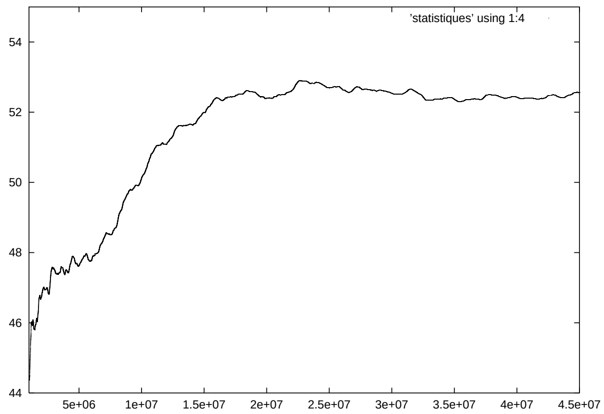

FIG. 6.4 – *Simulation 5 : Evolution de la moyenne empirique de* $p(R_{s1'},\mathbf{x}_t)$.

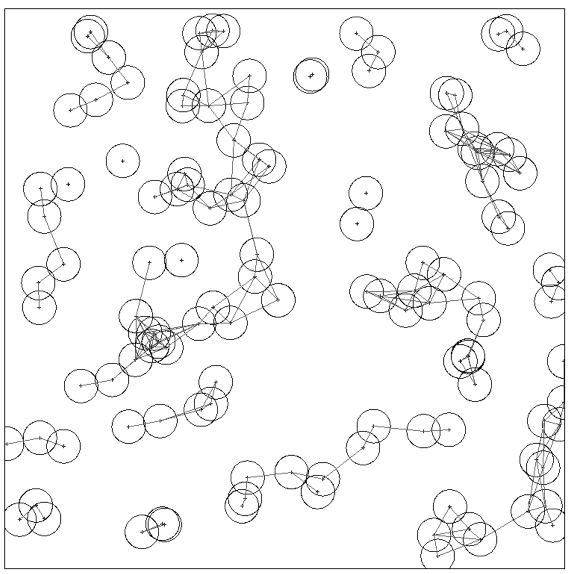

FIG. 6.5 – *Simulation 6 : Image de résultat.*

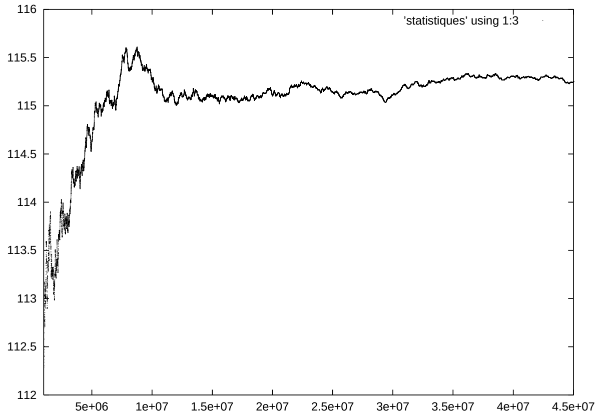

FIG. 6.6 – *Simulation 6 : Evolution de la moyenne empirique de* $n(\mathbf{x}_t)$.

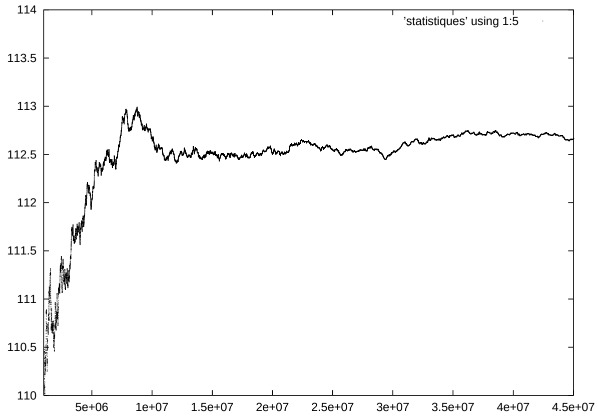

FIG. 6.7 – *Simulation 6 : Evolution de la moyenne empirique de* $p(R_{s1},\mathbf{x}_t)$.

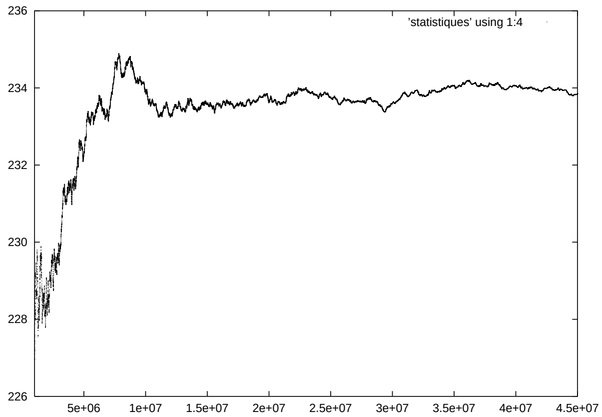

FIG. 6.8 – *Simulation 6 : Evolution de la moyenne empirique de* $p(R_{s1'},\mathbf{x}_t)$.

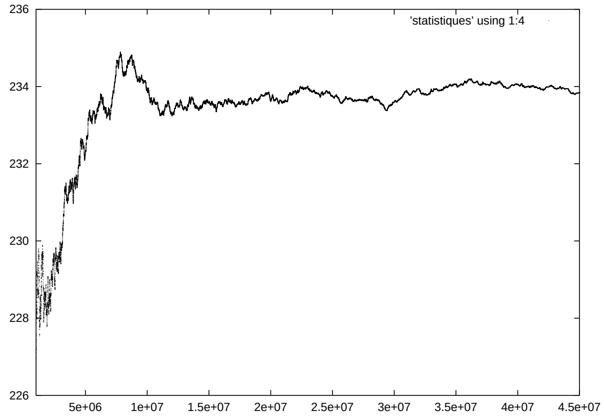

FIG. 6.9 – *Simulation 6 : Evolution de la moyenne empirique de* $p(R_3, \mathbf{x}_t)$.

- La figure 6.5 montre l'image obtenue. Là encore rien de surprenant. On note la presence des alignements sur cette image alors qu'ils n'interviennent pas dans la densité.
- La figure 6.6 montre l'évolution de la moyenne empirique de $n(\mathbf{x}_t)$.
- La figure 6.7 montre l'évolution de la moyenne empirique de $p(R_{s1}, \mathbf{x}_t)$. Cette évolution est liées à celle de $n(\mathbf{x}_t)$ du fait des contraintes imposées par le modèle sur l'intensité et la diffusion.
- La figure 6.8 montre l'évolution de la moyenne empirique de $p(R_{s1'}.\mathbf{x}_t)$.
- La figure 6.9 montre l'évolution de la moyenne empirique de $p(R_3.\mathbf{x}_t)$.

Ici les figures montrent que toutes ces grandeurs convergent. Celles qui nous intéressent vraiment sont les deux dernières grandeurs car elles ne sont pas contraintes par la densité du modèle.

Justification

Du point de vue théorique, on ne prouve pas du tout la convergence de la chaine. En fait on vérifie juste que l'on n'a pas de preuve de non convergence...

L'approche présentée dans les deux exemples précédents est justifiée par le théorème de Campbell-Mecke exposé dans le premier chapitre de ce rapport et le fait que les relations sont des fonctions sur les réalisations de processus ponctuels.