Point processes in forestry : an application to tree crown detection

Guillaume Perrin, Xavier Descombes, Josiane Zerubia

Guillaume Perrin, Xavier Descombes, Josiane Zerubia. Point processes in forestry : an application to tree crown detection. [Research Report] RR-5544, INRIA. 2006, pp.78. inria-00070463

HAL Id: inria-00070463
https://inria.hal.science/inria-00070463v1

Submitted on 19 May 2006

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Point processes in forestry : an application to tree crown detection

Guillaume Perrin — Xavier Descombes — Josiane Zerubia

N° 5544

April 2005

Thème COG

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--5544--FR+ENG

Point processes in forestry : an application to tree crown detection

Guillaume Perrin , Xavier Descombes , Josiane Zerubia

Thème COG — Systèmes cognitifs
Projet Ariana

Rapport de recherche n° 5544 — April 2005 — 78 pages

Abstract: In this research report, we aim at extracting tree crowns from remotely sensed images using marked point processes of discs and ellipses. Our approach is indeed to consider that the data are some realizations of a marked point process. Once a geometrical object is defined, we sample a marked point process defined by a density with a Reversible Jump Markov Chain Monte Carlo dynamics and simulated annealing to get the maximum a posteriori estimator of the tree crown distribution on the image.
In a first part, we propose to review the basis of marked point processes and some of their examples used in forestry statistic inference. Then, we present two new models, with discs and ellipses, and discuss some improvements made in the optimization or in the simulation.
Results are shown on high resolution aerial images of poplars provided by the French Forest Inventory (IFN), and synthetic images simulated with AMAP software (Bionatics, Digiplante project).

Key-words: Marked point process, RJMCMC, simulated annealing, automatic feature extraction, forests, tree crowns, aerial images.

Processus ponctuels en foresterie : une application à la détection de houppiers

Résumé : Dans ce rapport de recherche, notre but est d'extraire des houppiers à partir d'images aériennes de forêts à l'aide de processus ponctuels marqués de disques et d'ellipses. Notre approche consiste, en effet, à modéliser les données comme des réalisations de tels processus. Une fois l'objet géométrique de référence choisi, nous échantillonnons le processus objet défini par une densité grâce à un algorithme MCMC à sauts réversibles, optimisé par un recuit simulé afin d'extraire le maximum a posteriori de cette densité. Cette configuration optimale nous donnera l'extraction recherchée.

Dans une première partie, nous proposons de revenir quelque peu sur les processus ponctuels marqués et leur application dans la foresterie. Puis, nous présentons deux nouveaux modèles d'extraction de houppiers à base de disques et d'ellipses, et discutons de quelques améliorations au niveau de la simulation et de l'optimisation de notre algorithme.

Nous présentons des résultats obtenus sur des images aériennes très haute résolution fournies par l'Inventaire Forestier National (IFN), ainsi que sur des images synthétiques simulées avec le logiciel AMAP (Bionatics, projet Digiplante).

Mots-clés : Processus ponctuels marqués, RJMCMC, recuit simulé, extraction d'attributs, forêts, houppiers, images aériennes.

Contents

1 Introduction

Aerial and satellite imagery has a key role to play in forestry management. The increasing availability of data and their high spatial resolution, which is now submetric, allow developing automatic tools to analyze and monitor forests by evaluating accurately the vegetation resources. Some tools already exist for this purpose, using pixel-based methods or texture information and classification based on parameters such as covariance matrices (see [Gou98, LBN$^+$99, Lar99]). However, few works take advantage of the high data resolution. Nowadays, it is possible to study forests at the scale of trees. That is why we propose to use object processes for tree crown extraction.

1.1 Economical concerns

Digitized aerial photographs and satellite images of forests are convenient data to develop computerized assessments of forestry resources. First, automatic tools could help forest managers with species classification. This work is currently done thanks to a quite heavy analysis of these images by specialists, and some ground verifications. Then, we could also tackle the problem of the extraction of a single tree crown, and also assess some parameters such as the number of tree crowns, the distribution of their diameters or the stem density, which are of important economical and environmental concerns.

1.2 The proposed stochastic geometry approach

Basically, our approach consists in modelling the forestry images as realizations of a marked point process of trees. This stochastic framework aims at finding the best configuration of an unknown number of geometrical objects in the image, with respect to a probability density defined a priori. This density takes into account both the data, in order to fit the objects to the feature we want to extract, and the interactions between these objects, to favour or penalize some arrangements. In the case of tree crown extraction, we modeled the trees as ellipses, which are also defined by the position of their centre, their orientation, and their minor and major axes. After simulation, we obtain a collection of objects and have access to several statistics such as the number of trees in the stand, their position and their diameter.

In a first part of this report, we propose to review some key points of marked point processes and their application in forestry statistics inference. Then, we present two new models, with discs and ellipses. Some tests have been performed on digitized aerial photographs of poplars stands, courtesy of the French Forest Inventory (IFN), working on the infrared component which enhances the chlorophyll matter of vegetation. Other ones have been performed on synthetic images simulated with AMAP software, by courtesy of Digiplante project.

2 Marked point processes and their application in forestry

In this part we propose a review of the point process framework and its application to forestry. First we recall some definitions and give practical examples of point processes. Then, we exhibit a Markov Chain Monte Carlo sampler, the Reversible Jump MCMC algorithm, that we use in our model. People familiar with that part can directly jump to the third subsection which tackles the applications of point processes to forestry.

2.1 Definitions and first examples

2.1.1 Point process

Let X be a point process (cf [vL00] for more details) living in $\mathcal{P} = [0, X_M] \times [0, Y_M]$, a bounded set of $\mathbb{R}^2$. X is a measurable mapping from a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ to configurations of points of $\mathcal{P}$, in other words a random variable whose realizations are random configurations of points. These configurations, noted $\mathbf{x}$ in the following, belong to a space $(\mathbf{\Psi}, \mathcal{B}(\mathbf{\Psi}), \mu(.))$, where $\mathbf{\Psi}$ contains all configurations of a finite number of points of $\mathcal{P}$.

2.1.2 Marked point process

A marked point process living in $S = \mathcal{P} \times \mathcal{K}$, a bounded set of $\mathbb{R}^d$, is a point process where some marks are added to the positions of points. Note that we do not deal with points anymore, but with objects : a configuration of objects is a finite set of marked points. $\mathcal{K}$ is the space of the marks defining the objects geometry, while $\mathcal{P}$ is the space of the object position. A marked point process X is still a measurable mapping from a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ to configurations of points of S.

The marks are some parameters that fully describe the object, for example on a geometric point. Circles are described by the position of their center and their radius, ellipses by the position of their center, their major and minor axis, and their orientation (see Fig. (1)).

2.1.3 Poisson point process

Let $\nu(.)$ be a measure on S. X is a Poisson point process with intensity measure $\nu(.)$ if for each Borel set $B \in S$, the random variable $N(B)$ that counts the number of points of X falling in B follows a Poisson law with mean $\nu(B)$, and if for any k

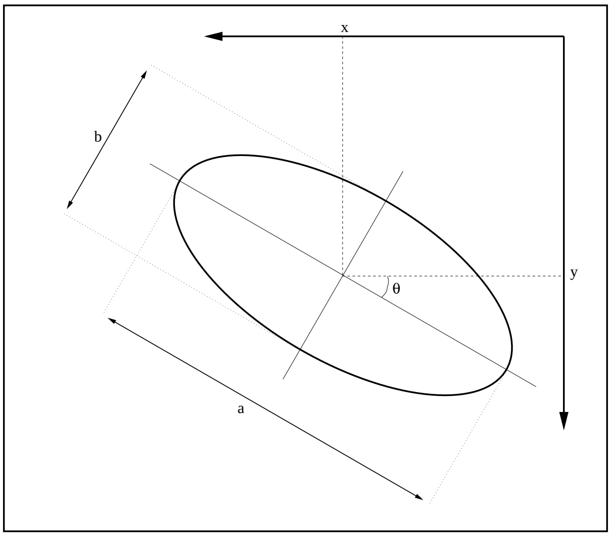

Figure 1: Position and marks of an ellipse.

disjoint Borel sets $B_k \in S$, $N(B_k)$ are independent random variables.

To sample a configuration $\mathbf{x}$, ie. a realization of a Poisson point process, a two-step algorithm is performed :

1. generate $n = N(S)$, the number of points of the configuration $\mathbf{x}$ in S, with respect to the discrete Poisson law of parameter $\nu(S)$.
2. generate n points $\nu(.)$-distributed on S independently.

If $\nu(.)$ is proportional to the Lebesgue measure $\lambda_S(.)$ of S, the Poisson process is called homogeneous, or stationary. This induces a complete spatial randomness, given the fact that the positions of the points are uniformly and independently distributed. Fig. (2) shows two realizations of such an homogeneous Poisson process.

Otherwise, the process is an inhomogeneous or general Poisson process. If $\nu(.)$ has a density $l(.)$ with respect to the Lebesgue measure $\lambda_S(.)$, this density is called the intensity function. Fig. (3) gives us two realizations of such an inhomogeneous Poisson process.

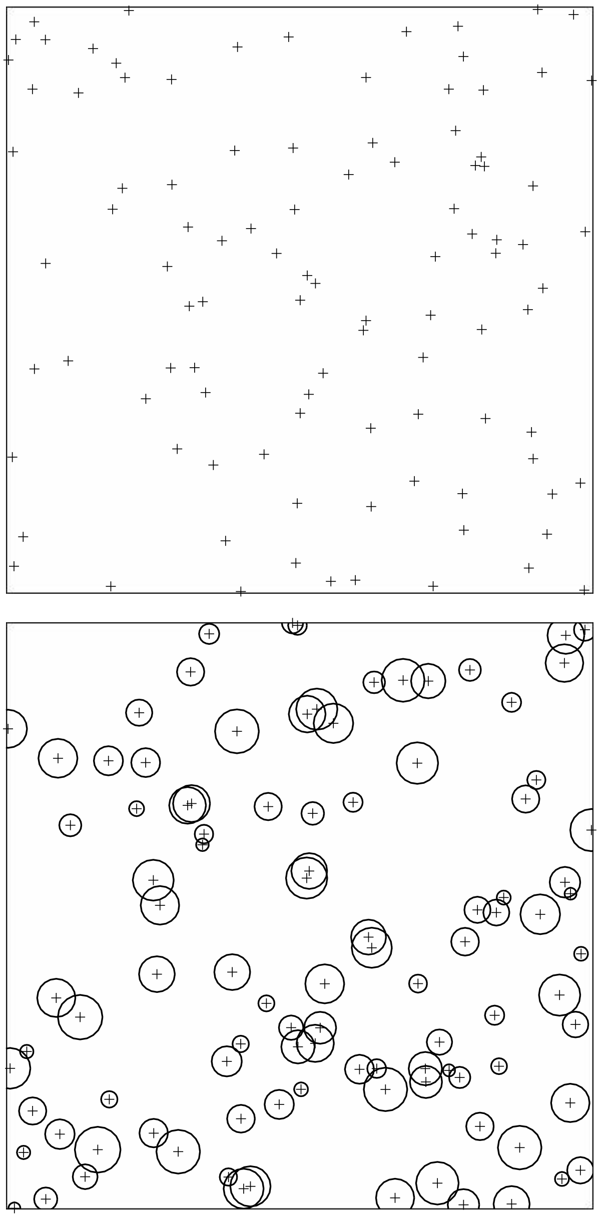

Figure 2: Realizations of two homogeneous Poisson processes of mean 100. Top : a point process. Bottom : a marked point process of discs.

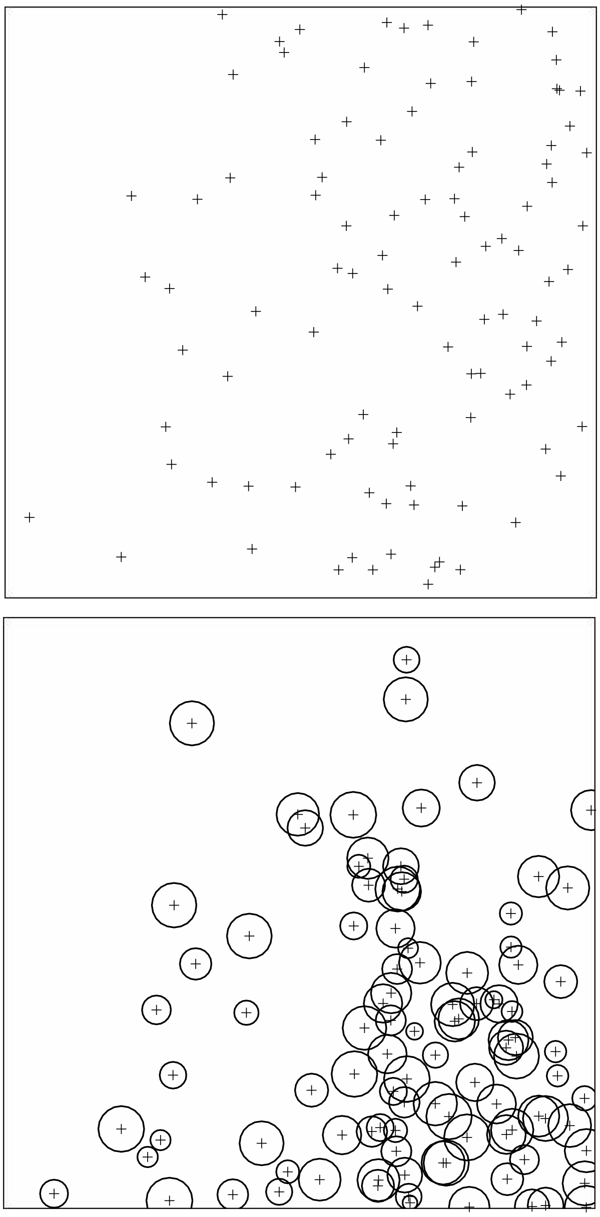

Figure 3: Realizations of two inhomogeneous Poisson processes of mean 100. Top : a point process with an intensity function $l\,(x = \{x_1, x_2\}) \propto x_1$ which favours the points at the right handside. Bottom : a marked point process of discs with an intensity function $l\,(x = \{x_1, x_2\}) \propto x_1.x_2.r$ which favours the discs at the bottom right with a large radius.

2.1.4 Density and reference measure of a marked point process

We now consider the probability distribution $\mu(.)$ of a Poisson point process on S. This gives us a probability measure on $\mathbf{\Psi}$.

Then, if the probability distribution of a marked point process X , written $\mathcal{P}_X(.)$, is uniformly continuous with respect to $\mu(.)$, we can define its unnormalized density $f(.)$ with respect to this dominating reference measure as :

$$\mathcal{P}_X(d\mathbf{x}) = \frac{1}{Z} f(\mathbf{x})\mu(d\mathbf{x}) \tag{1}$$

where Z is the normalizing constant :

$$Z = \int_{\mathbf{\Psi}} f(\mathbf{x}) d\mu(\mathbf{x})$$

2.1.5 Strauss point process

This leads to several examples of point processes in the literature. A broad family of point processes is the exponential family model, where the density $f(.)$ has the form :

$$f(\mathbf{x}) = e^{<\theta, t(\mathbf{x})>} \tag{2}$$

$< .,. >$ is a scalar product, θ a vector of a parameter space Θ (typically $\Theta \subset \mathbb{R}^n$), and $t(.)$ a map from $\mathbf{\Psi}$ to $\mathbb{R}^n$.

The Strauss process [Str75] belongs to this family. Let $n(\mathbf{x})$ be the number of points in a configuration $\mathbf{x}$, and $s(\mathbf{x})$ the number of pairs of points in $\mathbf{x}$ that are separated by a distance smaller than a fixed real number r, the density of the Strauss point process is :

$$f(\mathbf{x}) = e^{\theta_1 n(\mathbf{x}) + \theta_2 s(\mathbf{x})}$$

The natural parameter space Θ for this point process to be integrable (ie. for Z to be finite) is the open subset :

$$\Theta = \left\{ \theta \in \mathbb{R}^2 : \theta_2 \leq 0 \right\}$$

which induces a repulsion between points of the process. Given this parameter space, the process is a non-regular [GM98] family which affects likelihood inference for the model.

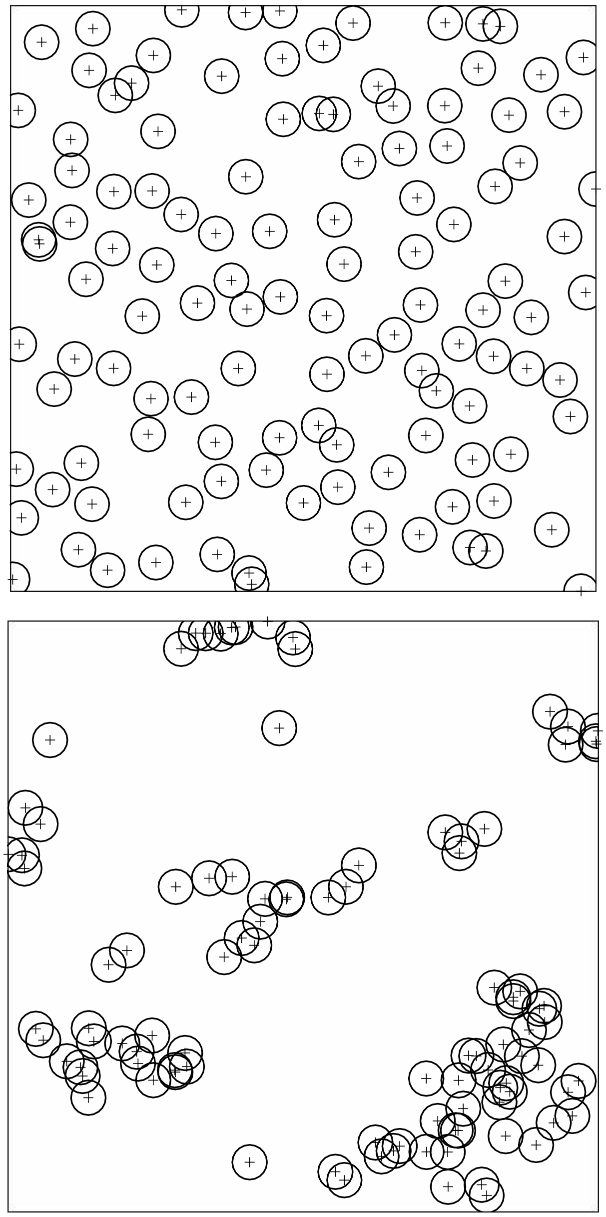

Figure 4: Realizations of two Strauss processes. Top : unconditional Strauss Process with $\theta_2 < 0$ which penalizes clustering. Bottom : conditional point process with $n(\mathbf{x}) = 100$ and $\theta_2 > 0$ which favours clustering.

This can be resolved by adding a condition on the process, ie. by fixing the total number of points : $n(\mathbf{x}) = k$. The new process is a conditional Strauss point process, with only one parameter : the parameter θ_2, which can now be positive and also favour the clustering of points.

Fig. (4) gives two realizations of Strauss point processes, the second one being the conditional process.

2.1.6 Energy formulation of the density

An interesting formulation of the probability density of a configuration $\mathbf{x}$ is given by its Gibbs energy $U(\mathbf{x})$:

$$U(\mathbf{x}) = -\log\left(f(\mathbf{x})\right) \tag{3}$$

For exponential family models, $U(\mathbf{x})$ equals to the negative scalar product detailed in (2).

This energy formulation will be useful for the discussions that will be proposed later in this report. They will be based on this formulation, for example by speaking about a configuration which minimizes the energy of the model, we will refer to the maximum of the density.

2.2 Simulation and Optimization

Once the probability distribution $\mathcal{P}_X(.)$ of a marked point process has been defined (1), the issue is to generate a realization of it. Obtaining a configuration $\mathbf{x}$ which is $\mathcal{P}_X(.)$ -distributed can be done with Markov Chain Monte Carlo algorithms.

2.2.1 Stability of a point process

A point process is said to be locally stable if its Papangelou conditional intensity satisfies :

$$\lambda(\mathbf{x}, u) = \frac{f\left(\mathbf{x} \cup \{u\}\right)}{f(\mathbf{x})} \leq M \tag{4}$$

with $M > 0$ finite, for all $\mathbf{x} \in \mathbf{\Psi}$ and $u \in S$.

When (4) holds, Ruelle's condition holds and also the integrability of the density with respect to a reference Poisson measure is ensured. Moreover, his condition is required to prove the convergence properties of Monte Carlo schemes [GM98].

2.2.2 Markov Chain Monte Carlo samplers

Basically, MCMC samplers consist in building a Markov chain (X_t) on $(\mathbf{\Psi}, \mathcal{B}(\mathbf{\Psi}), \mu(.))$, which converges ergodically to $\mathcal{P}_X(.)$. As we do not know a priori the total number of objects of the final configuration, we will have to jump between the dimensions of the configuration space, that is why the Reversible Jump MCMC algorithm of Geyer and Møller [GM94] is well adapted for our application.

This algorithm, based on Metropolis Hastings samplers, has been initially proposed in [GM94], then reviewed and generalized by Green in [Gre95]. Its other name is Metropolis Hastings Green algorithm (see [GM94] or [GM98] for more details).

2.2.3 Reversible Jump Markov Chain Monte Carlo algorithm

In [Gre95], Green proposes an algorithm that builds a Markov chain (X_t) which converges ergodically to $\mathcal{P}_X(.)$. This algorithm can be summarized in 3 steps, and the convergence relies on some conditions.

Algorithm ***RJMCMC***

For a given state $X_t = \mathbf{x}$ *:*

1. Let $Q(\mathbf{x},.) = \sum_m Q_m(\mathbf{x},.)$ *be the sum of several proposition kernels. Choose the kernel* m *with probability* $p_m(\mathbf{x}) = Q_m(\mathbf{x}, \mathbf{\Psi})$*, and then generate* $\mathbf{y} \sim \frac{Q_m(\mathbf{x},.)}{p_m(\mathbf{x})}$.

2. Compute Green's ratio :

$$R_m(\mathbf{x}, \mathbf{y}) = \frac{f_m(\mathbf{y}, \mathbf{x})}{f_m(\mathbf{x}, \mathbf{y})}$$

where $f_m(\mathbf{x}, \mathbf{y})$ *is detailed hereafter.*

3. With probability $\alpha_m = \min(1, R)$*, accept the move and set* $X_{t+1} = \mathbf{y}$*. Otherwise, stay at* $X_{t+1} = \mathbf{x}$.

Green's ratio is computed thanks to the introduction of a function $f_m(.,.)$ which itself relies on the existence of a symmetric measure $\xi_m(d\mathbf{x}, d\mathbf{y})$ on $\mathcal{B}(\mathbf{\Psi}) \times \mathcal{B}(\mathbf{\Psi})$ which dominates $\mathcal{P}_X(d\mathbf{x})Q_m(\mathbf{x}, d\mathbf{y})$. If for each kernel, we find such a symmetric measure $\xi_m(.,.)$, then $f_m(.,.)$ is the Radon Nikodym derivative :

$$f_m(\mathbf{x}, \mathbf{y}) = \frac{\mathcal{P}_X(d\mathbf{x})Q_m(\mathbf{x}, d\mathbf{y})}{\xi_m(d\mathbf{x}, d\mathbf{y})} \tag{5}$$

The choice of the acceptance ratio ensures the $\mathcal{P}_X(.)$ -reversibility of the Markov chain X_t. The other properties, such as the irreducibility, the Harris recurrence and the geometric ergodicity, are obtained if the stability condition (4) holds. This was proved by Geyer and Møller [GM98] in the case of a birth and death kernel, but needs to be proved again for other kernels.

As soon as we design new proposition kernels, we will have to prove the ergodic convergence of the Markov chain to $\mathcal{P}_X(.)$.

2.2.4 Bayesian modelling

Under some conditions, it is possible to write the density $f(.)$ using Bayes rule :

$$f(\mathbf{x}) = f(\mathbf{x}|\mathcal{I}) = \frac{f_p(\mathbf{x})\mathcal{L}(\mathcal{I}|\mathbf{x})}{f(\mathcal{I})} \propto f_p(\mathbf{x})\mathcal{L}(\mathcal{I}|\mathbf{x}). \tag{6}$$

Basically, it requires first to define a likelihood $\mathcal{L}(\mathcal{I}|\mathbf{x})$, which represents the probability for the data $\mathcal{I}$ to be this one, knowing the configuration $\mathbf{x}$, and then a prior density $f_p(\mathbf{x})$. All the prior knowledge we have on the kind of configurations we are looking for will be considered in the definition of this density.

In our application, this requires to be able to give a statistical model for the image. We also have to be able to know what the pixel distribution is inside and outside these objects, which is not so obvious in remote sensing images that can contain crops, buildings, lakes ...

The object detection consists then in finding the configuration of objects that maximizes this density, or minimizes the associated energy. To obtain this configuration, we use the Maximum A Posteriori estimator (MAP).

2.2.5 Simulated annealing

In order to find the MAP of the density $f(.)$, a simulated annealing scheme is possible. Indeed, starting from any configuration of objects $\mathbf{x}_0$, we would like to get one of the global maxima of $f(.)$.

The solution is to sample $f^{\frac{1}{T_t}}(.)$ instead of $f(.)$. T_t is a temperature parameter which tends to zero as t tends to ∞. A logarithmic decrease ensures the convergence to a sum of Diracs located in the global maxima of $f(.)$. In practice, as it is not possible to use a logarithmic decrease, we use a geometric one. Thus it becomes very important to enhance the mixing of the Markov chain with well-chosen proposition kernels, in order to avoid local maxima of $f(.)$.

2.3 Point processes in forestry statistics

Point processes have been widely used in forestry statistics problems, since this is the field of application in which spatial statistics are the most developed [PS00]. Tree crown extraction was performed using marked point processes on LIDAR images in

[And03], but this is the only application of marked point processes for extraction in the literature. Most of the time, point processes are used once this extraction has been done : other automatic methods or manual methods collect some data on the tree characteristics (position, tree height and diameter, ...), and people try to model them by means of marked point processes. The most attractive point process models in that prospect are Cox and Gibbs processes.

The Cox process [SKM95], also known as the doubly stochastic Poisson process, can be seen as a two-step random mechanism. First we choose a distribution on the space of non-negative locally finite measures on S, of density $l(.)$ with respect to the Lebesgue measure. Then, we sample the inhomogeneous Poisson process with intensity $l(.)$. This is also a generalization of Poisson process, made by supposing that the intensity measure is itself random. One example is the Matern's cluster process, where the points x_i of the configuration $\mathbf{x}$ form some clusters centered at other points u_j, obtained in a first step with an homogeneous Poisson process. The random intensity function is also piecewise constant arount these base points (see Fig. (5)).

The Gibbs process offers more flexibility in the density, because it consists in weighting the Poisson distribution with a density function $f(.)$, generally defined as a sum of interactions between the objects of the configuration. When the number of objects of the configuration is not fixed, they are defined with respect to a reference Poisson process. For example, the density of Gibbs process with pairwise interactions, to which the Strauss process belongs, is :

$$f(\mathbf{x}) = \frac{1}{Z} \exp\left(- \sum_{i<j} \phi(x_i, x_j) \right)$$

The model that we designed belongs to the family of Gibbs processes.

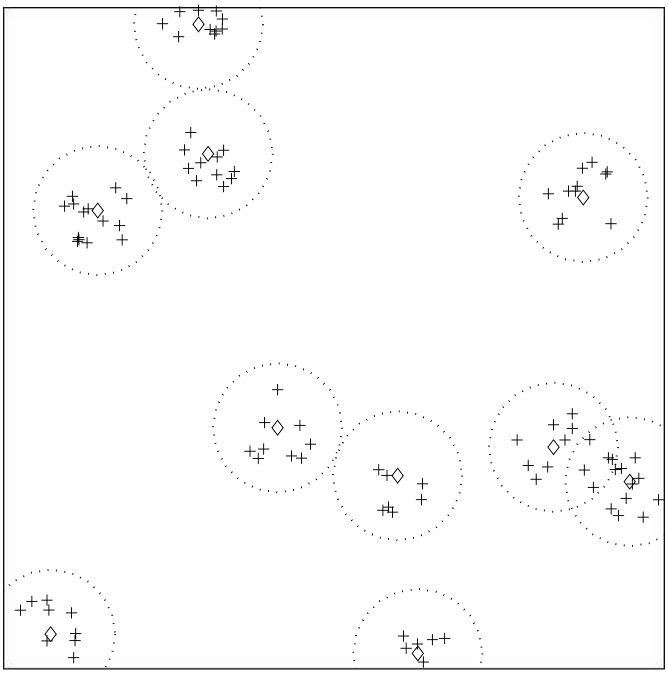

Figure 5: Matern point process : the base points are rhombus, and the dotted circle the area where the intensity is strictly positive.

3 Disc model

We use marked point processes to extract tree crowns from digital images. As detailed above, we have first to define the state space S, then to define a reference measure $\mu(.)$ and a density $f(.)$ for our marked point process. Let $\mathcal{I}$ be the image, ie. the observed data, and $\mathbf{x}$ a configuration of objects of S.

3.1 Description of the disc model

3.1.1 State space

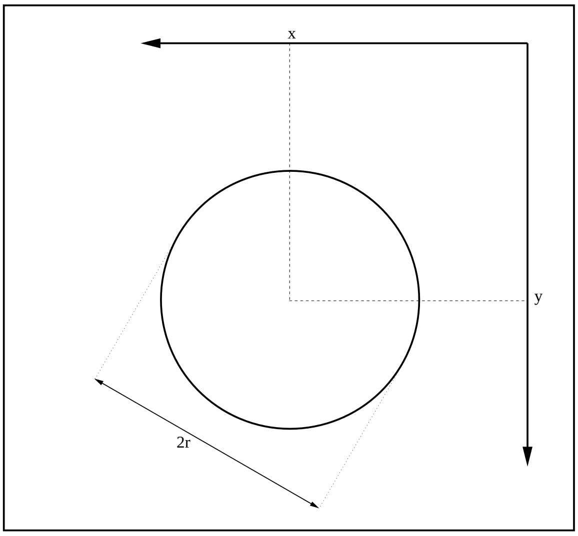

Figure 6: Position and marks of a disc.

This first model consists of a marked point process of discs. Thus, the state space S is a bounded set of $\mathbb{R}^3$:

$$S = \mathcal{P} \times \mathcal{K} = [0, X_M] \times [0, Y_M] \times [R_m, R_M] \tag{7}$$

where X_M and Y_M are respectively the height and the length of the image $\mathcal{I}$, and (R_m, R_M) the minimum and the maximum radius of our objects. We will see that finding good values for these two radii is essential in order to avoid sub-detection or over-detection of trees in the image, and we will see how we can estimate, or more precisely calibrate them.

In the following, an object $x \in S$ will be also noted $x = (p, r)$, where $p \in \mathcal{P}$ is the position of its center and $r \in \mathcal{K}$ its radius (see Fig. (6)).

3.1.2 Prior term

▷ **D**iscussion

Prior term gives a general aspect of the solution we desire. According to the geometric properties of the tree stands, Cox process or Gibbs process can be chosen. Working on plantations of poplars, it's obvious that Gibbs processes are recommended because they enable us to model the periodic pattern of the alignments.

Thus, the two basic rules we introduced in the prior term are :

- a repulsive term between two overlapping objects (see [PDZ03]) in order to avoid over-detection. Indeed, a single tree must be extracted by one and only one marked point of the configuration.
- an attractive term that favours regular alignments in the configuration. These alignments are defined by two vectors, which can be calculated using the Fourier transform of the image.

In the following, we will use the energy formulation presented in (3), saying that the prior density $f_p(\mathbf{x})$ can be defined by means of a prior energy $U_p(\mathbf{x})$:

$$U_p(\mathbf{x}) = -\log\left(f_p(\mathbf{x})\right) \tag{8}$$

▷ **R**epulsive interaction

First, we have to define a symmetric neighbourhood relation between objects of S to define the overlapping rule. Then, we will discuss the penalization we associate to each pair of overlapping objects.

Definition

Overlapping objects

Two objects $x_1 = (p_1, r_1)$ *and* $x_2 = (p_2, r_2)$ *are overlapping, and we write* $x_1 \sim_r x_2$*, if* $d(p_1, p_2) \leq r_1 + r_2$*. This means that the intersection of their silhouette is not empty on* $\mathcal{P}$*.*

Then, as we would like the pairs of overlapping objects to be penalized, we have to add a positive weight $U_r(\mathbf{x})$ to the prior energy $U_p(\mathbf{x})$ in (8) :

$$U_r(\mathbf{x}) = \sum_{x_1 \sim_r x_2} U_r(x_1, x_2) \geq 0$$

Moreover, it would be interesting to penalize more or less these cliques depending on the way their objects overlap (see Fig. (7)). Indeed, in dense forest areas, it will often occur that two discs overlap, because trees are very close from each other. That is why we define an overlapping coefficient.

Definition

Overlapping coefficient

Let $x_1 = (p_1, r_1)$ *and* $x_2 = (p_2, r_2)$ *be two overlapping objects* ($x_1 \sim_r x_2$)*. The overlapping coefficient,* $\mathcal{A}(x_1, x_2) \in [0, 1]$*, is defined as :*

$$\mathcal{A}(x_1, x_2) = \frac{\lambda_{\mathcal{P}}(x_1 \cap x_2)}{\max\left(\pi r_1^2, \pi r_2^2\right)}$$

where $\lambda_{\mathcal{P}}(x_1 \cap x_2)$ *is the Lebesgue measure on* $\mathcal{P}$ *of the intersection of the silhouettes.*

Eventually, we take a parameter $\gamma_r \geq 0$ that ponders the positive contribution to the energy, which is written as follows :

$$U_r(\mathbf{x}) = \gamma_r \sum_{x_i \sim_r x_j} \mathcal{A}(x_i, x_j) \tag{9}$$

▷ **A**ttractive interaction

Tree plantations look like grids, with two specific orientations and regular inter-objects distances. It's also important to enhance this property in the prior model,

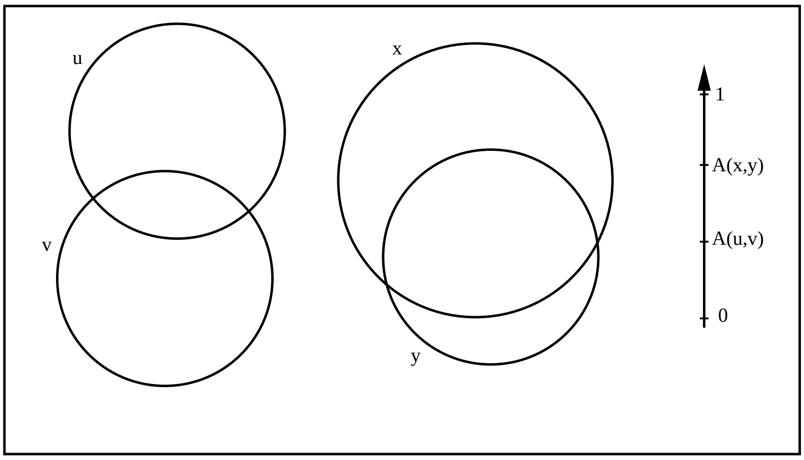

Figure 7: Different levels of repulsion between overlapping objects.

by adding an attractive interaction between two objects presenting such a property. Suppose we have the two vectors that characterize these alignments, denoted $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$ (see Fig. (8)), we can now define a symmetric neighbourhood relation on S.

Definition

Aligned objects

Two objects $x_1 = (p_1, r_1)$ *and* $x_2 = (p_2, r_2)$ *are aligned, and we write* $x_1 \sim_a x_2$*, if* $p_2 \in \bigcup_{i=\{1,2\}} \mathcal{B}_\epsilon(p_1 \pm \overrightarrow{v_i})$*, where* $\mathcal{B}_\epsilon(A)$ *is the closed ball of center* $A \in \mathcal{P}$ *and of radius* ϵ *(*$\epsilon = 3$ *in our model). Note that this relation is symmetric.*

Then, as we would like the pairs of aligned objects to be favoured, we add a negative weight for each of them, pondered by a parameter $\gamma_a \leq 0$ and a quality function $\mathcal{Q}(x_1, x_2)$ that quantifies the quality of the alignment :

$$\mathcal{Q}(x_1, x_2) = \sigma\left(\min_{i=\{1,2\}} \left(\|\overrightarrow{p_1 p_2} \pm \overrightarrow{v_i}\|\right), \epsilon\right) \in [0, 1]$$

where $\sigma(x, d)$ is the reward function proposed in [ODZ03a] :

$$\sigma(x, d) = \frac{1}{d^2}\left(\frac{1 + d^2}{1 + x^2} - 1\right)$$

The total contribution of the pairs of attracted objects is written as follows :

$$U_a(\mathbf{x}) = \gamma_a \sum_{x_i \sim_a x_j} \mathcal{Q}(x_i, x_j) \tag{10}$$

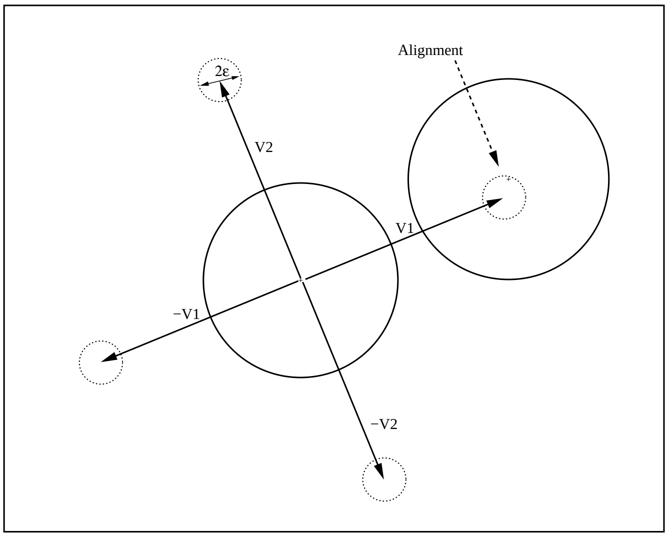

Figure 8: Attractive interaction between two aligned objects.

▷ **P**arameter space and stability of the prior model

The global energy of the prior model is given, using (9) and (10), by :

$$U_p(\mathbf{x}) = \gamma_r \sum_{x_i \sim_r x_j} \mathcal{A}(x_i, x_j) + \gamma_a \sum_{x_i \sim_a x_j} \mathcal{Q}(x_i, x_j) \ , \ (\gamma_r, \gamma_a) \in \mathbb{R}^+ \times \mathbb{R}^- \tag{11}$$

The question of stability is essential at this point in order to sample this process. So, as defined in (11), is our prior model stable with respect to (4) ? The answer is no. The logarithm of the Papangelou derivative is not majored by any constant $M > 0$

because of the attractive term in the density which can diverge :

$$\begin{aligned}\log\left(\lambda_p(\mathbf{x},u)\right) &= -\,\Delta\left(U_p(\mathbf{x})\right)\big|_{\mathbf{x}}^{\mathbf{x}\cup\{u\}}\\ &= -\gamma_r\sum_{x_i\sim_r u}\mathcal{A}(x_i,u)-\gamma_a\sum_{x_i\sim_a u}\mathcal{Q}(x_i,u)\\ &\leq -\gamma_a\sum_{x_i\sim_a u}\mathcal{Q}(x_i,u)=G\end{aligned}$$

Indeed, we could think of a sequence of configurations $\mathbf{x}_n$ for which the associated $G_n \to +\infty$. For example, $\mathbf{x}_n$ could be composed of exactly n objects, all located in a ball of radius ϵ and center $x \in \mathcal{P}$. Adding a marked point $u = (p_u, r_u)$ such that $\overrightarrow{xp_u} = \overrightarrow{v_i}$ would create n new pairs of $\sim_a$-neighbours. It is also not possible to find a constant $M > 0$ that bounds the G_n for all $\mathbf{x}_n$.

As a conclusion, for stability reasons, we have to avoid extreme closeness of objects. This can be done by adding an hard core term in our prior process. Thus, an infinite energy is given to pairs of too-close objects (see [vL00]) :

$$U_h(\mathbf{x}) = \begin{cases} +\infty & \text{if } \exists (x_i, x_j) \in \mathbf{x} \mid d(x_i, x_j) < \eta \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where η is set to one. And the global prior energy becomes :

$$U_p(\mathbf{x}) = U_r(\mathbf{x}) + U_a(\mathbf{x}) + U_h(\mathbf{x}) \tag{13}$$

Thus, the minimum distance between two objects of the configuration is 1, which corresponds to one pixel on the image. It is significative in the sense that we do not expect to detect two trees with an inter-distance on the ground less than the resolution of the image. Note that due to the hard-core energy, the stability is now ensured, because the number of objects of any configuration with a positive prior density $f_p(.)$ is bounded :

$$\lambda_p(\mathbf{x}, u) = \frac{f_p(\mathbf{x} \cup \{u\})}{f_p(\mathbf{x})} \leq M_p \tag{14}$$

We could now work on a smaller space of the configurations, considering that the configurations which present some hard core repulsion will have a null density, and will be of no interest for us in our optimization. From now on, $\boldsymbol{\Psi}$ will also only contain the configurations that have a finite energy $U_p(\mathbf{x})$.

3.1.3 Likelihood

The likelihood of the data $\mathcal{I}$ given a configuration $\mathbf{x}$, written $\mathcal{L}(\mathcal{I}|\mathbf{x})$, can be defined if our images are predictible. This means that we are supposed to predict the data from the configuration of extracted objects $\mathbf{x}$.

In some images like in Fig. (9), this is possible to define such a likelihood using a Gaussian mixture (see [PDZ03]). Each pixel of the image is associated to one of the two Gaussian classes :

- $\mathcal{C}_i = \mathcal{N}(\mu_i, \sigma_i)$ for the pixels inside at least one of the objects of the configuration,
- $\mathcal{C}_o = \mathcal{N}(\mu_o, \sigma_o)$ for the pixels outside.

But for images like in Fig. (10), this kind of likelihood is not appropriate anymore. We will discuss later this problem.

3.1.4 Conclusion : stability and probablity measure

As a conclusion, the density of our disc marked point process can be written as :

$$f(\mathbf{x}) = f_p(\mathbf{x}).\mathcal{L}(\mathcal{I}|\mathbf{x})$$

The stability condition (4) is obtained because the prior term $f_p(\mathbf{x})$ is stable (14) and the likelihood is bounded : $m_{\mathcal{L}} \leq \mathcal{L}(\mathcal{I}|\mathbf{x}) \leq M_{\mathcal{L}}$ (see [PDZ03]). Thus, we can write the following inequality :

$$\lambda(\mathbf{x}, u) = \frac{f(\mathbf{x} \cup \{u\})}{f(\mathbf{x})} \leq M = \frac{M_p m_{\mathcal{L}}}{M_{\mathcal{L}}} \tag{15}$$

Moreover, as we define this marked point process with respect to an homogeneous Poisson process, the probability distribution $\mathcal{P}_X(.)$ is given by :

$$\mathcal{P}_X(d\mathbf{x}) = \frac{1}{Z} f(\mathbf{x}) \mu(d\mathbf{x})$$

where Z is the normalizing constant, and $\mu(d\mathbf{x})$ the distribution of the reference Poisson process of intensity measure $\nu(.)$.

In order to control the expectation of the number of points of the reference process, we weight this measure with a parameter β, stating that the intensity measure $\nu(.) = \beta\lambda_S(.)$, where $\lambda_S(.)$ is the Lebesgue measure on S. We will see in the section about the simulations of the prior model how this parameter β appears.

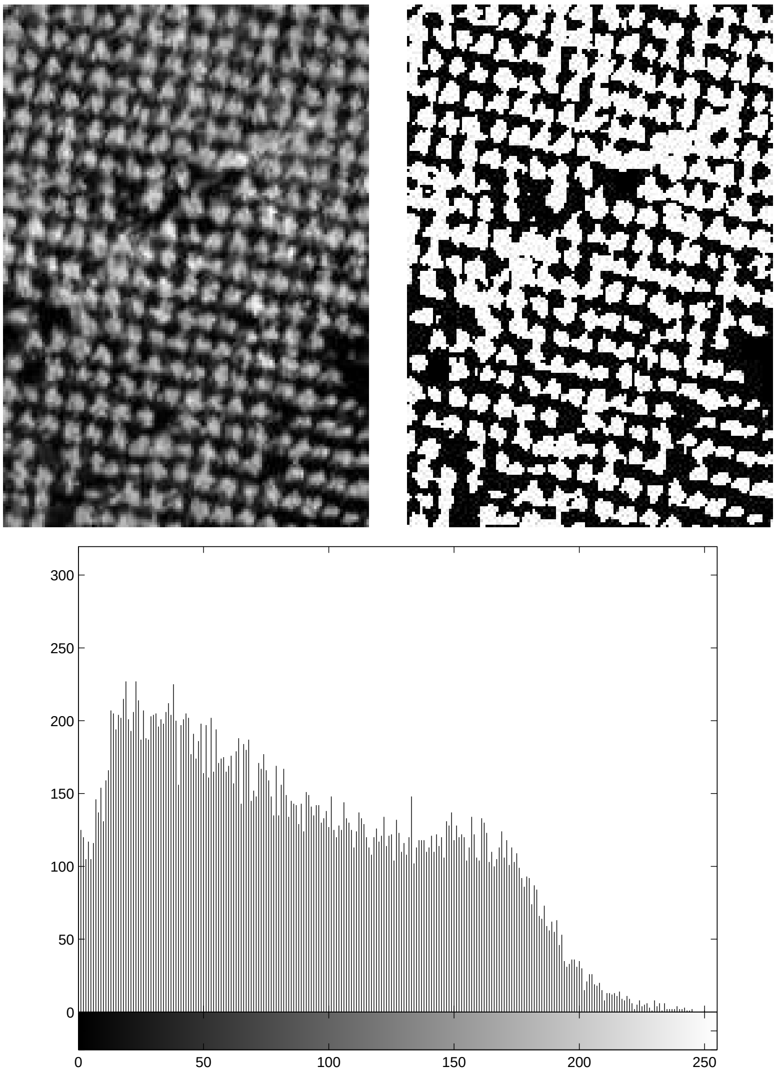

Figure 9: Top left : an image of a forest. Top right : its histogram. Bottom : the KMeans binary image associated to it.

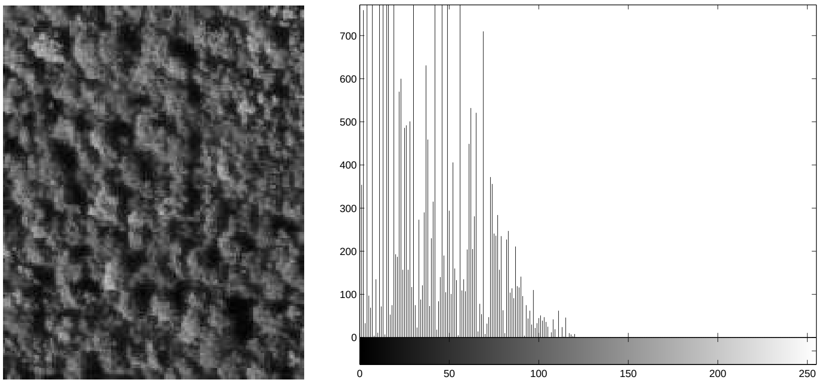

Figure 10: An image where a mixture of two Gaussians is a poor model

3.2 MCMC sampler

Here we present the different sub-kernels of our MCMC sampler. We develop neither the birth and death nor the non-jumping transformations anymore (see [PDZ03]), but we expose the new split and merge and the birth and death in a neighbourhood. All the density $f(.)$ that compose the different Green's ratios are obviously replaced by $f^{\frac{1}{T_t}}(.)$ in a simulated annealing scheme.

Note that the moves detailed hereafter do not consider the reduced workspace of configurations, which only contains finite energy configurations. In order to stay in this space $\mathbf{\Psi}$ during the simulation, we propose to add one rule : from a state $X_t = \mathbf{x} \in \mathbf{\Psi}$, if the proposed configuration $X_{t+1} = \mathbf{y}$ presents some hard core repulsion ($\mathbf{y} \notin \mathbf{\Psi}$), we stay at $\mathbf{x}$.

3.2.1 Birth and Death and non jumping transformations

▷ Birth and Death of an object

The birth or death kernel Q_{BD} consists in proposing with a probability p_B to uniformly add in S an object to the current configuration $\mathbf{x}$, or with probability p_D to remove at random one object of the configuration (see Fig. (11)). Here we only recall the Green's ratios, from which are infered the acceptance ratios. All details about them can be found in [PDZ03]. For the birth of an object $u \in S$ in the configuration

$\mathbf{x}$, the ratio is :

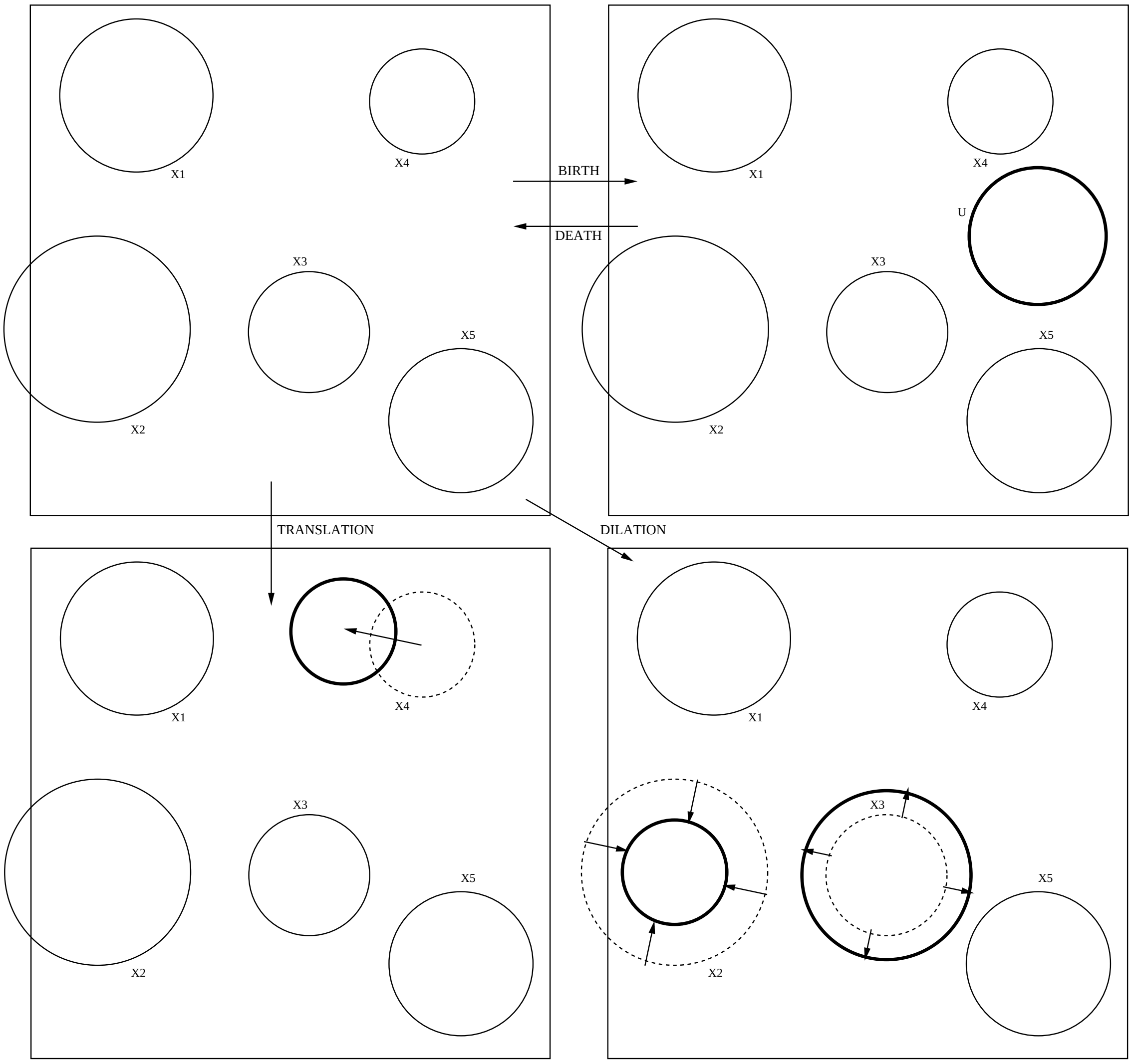

Figure 11: Birth and Death and non jumping kernels.

$$R_{BD}(\mathbf{x}, \mathbf{y} = \mathbf{x} \cup \{u\}) = \frac{p_D}{p_B} \frac{f(\mathbf{y})}{f(\mathbf{x})} \frac{\nu(S)}{n(\mathbf{x}) + 1} \tag{16}$$

and for a death :

$$R_{BD}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{u\}) = \frac{p_B}{p_D} \frac{f(\mathbf{y})}{f(\mathbf{x})} \frac{n(\mathbf{x})}{\nu(S)} \tag{17}$$

where $n(\mathbf{x})$ is the number of objects of the configuration $\mathbf{x}$.

▷ **N**on jumping transformations

The non-jumping kernels consist in proposing to a uniformly-chosen object of the configuration $\mathbf{x}$ a translation, a dilation, or both of them (see Fig. (11)). The Green's ratio of all these sub-kernels is :

$$R_{NJ}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{u\} \cup \{v\}) = \frac{f(\mathbf{y})}{f(\mathbf{x})} \tag{18}$$

where $u \in \mathbf{x}$ has been transformed in $v \in S$ by the kernel.

3.2.2 Split and Merge revisited

The split and merge kernel has been developed in order to avoid sub-detection or over-detection of trees. Indeed, without this kernel we could be stuck in a local minimum of the energy $U(\mathbf{x})$ in the simulated annealing, because there is no simple move (non-jumping transformation or birth or death) which enables the Markov chain to go from one "bad" state $\mathbf{x}$ to a "good" one $\mathbf{y}$ (see Fig. (12)).

Basically, if the split and merge kernel $Q_{SM}(\mathbf{x}, .)$ is chosen, with a probability $p_M(\mathbf{x})$ detailed below a merge $Q_M(\mathbf{x}, .)$ is proposed, and with a probability $p_S(\mathbf{x})$ a split $Q_S(\mathbf{x}, .)$ is proposed :

$$Q_{SM}(\mathbf{x}, .) = p_M(\mathbf{x}) Q_M(\mathbf{x}, .) + p_S(\mathbf{x}) Q_S(\mathbf{x}, .)$$

▷ **M**erge

As the objects will tend to occupy most of the pixels of the tree class thanks to the likelihood term of the density, but will not compulsorily overlap themselves because of the repulsive energy in the prior term, a merge has to be proposed if two objects are 'close enough'. Let us now define what 'close enough' means.

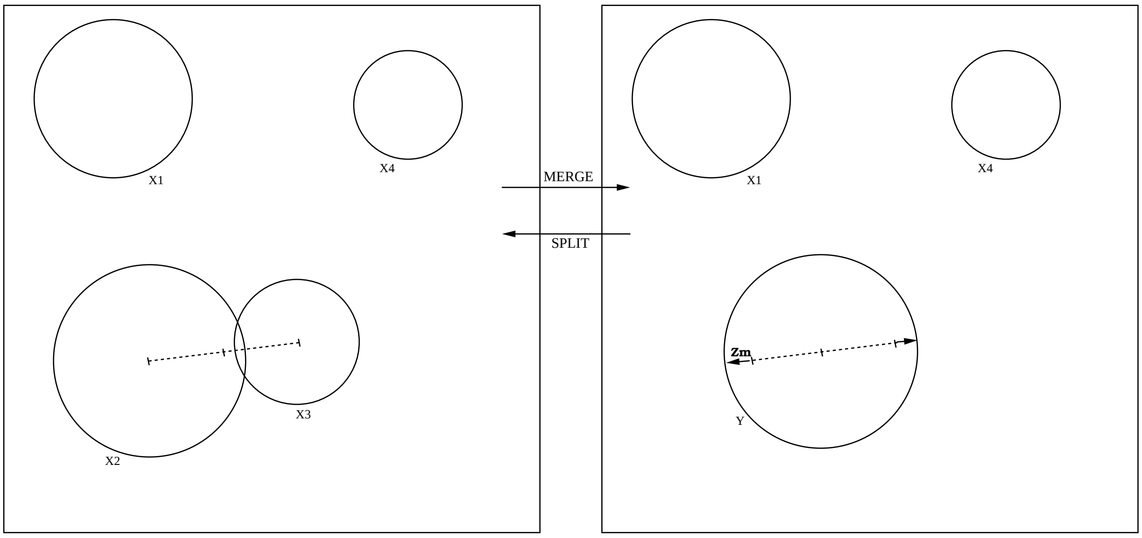

Figure 12: Split and Merge of objects

Definition | ***Mergeable objects***

Two objects $x_1 = (p_1, r_1)$ *and* $x_2 = (p_2, r_2)$ *are mergeable, and we write* $x_1 \sim_m x_2$*, if* $d(p_1, p_2) \leq r_1 + r_2 + \rho$*, with* $\rho = 1$ *in our model.*

A merge will consist of two steps. First, we uniformly choose one pair among the $n_{\sim_m}(\mathbf{x})$ pairs of mergeable objects in $\mathbf{x}$. Suppose that this pair is composed of $x_i = (p_i, r_i)$ and $x_j = (p_j, r_j)$. Then, the map of merging transformation $T^M_{x_i \sim_m x_j}(z_m)$ gives us the new object $y \in S$ according to :

$$y = T^M_{x_i \sim_m x_j}(z_m) = \left(\frac{p_i + p_j}{2}, \frac{r_i + r_j}{2} + z_m \right)$$

where z_m is a random variable uniformly distributed in $Z_m = [0, R_M - \frac{r_i + r_j}{2}]$. The position p_y of the center of the new object y will also be the middle of the positions of the two former ones, and a dilation will be proposed to the radius. We fix $M_{Z_m} = \lambda(Z_m)$.

The associated proposition kernel is, $\forall B \in \mathcal{B}(\mathbf{\Psi})$:

$$Q_M(\mathbf{x}, B) = \frac{1}{n_{\sim_m}(\mathbf{x})} \frac{1}{M_{Z_m}} \sum_{x_i \sim_m x_j} \int_{Z_m} \mathbf{1}_B(\mathbf{y} = \mathbf{x} \setminus \{x_i, x_j\} \cup \{T^M_{x_i \sim_m x_j}(z_m)\}) \lambda(dz_m) \tag{19}$$

▷ **S**plit

The split is the opposite. First, we uniformly choose one object $x_i = (p_i, r_i)$ in the configuration $\mathbf{x}$, and we propose to replace it by two mergeable objects y_1 and y_2 according to $T^S_{x_i}(z_{xy}, z_r, z_s)$:

$$[y_1, y_2] = T^S_{x_i}(z_{xy}, z_r, z_s) = [(p_i + z_{xy}, r_i - z_r + z_s), (p_i - z_{xy}, r_i - z_r - z_s)]$$

where z_r is a random variable uniformly distributed in $Z_r = [0, r_i - R_m]$, z_s a random variable uniformly distributed in $Z_s = [0, \min(r_i - z_r - R_m, R_M - r_i + z_r)]$, and $z_{xy} = (z_x, z_y)$ a double random variable uniformly distributed in :

$$Z_{xy} = \left\{ z \,|\, p_i + z \text{ and } p_i - z \in \mathcal{B}_{r_i - z_r + \frac{\rho}{2}}(p_i) \bigcap \mathcal{P} \right\}$$

In other words, Z_{xy} is the biggest set included in $\mathcal{B}_{r_i - z_r + \frac{\rho}{2}}(p_i) \bigcap \mathcal{P}$ for which p_i is the center of symmetry. We fix $M_{Z_r} = \lambda(Z_r)$, $M_{Z_s} = \lambda(Z_s)$, and $M_{Z_{xy}} = \lambda_{\mathcal{P}}(Z_{xy})$.

The associated proposition kernel is : $\forall B \in \mathcal{B}(\mathbf{\Psi})$,

$$Q_S(\mathbf{x}, B) = \frac{1}{n(\mathbf{x})} \sum_{x_i \in \mathbf{x}} \frac{1}{M_{Z_{xy}} M_{Z_r} M_{Z_s}} \int_{\mathbf{Z}} \mathbf{1}_B(\mathbf{y} = \mathbf{x} \setminus \{x_i\} \cup \{T^S_{x_i}(z_{xy}, z_r, z_s)\}) \lambda^2(dz_{xy}) \lambda(dz_r) \lambda(dz_s) \tag{20}$$

with $\mathbf{Z} = Z_{xy} \times Z_r \times Z_s$.

▷ **G**reen's ratios of the split and merge kernel

The symmetric measure that dominates $\mathcal{P}_X(d\mathbf{x}) Q_{SM}(\mathbf{x}, d\mathbf{y})$ is given in Appendix A.1, as for the details about the calculation of Green's ratios. We obtain in the case of a merge :

$$R_{SM}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{x_i, x_j\} \cup \{y\}) = \frac{p_S(\mathbf{y})}{p_M(\mathbf{x})} \frac{f(\mathbf{y})}{f(\mathbf{x})} \frac{n_{\sim_m}(\mathbf{x}) M_{Z_m}}{8 \beta n(\mathbf{y}) M_{Z_{xy}} M_{Z_r} M_{Z_s}}$$

and for a split :

$$R_{SM}(\mathbf{x},\mathbf{y}=\mathbf{x}\setminus\{x_i\}\cup\{y_1,y_2\}) = \frac{p_M(\mathbf{y})}{p_S(\mathbf{x})}\frac{f(\mathbf{y})}{f(\mathbf{x})}\frac{8\beta n(\mathbf{x})M_{Z_{xy}}M_{Z_r}M_{Z_s}}{n_{\sim_m}(\mathbf{y}))M_{Z_m}}$$

If we choose $p_M(\mathbf{x}) = \frac{n_{\sim_m}(\mathbf{x})}{n_{\sim_m}(\mathbf{x})+n(\mathbf{x})}$ and $p_S(\mathbf{x}) = \frac{n(\mathbf{x})}{n_{\sim_m}(\mathbf{x})+n(\mathbf{x})}$, which is understandable because we do not need to propose a merge if there is no mergeable objects, the ratios eventually become :

$$R_{SM}(\mathbf{x},\mathbf{y}=\mathbf{x}\setminus\{x_i,x_j\}\cup\{y\}) = \frac{n_{\sim_m}(\mathbf{x})+n(\mathbf{x})}{n_{\sim_m}(\mathbf{y})+n(\mathbf{y})}\frac{f(\mathbf{y})}{f(\mathbf{x})}\frac{M_{Z_m}}{8\beta M_{Z_{xy}}M_{Z_r}M_{Z_s}} \tag{21}$$

and

$$R_{SM}(\mathbf{x},\mathbf{y}=\mathbf{x}\setminus\{x_i\}\cup\{y_1,y_2\}) = \frac{n_{\sim_m}(\mathbf{x})+n(\mathbf{x})}{n_{\sim_m}(\mathbf{y})+n(\mathbf{y})}\frac{f(\mathbf{y})}{f(\mathbf{x})}\frac{8\beta M_{Z_{xy}}M_{Z_r}M_{Z_s}}{M_{Z_m}} \tag{22}$$

3.2.3 Birth and Death in a neighbourhood

The birth and death in a neighbourhood kernel has been first developed in [ODZ03b] for an application to building extraction. It was shown that in an image with specific areas of interest (with higher density of objects in some part of it), the convergence of the Markov chain could be speeded up by a kernel that proposes some births, and also some deaths for the reversibility, in the neighbourhood of already extracted objects. We adapt it here and generalize it, in order to enable the birth or the death of up to 4 objects in one single step.

Basically, if the birth and death in a neighbourhood kernel $Q_{BDN}(\mathbf{x},.)$ is chosen, we first select uniformly one object $x=(p,r)$ in the configuration $\mathbf{x}$. Then, we check if this object has already some $\sim_a$-neighbours located in its four neighbourhoods $\mathcal{B}_\epsilon\left(p\pm\overrightarrow{v_i}\right)\bigcap\mathcal{P}$, where $i=\{1,2\}$, and where $\pm\overrightarrow{v_i}$ are the four vectors of alignment.

In order to simplify the notations, we will replace the four meighbourhoods $\mathcal{B}_\epsilon\left(p\pm\overrightarrow{v_i}\right)\bigcap\mathcal{P}$ by $\mathcal{B}_k^{x\in\mathbf{x}}$, with ($k\in\{1,2,3,4\}$). Moreover, we will only keep in the following the sets $\mathcal{B}_k^{x\in\mathbf{x}}$ which have a positive measure $\lambda_{\mathcal{P}}(\mathcal{B}_k^{x\in\mathbf{x}})$ (indeed, some neighbourhoods could be outside $\mathcal{P}$).

Then, we calculate the number of free sites $n_f(x,\mathbf{x})$ around x, ie. the number of neighbourhoods where there is no object inside, and the number of unique sites $n_u(x,\mathbf{x})$, where there is exactly one object inside. We note $\mathcal{B}_f^{x\in\mathbf{x}}$ the union of the balls that are free sites in the neighbourhood of $x \in \mathbf{x}$, and $\mathcal{B}_u^{x \in \mathbf{x}}$ the union of the balls that are unique sites.

From that point, if $n_f(x, \mathbf{x}) + n_u(x, \mathbf{x}) = 0$, ie. if there is neither free, nor unique sites, we propose to stay at $\mathbf{x}$. Otherwise, we calculate 8 coefficients that will be the probability to propose up to 4 births and up to 4 deaths :

$$p_{B_i}(x, \mathbf{x}) = \begin{cases} \frac{1}{n_f(x,\mathbf{x}) + n_u(x,\mathbf{x})} & \text{if } i \leq n_f(x, \mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \quad , \forall i \in \{1, 2, 3, 4\}$$

$$p_{D_i}(x, \mathbf{x}) = \begin{cases} \frac{1}{n_f(x,\mathbf{x}) + n_u(x,\mathbf{x})} & \text{if } i \leq n_u(x, \mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \quad , \forall i \in \{1, 2, 3, 4\}$$

Note that $n_{fu}(x, \mathbf{x}) = n_f(x, \mathbf{x}) + n_u(x, \mathbf{x}) \leq 4$, which means that at most 4 of the previous probabilities are positive.

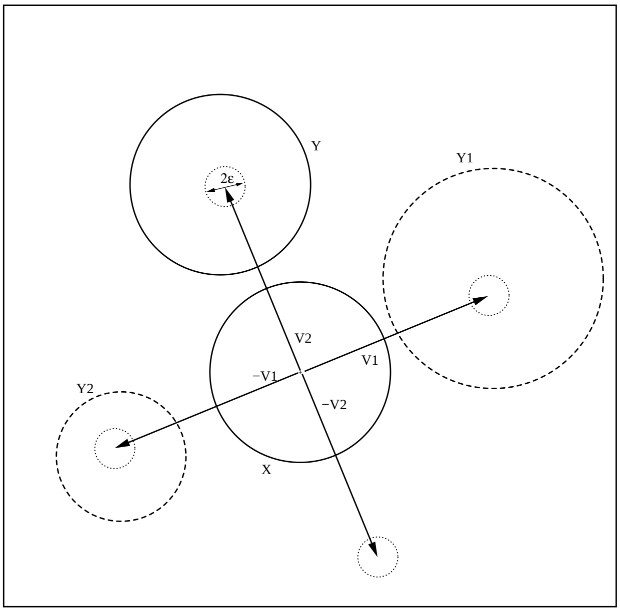

Figure 13: Births or deaths of y_1 and y_2 in the neighbourhood of $x \in \mathbf{x}$.

We thus propose some birth(s) in the free sites, and some death(s) in the unique sites. The proposition kernel of the birth and death in a neighbourhood $Q_{BDN}(\mathbf{x}, .)$

is, $\forall B \in \mathcal{B}(\boldsymbol{\Psi})$:

$$Q_{BDN}(\mathbf{x}, B) = \frac{1}{n(\mathbf{x})} \sum_{x \in \mathbf{x}} \sum_{i=1}^{4} p_{B_i}(x, \mathbf{x}) Q_{B_i}(x, \mathbf{x}, B) + \sum_{i=1}^{4} p_{D_i}(x, \mathbf{x}) Q_{D_i}(x, \mathbf{x}, B) \quad (23)$$

where $Q_{B_i}(x, \mathbf{x}, .)$ proposes the birth(s) of i objects in $\mathcal{B}_f^{x \in \mathbf{x}}$ and $Q_{D_i}(x, \mathbf{x}, .)$ the death(s) of i objects in $\mathcal{B}_u^{x \in \mathbf{x}}$ (see Fig. (13)).

$\triangleright$ **B**irths in a neighbourhood

In this part, we detail the sub-kernel $Q_{B_i}(x, \mathbf{x}, .)$, which proposes the birth(s) of i objects in $\mathcal{B}_f^{x \in \mathbf{x}}$.

First, we choose uniformly the i sites among the $i \leq n_f(x, \mathbf{x})$ free sites. There is $C^i_{n_f(x,\mathbf{x})}$ possible combinations for them, and we note $\mathcal{C}(i, n_f(x, \mathbf{x}))$ the union of all these combinations. Let us say that $\mathcal{J} = (k_1, \ldots, k_i) \in \mathcal{C}(i, n_f(x, \mathbf{x}))$ has been chosen. Then, for each site $k \in \mathcal{J}$ elected, we add an object $y_k = (z_{x_k}, z_{y_k}, z_{r_k}) \in \mathcal{B}_k^{x \in \mathbf{x}}$ thanks to the introduction of 3 random variables $(z_{x_k}, z_{y_k}, z_{r_k}) \in \mathcal{B}_k^{x \in \mathbf{x}} \times \mathcal{K}$. The Lebesgue measure of $\left(\mathcal{B}_k^{x \in \mathbf{x}} \times \mathcal{K}\right) \in S$ will be noted $M_k = \lambda_S\left(\mathcal{B}_k^{x \in \mathbf{x}} \times \mathcal{K}\right)$.

The associated proposition kernel of the births of i objects in the free neighbourhoods of $x \in \mathbf{x}$ is also : $\forall B \in \mathcal{B}(\boldsymbol{\Psi})$,

$$\begin{aligned} Q_{B_i}(x, \mathbf{x}, B) \quad = \quad & \frac{1}{C^i_{n_f(x,\mathbf{x})}} \sum_{\mathcal{J} \in \mathcal{C}(i, n_f(x,\mathbf{x}))} \frac{1}{M_{k_1} \ldots M_{k_i}} \int_{\mathcal{B}_{k_1} \times \mathcal{K} \times \ldots \times \mathcal{B}_{k_i} \times \mathcal{K}} \mathbf{1}_B(\mathbf{y} = \mathbf{x} \cup \{y_{j_1}, \ldots, y_{j_i}\}) \\ & \lambda(dz_{x_{k_1}}) \lambda(dz_{y_{k_1}}) \lambda(dz_{r_{k_1}}) \ldots \lambda(dz_{x_{k_i}}) \lambda(dz_{y_{k_i}}) \lambda(dz_{r_{k_i}}) \end{aligned}$$

$\triangleright$ **D**eaths in a neighbourhood

In this part, we detail the sub-kernel $Q_{D_i}(x, \mathbf{x}, .)$ which proposes the death(s) of i objects in $\mathcal{B}_u^{x \in \mathbf{x}}$.

First, we uniformly choose the i sites among the $n_u(x, \mathbf{x}) \geq i$ unique sites. There is $C^i_{n_u(x,\mathbf{x})}$ possible combinations for them, and we still note $\mathcal{C}(i, n_u(x, \mathbf{x}))$ the union of all these combinations. Let us say that $\mathcal{J} = (k_1, \ldots, k_i) \in \mathcal{C}(i, n_u(x, \mathbf{x}))$ has been chosen. Then, for each site $k \in \mathcal{J}$ elected, we remove the $\sim_a$-neighbour of x that belongs to $\mathcal{B}_k^{x\in\mathbf{x}} \times \mathcal{K}$.

The associated proposition kernel of the deaths of i objects in the unique neighbourhood of $x \in \mathbf{x}$ is also, $\forall B \in \mathcal{B}(\mathbf{\Psi})$:

$$Q_{D_i}(x, \mathbf{x}, B) = \frac{1}{C^i_{n_u(x,\mathbf{x})}} \sum_{\mathcal{J}\in\mathcal{C}(i,n_u(x,\mathbf{x}))} \mathbf{1}_B(\mathbf{y} = \mathbf{x} \setminus \{y_{k_1}, \dots, y_{k_i}\})$$

▷ **G**reen's ratio

The symmetric measure that dominates $\mathcal{P}_X(d\mathbf{x})Q_{BDN}(\mathbf{x}, d\mathbf{y})$ is given in Appendix A.1, as for the details about the calculation of Green's ratios. As there could be several ways to go from $\mathbf{x}$ to $\mathbf{y}$, we note $\mathcal{G}(\mathbf{x} \to \mathbf{y})$ the group of objects $x \in \mathbf{x}$ which, by means of births or deaths in their neighbourhood, enable to obtain $\mathbf{y}$. In the case of i births in a neighbourhood, the Green's ratio is written as follows :

$$R_{BDN}(\mathbf{x}, \mathbf{y} = \mathbf{x} \cup \{y_{k_1}, \dots, y_{k_i}\}) = \frac{n(\mathbf{x})}{n(\mathbf{y})} \frac{f(\mathbf{y})}{f(\mathbf{x})} \frac{\beta^i \sum_{x\in\mathcal{G}(\mathbf{y}\to\mathbf{x})} \frac{1}{n_{fu}(x,\mathbf{y})} \frac{1}{C^i_{n_u(x,\mathbf{y})}}}{\sum_{x\in\mathcal{G}(\mathbf{x}\to\mathbf{y})} \frac{1}{n_{fu}(x,\mathbf{x})} \frac{1}{C^i_{n_f(x,\mathbf{x})}} \frac{1}{M_{k_1} \dots M_{k_i}}} \tag{24}$$

$$R_{BDN}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{x_{k_1}, \dots, x_{k_i}\}) = \frac{n(\mathbf{x})}{n(\mathbf{y})} \frac{f(\mathbf{y})}{f(\mathbf{x})} \frac{\sum_{x\in\mathcal{G}(\mathbf{y}\to\mathbf{x})} \frac{1}{n_{fu}(x,\mathbf{y})} \frac{1}{C^i_{n_f(x,\mathbf{y})}} \frac{1}{M_{k_1} \dots M_{k_i}}}{\beta^i \sum_{x\in\mathcal{G}(\mathbf{x}\to\mathbf{y})} \frac{1}{n_{fu}(x,\mathbf{x})} \frac{1}{C^i_{n_u(x,\mathbf{x})}}} \tag{25}$$

3.2.4 Convergence proofs

We need to prove that the RJMCMC algorithm presented above makes the Markov chain X_t ergodically converge to the target distribution $\mathcal{P}_X(.)$. The reversibility is obtained thanks to the choice of the acceptance ratio at each step. Then, the irreducibility, the Harris recurrence and the geometric ergodicity are consequences of the stability condition (4), which has been demonstrated in (15). Details of the proof can be found in Appendix A.1.

3.2.5 Convergence results

In order to check the validity of the different kernels we added to the sampler, we propose to simulate a Poisson point process and to compare the statistics we obtain with the statistics of a Poisson point process simulated with the two-step algorithm presented before.

Thus, we consider a Poisson point process X with density $f(\mathbf{x}) = 1$ with respect to the reference Poisson process of intensity measure $\nu(d\mathbf{x}) = \beta\lambda_S(d\mathbf{x})$. For each Borel set $B \in S$, the random variable $N(B)$, that counts the number of points of X falling in B, should be Poisson law distributed with mean $\nu(B)$. Moreover, for k Borel sets B_k, these random variables should be independently distributed.

We also simulate a Poisson point process lying on $S = [0, 100] \times [0, 100]$. For each Borel set $B \in S$, the expectation of $N(B)$ is

$$\mathbb{E}[N(B)] = \beta\nu(B)$$

and we set β so as $\mathbb{E}[N(S)] = 100$. As we cannot check the independence of $N(B_k)$ in general, we propose to take a particular case of four disjoint Borel sets B_k (see Fig. (14)) that split S in 4 parts :

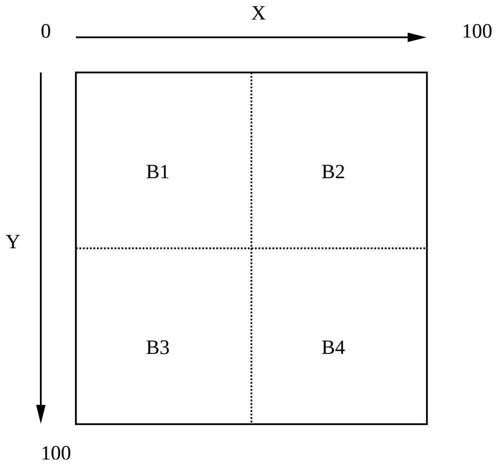

Figure 14: 4 disjoint Borel sets $B_k \in S = [0, 100] \times [0, 100]$, $k \in \{1, 2, 3, 4\}$.

$$\begin{aligned} B_1 &= \,]0,50[\times]0,50[\\ B_2 &= \,]50,100[\times]0,50[\\ B_3 &= \,]0,50[\times]50,100[\\ B_4 &= \,]50,100[\times]50,100[\end{aligned}$$

Then, we compute the mean and the variance of $N(B_k)$, and also the covariance matrix of the random variables $N(B_k)$, in order to compare it with the one obtained with a Poisson point process generated with the two-step algorithm.

On the one hand, we sample $N = 500.000$ realizations of a Poisson point process of expectation $N(S) = 100$ with the two-step algorithm, and for each realization X_i, we compute the associated $N_{X_i}(B_k)$ in order to get the covariance matrix :

$$Cov(N(B_k), N(B_l)) = \sum_{i=1}^{N} (N_{X_i}(B_k) - \mathbb{E}[N(B_k)])\,(N_{X_i}(B_l) - \mathbb{E}[N(B_l)])$$

On the other hand, we run our RJMCMC sampler, and select $N = 500.000$ samples of the Markov chain after having reached the equilibrium ($N_{eq} = 1.000.000$ iterations). These samples are taken every N_{it} iterations, where N_{it} is detailed hereafter. The total cost of such a simulation is also $N_{eq} + N_{it} * N$ iterations.

We have performed two tests with different kernels : birth and death only (Test 1), and the global kernel (Test 2). The Poisson process simulated with the two-step algorithm will be taken as a reference.

▷ **T**he two-step algorithm reference

The covariance matrix of the 4 random variables $N(B_k)$ is given in Tab. (1). Moreover, Fig. (15) shows the graphs of the distribution of $N(S)$ and some of the $N(B_k)$: they fit well with the discrete Poisson law function of parameters $\nu(S) = 100$ and $\nu(B_k) = 25$.

▷ **T**est 1 : sampler with Q_{BD} only

First we set $N_{it} = 20$. The statistics of the random variables $N(B_k)$ and $N(S)$ are presented in Tab. (2) (see Fig. (16)), and the covariance matrix of the 4 random variables $N(B_k)$ in Tab. (3).

	$N(B_1)$	$N(B_1)$	$N(B_1)$	$N(B_1)$
$N(B_1)$	24.9897	-0.0208	0.0034	0.0255
$N(B_1)$	-0.0208	24.9983	-0.0388	-0.0151
$N(B_1)$	0.0034	-0.0388	24.9227	0.0744
$N(B_1)$	0.0255	-0.0151	0.0744	25.0038

Table 1: The two-step algorithm. Covariance matrix of $N(B_k)$

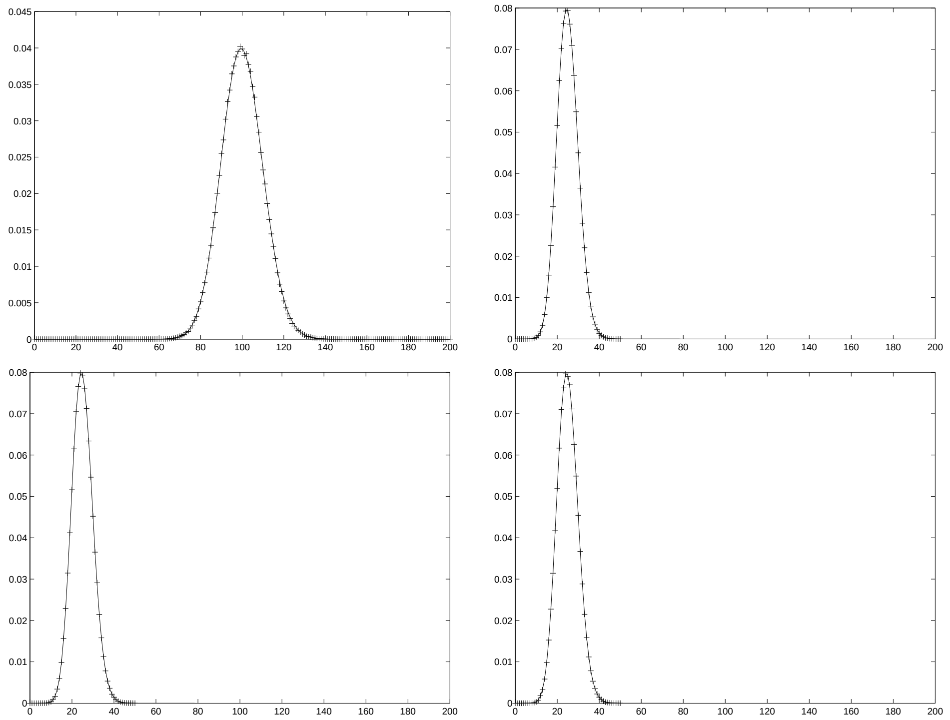

Figure 15: The two-step algorithm. Top Left : crosses represent the ditribution of $N_i(S)$, compared to the Poisson law of parameter $\nu(S) = 100$ in red. Top Right : the same with B_1. Bottom Left : the same with B_2. Bottom Right : the same with B_3.

	Mean	Variance
$N(S)$	100.0458	101.0781
$N(B_1)$	24.9747	25.2039
$N(B_2)$	25.0273	25.2530
$N(B_3)$	25.0321	25.1387
$N(B_4)$	25.0116	24.6765

Table 2: With Q_{BD} only and $N_{it} = 20$. Statistics of $N(S)$ and $N(B_k)$.

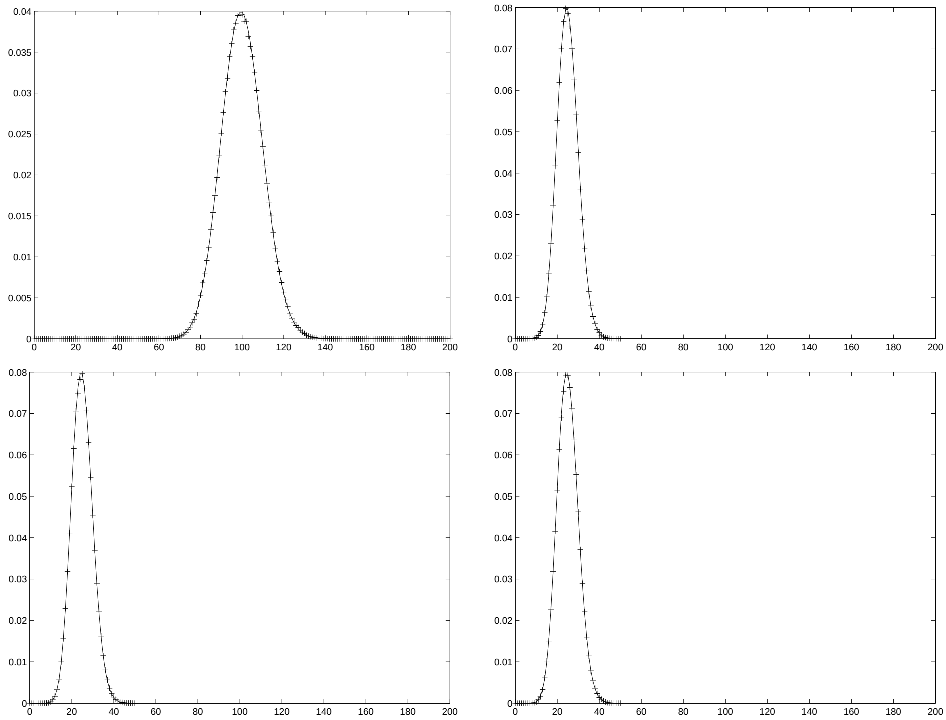

Figure 16: With Q_{BD} only and $N_{it} = 20$. Top Left : crosses represent the distribution of $N_i(S)$, compared to the Poisson law of parameter $\nu(S) = 100$ in red. Top Right : the same with B_1. Bottom Left : the same with B_2. Bottom Right : the same with B_3.

	$N(B_1)$	$N(B_2)$	$N(B_3)$	$N(B_4)$
$N(B_1)$	25.2039	0.0401	0.0355	0.1483
$N(B_2)$	0.0401	25.230	-0.1389	0.0672
$N(B_3)$	0.0355	-0.1389	25.1387	0.2509
$N(B_4)$	0.1483	0.0672	0.2509	24.6765

Table 3: With Q_{BD} only and $N_{it} = 20$. Covariance matrix of $N(B_k)$.

	Mean	Variance
$N(S)$	99.9680	99.9208
$N(B_1)$	24.9852	24.9159
$N(B_2)$	24.9926	24.8473
$N(B_3)$	24.9687	25.0660
$N(B_4)$	25.0217	25.1395

Table 4: With Q_{BD} only and $N_{it} = 40$. Statistics of $N(S)$ and $N(B_k)$.

Thus we can see that these statistics are very close to the reference ones. The correlation is quite low, but can still reach some value greater than 0.2 because of the correlation that exists in the Markov chain between two consecutive samples. Indeed, in $N_{it} = 20$ iterations, most of the objects of the scene are the same because either no move has been proposed to them, or perhaps it has been refused. We could wait more between two samples in order to avoid this high correlation.

That is why we now set $N_{it} = 40$. The statistics of the random variables $N(B_k)$ and $N(S)$ are in Tab. (4), while the covariance matrix of the 4 random variables $N(B_k)$ is given in Tab. (5).

This is better decorrelated.

▷ **T**est 2 : global sampler

Here we directly set $N_{it} = 80$, because the statistics obtained $N_{it} = 20$ and $N_{it} = 40$ showed that the samples were too much correlated. The statistics of the random variables $N(B_k)$ and $N(S)$ are in Tab. (6) (see Fig. (17)) and the covariance matrix

	$N(B_1)$	$N(B_2)$	$N(B_3)$	$N(B_4)$
$N(B_1)$	24.9159	-0.0487	0.0678	-0.0323
$N(B_2)$	-0.0487	24.8473	-0.0212	0.0047
$N(B_3)$	0.0678	-0.0212	25.0660	0.0057
$N(B_4)$	-0.0323	0.0047	0.0057	25.1395

Table 5: With Q_{BD} only and $N_{it} = 40$. Covariance matrix of $N(B_k)$.

of the 4 random variables $N(B_k)$ is given in Tab. (7).

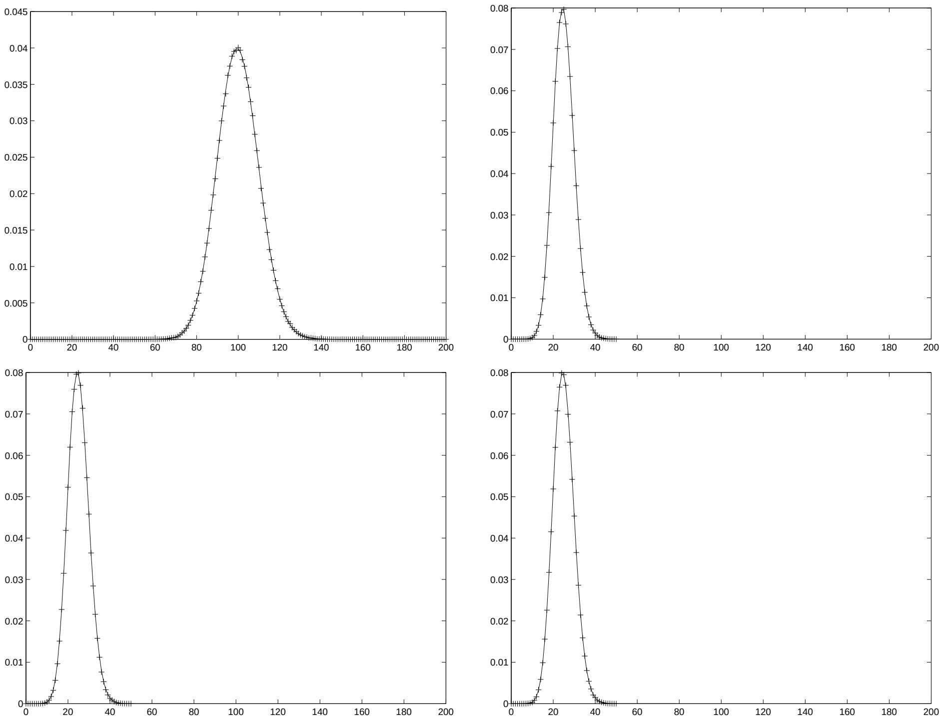

Figure 17: With the full kernel and $N_{it} = 80$. Top Left : crosses represent the distribution of $N_i(S)$, compared to the Poisson law of parameter $\nu(S) = 100$ in red. Top Right : the same with B_1. Bottom Left : the same with B_2. Bottom Right : the same with B_3.

	Mean	Variance
$N(S)$	100.0311	100.7981
$N(B_1)$	25.0182	25.0858
$N(B_2)$	24.9926	24.7570
$N(B_3)$	24.9977	25.0096
$N(B_4)$	25.0224	25.0642

Table 6: With the full kernel and $N_{it} = 80$. Statistics of $N(S)$ and $N(B_k)$.

	$N(B_1)$	$N(B_2)$	$N(B_3)$	$N(B_4)$
$N(B_1)$	25.0858	0.1081	-0.0247	0.0949
$N(B_2)$	0.1081	24.7570	0.0270	0.0536
$N(B_3)$	-0.0247	0.0270	25.0096	0.1819
$N(B_4)$	0.0949	0.0536	0.1819	25.0642

Table 7: With the full kernel and $N_{it} = 80$. Covariance matrix of $N(B_k)$.

▷ Conclusion

The results obtained with a Poisson point process exhibit some satisfying statistics. For example, an error in one of the Green's ratio would have been detected with these tests.

3.3 Results on aerial images

In this part we present some results obtained on aerial images of French forests provided by IFN (Inventaire Forestier National). As presented above, our model contains several parameters that should be fixed. This will be done before presenting the results and some statistics about them. As there is no ground truth, we will try to assess the correctness of the different extractions on our own.

3.3.1 Parameters

▷ State space S

We recall that $S = \mathcal{P} \times \mathcal{K}$, where $\mathcal{P} = [0, X_M] \times [0, Y_M]$ corresponds to the size of the data. The resolution of the images is approximatively 50cm/pixel, so the global area of interest in m^2 is approximatively $\frac{X_M Y_M}{4} m^2$.

The parameters of $\mathcal{K} = [R_m, R_M]$ depend on the tree size (species, age) and the image resolution, but they can be calibrated. Most of the time, we took $R_m = 3$ and $R_M = 7$. We will see in the discussion how we can cope with bigger R_M.

$\mathcal{K} = [R_m, R_M]$
$R_m = 3$ $R_M = 7$

Table 8: Dimensions of $\mathcal{K}$

▷ **P**rior density $f_p(\mathbf{x})$

The relations of repulsion and alignment, as defined before, require a parameter that ponders them in the total prior energy. Moreover, with the Fourier transform of the image, we can obtain the vectors $\overrightarrow{v_i}$ that define the neighbourhood of alignment. These two parameters are the only ones to settle in our model.

$U_p(\mathbf{x}) = \gamma_r \sum_{x_i \sim_r x_j} \mathcal{A}(x_i, x_j) + \gamma_a \sum_{x_i \sim_a x_j} \mathcal{Q}(x_i, x_j) + U_h(\mathbf{x})$
$\gamma_r = 1000$ $\gamma_a = -10$

Table 9: Prior density parameters

▷ **R**eference measure

The reference measure parameter β is set so as the expectation of the number of points in the Poisson process of intensity measure $\nu(.) = \beta.\lambda(.)$ corresponds to the number of points of the final configuration. In practice, with the vectors of alignment we can approximatively estimate the density of trees per unit of area, and have a good guess for β.

▷ **L**ikelihood

The parameters of the Gaussian classes can be obtained thanks to a K-means algorithm. So for each image these parameters are different and estimated.

▷ **S**ampler probabilities

First, the mixture parameters such as p_B and p_D in Q_{BD} are equal : $p_B = p_D = 0.5$. Then, the global parameters $p_m(\mathbf{x})$ that correspond to the probability to choose the kernel $Q_m(\mathbf{x}, .)$ correspond to the Tab. (10).

Uniform birth and death Q_{BD}
$p_{BD} = \frac{1}{10}$
Translation Q_T
$p_T = \frac{2}{10}$
Dilation Q_D
$p_D = \frac{2}{10}$
Translation and Dilation Q_{TD}
$p_{TD} = \frac{2}{10}$
Split and Merge Q_{SM}
$p_{TD} = \frac{2}{10}$
Birth and death in a neighbourhood Q_{BDN}
$p_{BDN} = \frac{1}{10}$

Table 10: Parameters of the global kernel

3.3.2 Results

Some results are shown in Fig. (18) and Fig. (19). The images are 50cm/pixel resolution. All these results were obtained after 2 millions iterations ($\simeq$ 1 minute), which represents a fast decrease of the temperature in the simulated annealing scheme. They were performed on a Red Hat Linux 3GHz machine. The CPU time depend on the number of objects to be extracted, and on the size of the image.

In Fig. (19), we extracted the tree crowns on the pre-segmented area of poplars, and pasted it on the original image to show the result.

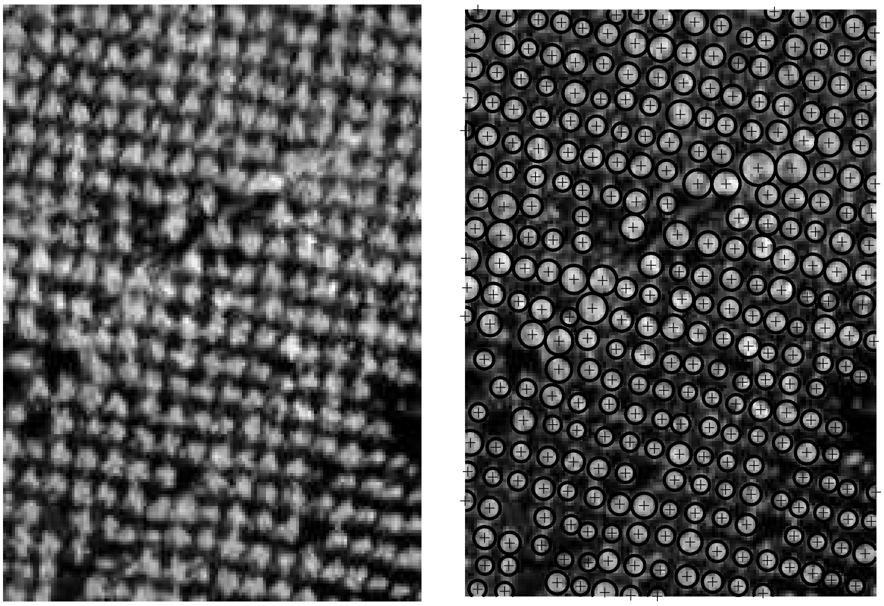

Figure 18: Left : original image. Right : extraction result, in 2 millions iterations (90 seconds).

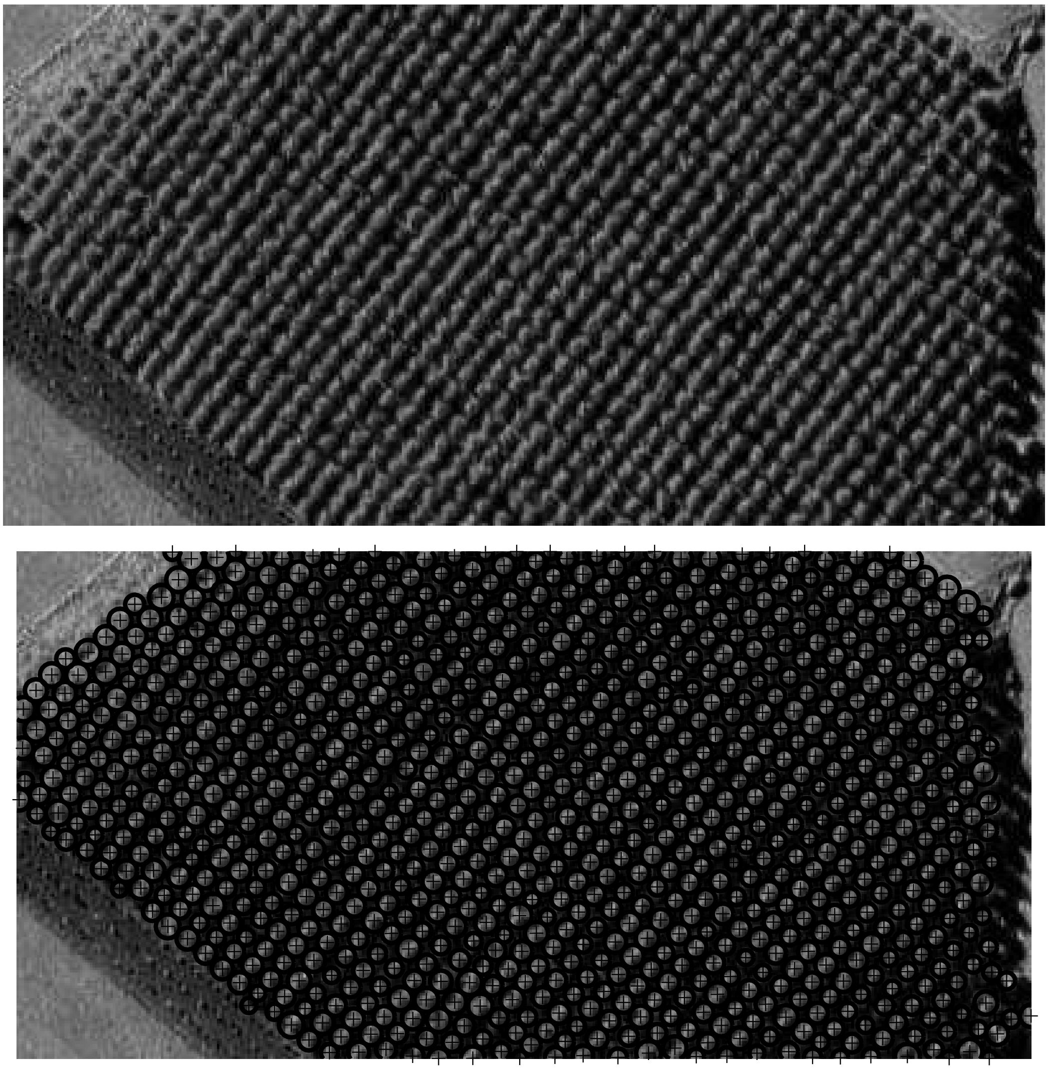

Figure 19: Left : original image. Right : extraction result, in 2 millions iterations (ie. 150 seconds).

3.4 Discussions and improvement

3.4.1 About the data

We worked on grey-level images obtained by taking the red component of the digital images. As the images provided by IFN are analogic photographs scanned by an operator, this component does not exactly correspond to the infrared component. But it is the one which distinguishes the best the forests from the rest of the image.

3.4.2 About the kernel

Some of the sub-kernels such as the split and merge and the birth and death in a neighbourhood are a little bit more time consuming than the other ones, but they have either an impact on the quality of the extraction, or on the cooling schedule we can afford.

For example, Fig. (20) presents two extractions that were processed on the same image and with the same parameters (2.000.000 iterations). The only difference is that in the lefthandside result, there was no split and merge in the proposition kernel. We have several sub-detections, which means that some discs are detecting at least two trees. We are stuck in a local minimum of the global energy $U(\mathbf{x})$, because the only way to go from $\mathbf{x}_a$ to $\mathbf{x}_b$ is to join $\mathbf{x}_c$. The colder the temperature is, the higher the positive energy gap is to go through (see Fig. (21)), therefore a one-step move from $\mathbf{x}_a$ to $\mathbf{x}_b$ is needed to leave this mimimum. That is why split and merge kernel is useful in our application.

Here are some statistics that we obtained simulating 10 times the above experiment. The mean of the final energy and of the final number of objects extracted show that the split and merge kernel gives us better results.

Statistic	Without Q_{SM}	With Q_{SM}
Iterations ♯	2000000	2000000
Objects detected (mean)	283.7	290.5
Final energy	$U(\mathbf{x}) = 139287$	$U(\mathbf{x}) = 138283$

Table 11: Some statistics to compare simulations with and without the split and merge kernels

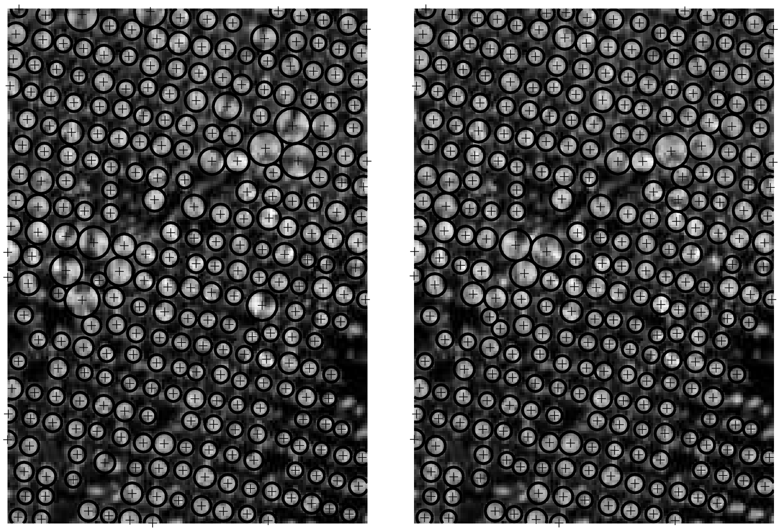

Figure 20: Left : without split and merge kernel. Right : with split and merge kernel.

3.4.3 Sampler efficiency

We can also work on the probabilities to propose the different moves in the global kernel. As shown in Fig. (22), the acceptance ratio for each move considerably decreases when the temperature decreases. It would also be interesting to work on it in order to propose more often some moves when they are about to be accepted.

An heuristic method could be proposed to adapt the probabilities of the proposition kernels during the simulation. It would consist in updating each probability p_i depending on the acceptance probability of each move $Q_i(.,.)$ during N_{up} iterations. In the following, we note p_i^k the probability of proposing the sub-kernel $Q_i(.,.)$ between iteration $k*N_{up}+1$ and iteration $(k+1)*N_{up}$. Let us say that the sub-kernel $Q_i(.,.)$ has been proposed $N_p(i,k)$ times during these N_{up} iterations, and accepted $N_a(i,k)$ times. Now, considering that each kernel should be at least proposed with a probability 0.05, the p_i^k would become :

$$\forall i \in [1,\ldots,N_Q]\ ,\ p_i^k = 0.05 + (1 - 0.05 * N_Q)\frac{\frac{N_a(i,k)}{N_p(i,k)}}{\sum_j \frac{N_a(j,k)}{N_p(j,k)}} \tag{26}$$

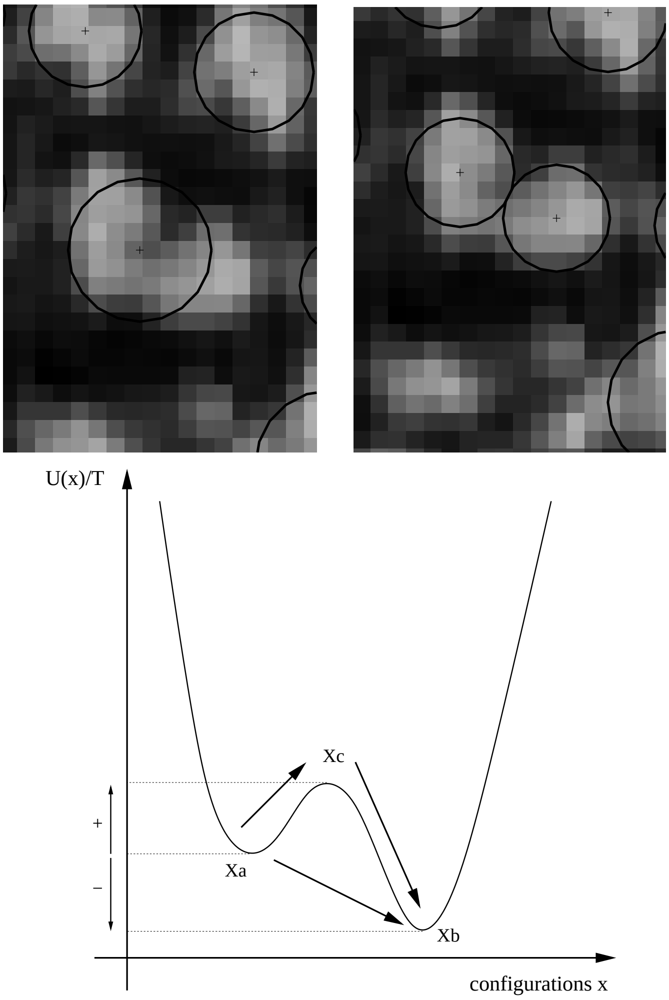

Figure 21: A local minimum ($\mathbf{x}_a$, top lefthandside image) and a global minimum ($\mathbf{x}_b$, top righthandside image)) of an energy and the associated graphic (bottom graph).

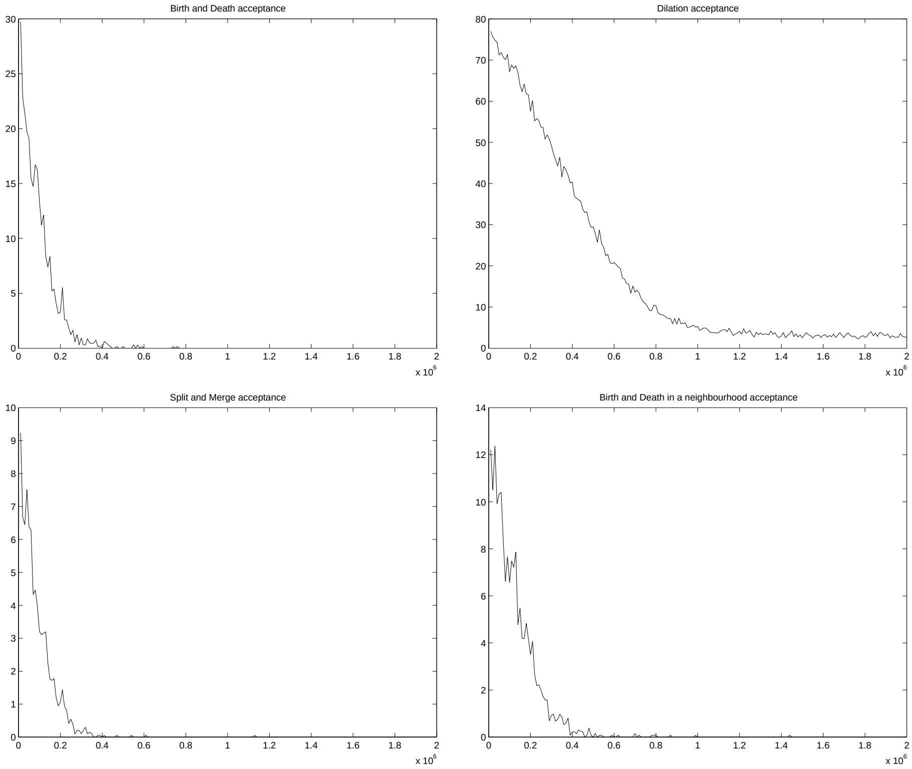

Figure 22: Top left : Acceptance percentage of Birth and Death kernel. Top right : Acceptance percentage of Dilation kernel. Bottom left : Acceptance percentage of Split and Merge kernel. Bottom right : Acceptance percentage of Birth and Death in a neighbourhood kernel

where N_Q is the number of sub-kernels (supposed to be less than 20) that compose the global proposition kernel. Changing the p_i slightly affects our Green ratios every $k * N_{up}$ iterations, by adding in each of them an extra factor $\frac{p_i^{k+1}}{p_i^k}$ which takes into account these updates. As shown in Fig. (23), simple moves like dilation and translation are the most accepted, since birth and death or split and merge tend to rarefy at the end of the simulated annealing.

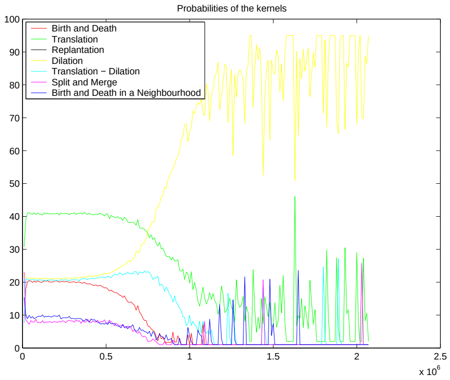

Figure 23: Evolution of the different probabilities of propositions with the adaptive scheme.

3.4.4 Cooling schedule of the simulated annealing

A key element in our simulations is the simulated annealing scheme itself, with the choice of the initial temperature and the decrease schedule of the temperature, as pointed out in [Ort04] or [SGM04] for example. The concept of critical temperature has been pointed out, this is the temperature where the systems chooses its local minimum of energy. Then, adaptive cooling schedules tend to make the Markov chain spend more time at this temperature to select the correct minimum.

Moreover, the choice of the initial temperature is crucial : choosing a too small temperature would freeze the configurations in a local minima of $U(\mathbf{x})$, while choosing a too high one would make us wait much longer before convergence. The idea is actually to find a first temperature T_0 that 'corresponds' to the first configuration of objects $\mathbf{x}_0$, which means that the statistics of $\mathbf{x}_0$ should be representative of $f(\mathbf{x})^{\frac{1}{T_0}}$.

But this problem is linked with the decrease of the temperature, because one can select a low initial temperature, then wait more time until the equilibrium is obtained, and finally decrease the temperature at a correct speed to obtain a good extraction.

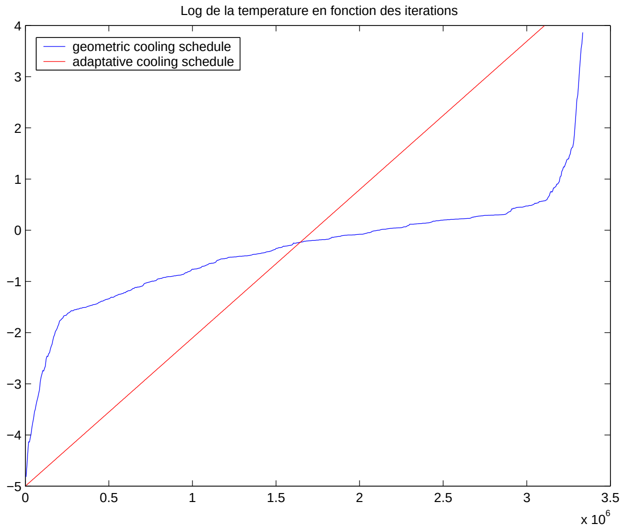

Figure 24: Adaptive cooling schedule of the temperature, compared to the geometric one.

Using an adaptive cooling schedule as proposed in [Ort04], we obtain better results with the same number of iterations (3 millions), and the same starting temperature 10^5 and final temperature 10^{-4}. Fig. (24) presents the different cooling schedules that we studied, while Tab. (12) gives the final energy for these different simulations. The 3 geometric decreases have different geometric coefficient $\alpha < 1$, and also different platform lengths. The higher α is, the fewer time we spend at the temperature $T_0\alpha^k$ (platform k).

3.4.5 Initialization $\mathbf{x}_0$

We always initialized our algorithm with an empty configuration. However, it could be interesting to initialize it with a grid of objects, thanks to the periodic vectors we have and the phase of the Fourier transform (see Fig. (25)). It is then possible to start the simulated annealing with a colder temperature.

Model	Final Energy
Geometric decrease 1 : $\alpha = 0.999$	$U(\mathbf{x}) = 138337$
Geometric decrease 2 : $\alpha = 0.99$	$U(\mathbf{x}) = 138143$
Geometric decrease 3 : $\alpha = 0.95$	$U(\mathbf{x}) = 138146$
Adaptive decrease	$U = 137812$

Table 12: Different energies obtained in 3 millions iterations, with the adaptive cooling schedule, and with 3 geometric cooling schedules.

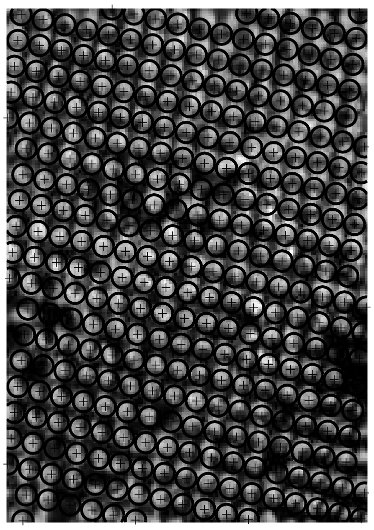

Figure 25: A possible initialization for the Markov chain

3.4.6 About the prior term

We can study the importance of each part of the prior term of the density :

$$U_p(\mathbf{x}) = U_r(\mathbf{x}) + U_a(\mathbf{x}) + U_h(\mathbf{x})$$

Without the repulsion term, the marked point process tends to put a maximum number of objects inside the trees and is of no interest to us. Hopefully, the hard core avoids the process to explode, the total number of marked points is bounded. Without the attractive term, it becomes more interesting to analyse what is happening. We can see that this attractive term in the density does not have a big impact on the number of points, but only on their distribution in the state space. They are not as well aligned as they are in the full model. Tab. (13) summarizes these results.

Statistic	$\gamma_r = 1000$ and $\gamma_a = 0$	$\gamma_r = 1000$ and $\gamma_a = -10$
Iterations ♯	2000000	2000000
Objects detected (mean)	290.2	290.9
$U_r(\mathbf{x})$ ($\gamma_r = 1000$)	19	27
$\sum_{x_i \sim_a x_j} \mathcal{Q}(x_i, x_j)$ $U_a(\mathbf{x})$	147.4 0	180.6 -1806
Final energy	$U(\mathbf{x}) = 138585$	$U(\mathbf{x}) = 138283$

Table 13: Some statistics to compare simulations with and without the attractive term in $U_p(\mathbf{x})$

3.4.7 Parameters

Some parameters of the model like the ones of the two Gaussian classes can be estimated thanks to a K-means algorithm. We can also make some parameter estimation for the parameters of the prior density, because our Bayesian model is very practical to do so (see [GT92, DvLSZ01]).

However, the parameters of $\mathcal{K} = [R_m, R_M]$, the space of the marks, are difficult to estimate. They have a big impact on the final configuration (see Fig. (26) and Tab. (14)) : it is also not sufficient to set a very small R_m and a very big R_M. They have to be carefully calibrated, with a first small temperature extraction for example

Statistic	$\mathcal{K} = [1,7]$	$\mathcal{K} = [3,10]$
Iterations $\sharp$	2000000	2000000
Objects detected	386.9	221.2
Final energy	$U(\mathbf{x}) = 136271$	$U(\mathbf{x}) = 147886$

Table 14: Some statistics to compare simulations with different spaces $\mathcal{K}$

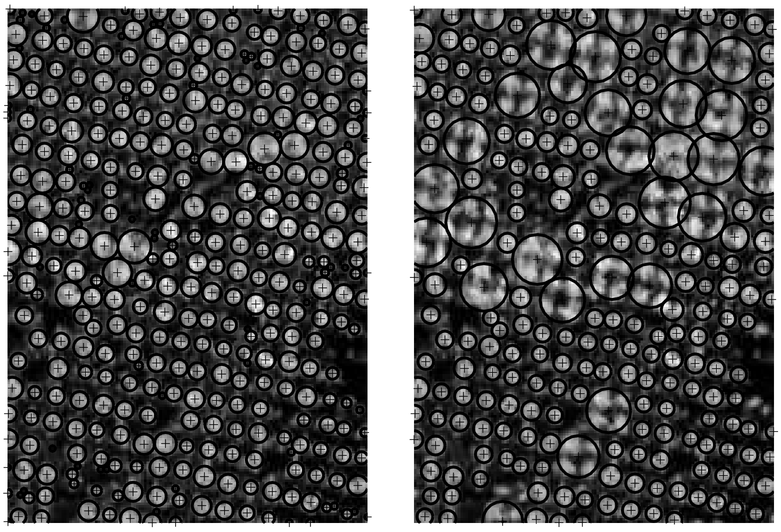

Figure 26: Left : an extraction of tree crowns with $R_m = 1$, 384 objects detected. Right : an extraction of tree crowns with $R_M = 10$, 222 objects detected.

that could give us a range of possible radius, or with other algorithms using the Fourier transform for instance.

The split and merge kernel is not good enough to avoid clustering of very small objects in one by tree crown like in Fig. (26), or the opposite case with a big object that contains several trees. It is efficient when there is only two objects to merge, or one to split into two, but when there are more than two, it is not possible. That is why we have to find other solutions. For a too large R_M, one possibility would be to change the likelihood $\mathcal{L}(\mathcal{I}|\mathbf{x})$ as suggested in the next paragraph.

3.4.8 Different likelihood / non Bayesian model

In the model presented above, considering that the pixels are independent with respect to the class they belong to, the likelihood term in the density $f(\mathbf{x})$ is :

$$\mathcal{L}(\mathcal{I}|\mathbf{x}) = \prod_{p \in \mathcal{C}_o} \frac{1}{\sqrt{2\pi}\sigma_o} \exp\left(\frac{-(y_p - \mu_o)^2}{2\sigma_o^2}\right) \prod_{p \in \mathcal{C}_i} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(\frac{-(y_p - \mu_i)^2}{2\sigma_i^2}\right)$$

where y_p is the grey level of the pixel p. With the final configuration of result 1 presented before, Fig. (28) presents one realization of the image with respect to this probability distribution. We can remark that the image obtained is quite different from the original image, some properties are missing.

One of these properties is that the distribution of the pixel values inside a tree is not random, as suggests $\mathcal{L}(\mathcal{I}|\mathbf{x})$. We can observe that depending on the sun position, some of the pixels will appear brighter, and other ones darker, because of the illumination and the geometric shape of the tree crown. That is why we propose two other models for the likelihood, which take into account this point. Supposing that the image is close to the Nadir position, our two new models of density are described in Fig. (27). Both of them consist in first calculating the relative position of the pixel inside the object $x \in \mathbf{x}$, and then associating to this relative position a probability which responds to a Gaussian distribution of mean $\mu_i^l(p, x)$ and variance σ_i^l. The $\mu_i^l(p, x)$ are calculated in order to compute the integrals :

$$\int_0^R \mu_i(p) \frac{2r}{R^2} dr$$

and

$$\int_0^R (\mu_i(p) - \mu_i)^2 \frac{2r}{R^2} dr$$

are respectively equal to μ_i and σ_i, the mean and the variance calculated with the K-means algorithm. The other parameter σ_i^l is only measuring how far from that model of illumination we accept to be.

▷ **L**inear model

In the linear model, the mean $\mu_i(p)$ is linear inside the tree. Let p be a pixel inside one object of the configuration $x = (p_x, r_x) \in \mathbf{x}$, and $d_p(x)$ the distance in $\mathcal{P}$ between

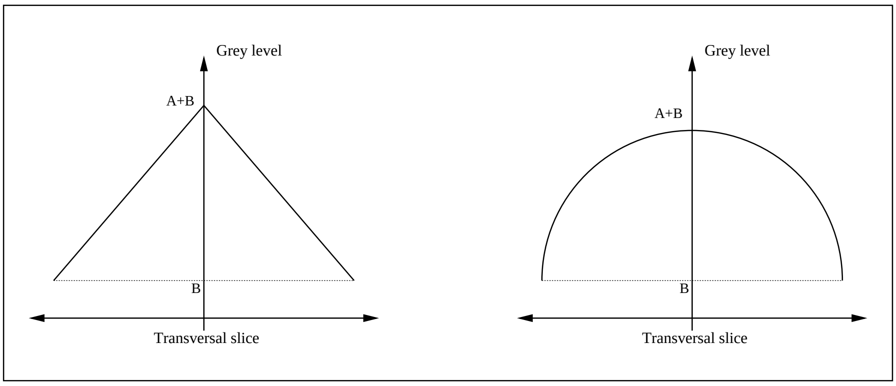

Figure 27: Left : a linear distribution of the mean of the grey values. Right : a spherical distribution of the mean of the grey values.

this pixel and p_x. We have

$$\mu_i^l(p,x) = A\frac{d_p(x)}{r_x} + B$$

where $A < 0$ and $B > 0$ are two constant values (see Fig. (27)). We have now to solve the following problem, when one pixel is inside several objects, which likelihood should we associate to him ? We decide to give him the maximum of the different probabilities calculated for each object $x \in \mathbf{x}$ that contains it. So, $\mathcal{L}_l(\mathcal{I}|\mathbf{x})$ becomes :

$$\mathcal{L}_l(\mathcal{I}|\mathbf{x}) = \prod_{p\in\mathcal{C}_o} \frac{1}{\sqrt{2\pi}\sigma_o} \exp\left(\frac{-(y_p-\mu_o)^2}{2\sigma_o^2}\right) \prod_{p\in\mathcal{C}_i} \max\left\{\mathbb{P}\left(y_p \in \mathcal{N}(\mu_i^l(p,x),\sigma_i^l)\right)\right\}$$

▷ Spherical model

In the spherical model, the mean $\mu_i(p)$ is sinusoidal inside the tree. Let p be a pixel inside one of the object of the configuration $x = (p_x, r_x) \in \mathbf{x}$, and $d_p(x)$ the distance in $\mathcal{P}$ between this pixel and the center of the object p_x. We have :

$$\mu_i^s(p,x) = A\cos\left(\frac{\pi}{2}\frac{d(p,x)}{r_x}\right) + B$$

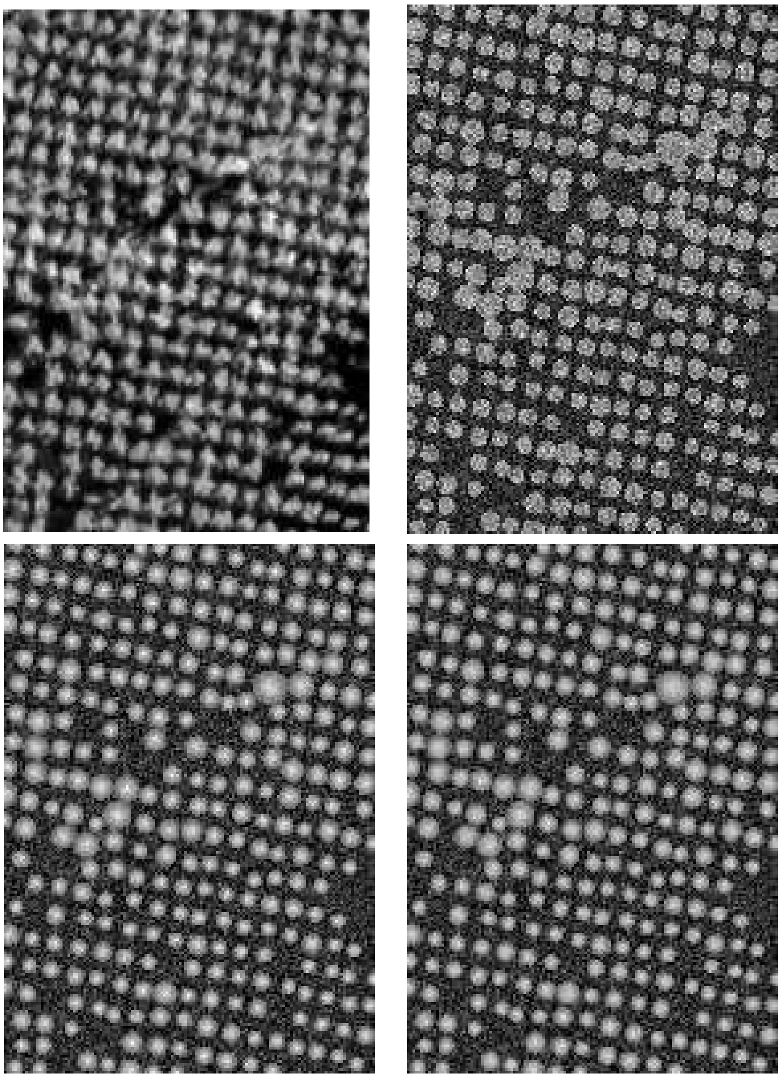

Figure 28: Top left : the original image. Top right : a realization of $\mathcal{I}$ with $\mathcal{L}(\mathcal{I}|\mathbf{x})$. Bottom left : a realization of $\mathcal{I}$ with $\mathcal{L}_l(\mathcal{I}|\mathbf{x})$. Bottom right : a realization of $\mathcal{I}$ with $\mathcal{L}_s(\mathcal{I}|\mathbf{x})$

.

where $A < 0$ and $B > 0$ are two constant (see Fig. (27)). The likelihood $\mathcal{L}_d(\mathcal{I}|\mathbf{x})$ becomes :

$$\mathcal{L}_s(\mathcal{I}|\mathbf{x}) = \prod_{p\in\mathcal{C}_o} \frac{1}{\sqrt{2\pi}\sigma_o} \exp\left(\frac{-(y_p-\mu_o)^2}{2\sigma_o^2}\right) \prod_{p\in\mathcal{C}_i} \max\left\{\mathbb{P}\left(y_p \in \mathcal{N}(\mu_i^s(p,x),\sigma_i^l)\right)\right\}$$

▷ **R**esults

Fig. (28) shows the impact of these two models. We can see that some work has to be done on the modeling of the background which is brighter than the original one. Moreover, in Fig. (29), we can see the difference between the extraction with the former likelihood model and $R_M = 10$ (see Fig. (26)). The spherical model seems more suitable. We have no false alarms anymore, the non-detected objects could be extracted in a second step, or in a single step if more care is taken for the choice of σ_i^l and σ_i^s which were directly set to 20 here.

Statistic	Linear likelihood $\mathcal{L}_l(\mathcal{I}|\mathbf{x})$	Spherical likelihood $\mathcal{L}_s(\mathcal{I}|\mathbf{x})$
Iterations ♯	2000000	2000000
Objects detected	244	284

Table 15: Some statistics with the new likelihood

▷ **N**on Bayesian model

The Bayesian model does not seem to be appropriate for most of the images, because the pixel distribution outside the trees is unknown. Another way to proceed is to introduce, instead of the likelihood $\mathcal{L}(\mathcal{I}|\mathbf{x})$, a data energy term $U_d(\mathbf{x})$ that would be the sum of an attachment energy defined for every object in the configuration $\mathbf{x}$:

$$U_d(\mathbf{x}) = \sum_{x\in\mathbf{x}} U_d(x) \tag{27}$$

Some examples of this approach can be found in [Ort04] or [Lac04] for other applications.

3.4.9 Multi-zone detector

Another limit in our model is that it is designed for extracting trees that come from a single plantation in the image, because of the prior density which takes into account

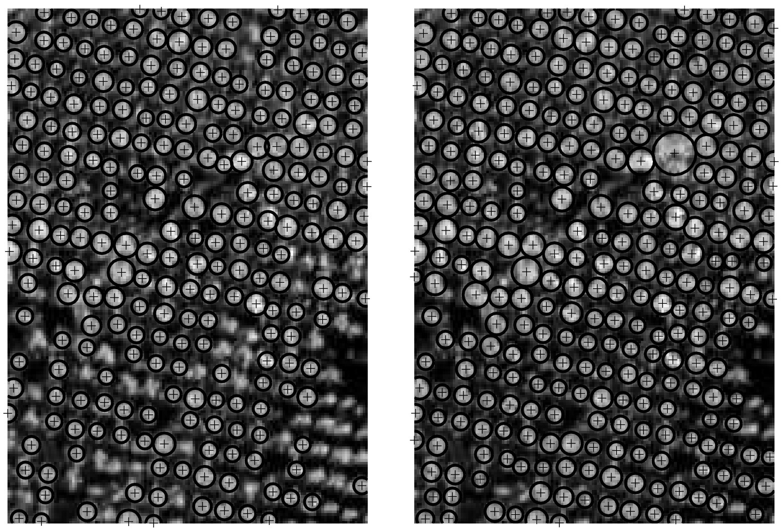

Figure 29: Left : extraction with $R_M = 10$ and the linear model of likelihood. Right : extraction with $R_M = 10$ and the spherical model of likelihood.

one kind of alignment with the four vectors $\pm\overrightarrow{v_i}$. This does not mean that we cannot extract the trees that come from different plantations, this is still possible thanks to the likelihood. However, it is interesting to study the impact of adding one mark to the object space that defines the type of plantation to which an object belongs.

Thus, in the case of an image composed of several plantations of different orientations, we can add a mark to the space $\mathcal{K}$. If n_p is the total number of different plantations we would like to extract in the image, $\mathcal{K}$ becomes :

$$\mathcal{K} = [R_m, R_M] \times [1, n_p] \tag{28}$$

We write $x = (p, k) \in \mathcal{P} \times \mathcal{K}$ where $k = (r, t)$, with t the label of the plantation. The alignment vectors that characterize a plantation of mark t are now written $\pm\overrightarrow{v_i^t}$. A slight change in the definition of the alignment neighbourhood relationship occurs.

Definition

Aligned objects in a multi-plantation image

Two objects $x_1 = (p_1, k_1)$ *and* $x_2 = (p_2, k_2)$ *are aligned, and we write* $x_1 \sim_a x_2$*, if* $t_1 = t_2 = t \in [1, n_p]$ *(same plantation) and if* $p_2 \in \bigcup_{i=\{1,2\}} \mathcal{B}_\epsilon \left(p_1 \pm \overrightarrow{v_i^t}\right)$.

Two objects x_i and x_j can be aligned if and only if they belong to the same plantation. The attractive energy of the prior term $U_p(\mathbf{x})$ is still :

$$U_a(\mathbf{x}) = \gamma_a \sum_{x_i \sim_a x_j} \mathcal{Q}(x_i, x_j) \tag{29}$$

Eventually, we add to the RJMCMC sampler a non-jumping transformation $Q_p(\mathbf{x}, B)$ that proposes to an object $x \in \mathbf{x}$ a swap of plantation. The Green's ratio is the same as the other non-jumping transformations :

$$R_{NJ}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{u\} \cup \{v\}) = \frac{f(\mathbf{y})}{f(\mathbf{x})} \tag{30}$$

We present in the next chapter some results obtained on images simulated with AMAP software that contains several plantations.

4 Ellipse model

In the previous chapter, a marked point process of discs has been proposed in order to extract tree crowns from digital images. In the case of non-disc shaped tree crown extraction, or with more complicated images especially with a specific shadow orientation, this model is quite limited. That is why we propose to extend it by using ellipses instead of discs.

An ellipse could help us to model both the tree crown and its shadow on the ground, which is actually what distinguishes it from the rest of the image. In this chapter, we only tackle the description of this new marked point process and its simulation, and give some results compared with previous ones obtained with the disc process. The shadow is still not taken into account. This will be done in the future.

4.1 Ellipse marked point process and its simulation

In this part, we deliberately skip the details that were previously addressed in the disc process. We only focus on the new aspects.

4.1.1 Description of the marked point process

▷ State space

The state space S is now a bounded set of $\mathbb{R}^5$:

$$S = \mathcal{P} \times \mathcal{K} = [0, X_M] \times [0, Y_M] \times [a_m, a_M] \times [b_m, b_M] \times [0, \pi[\tag{31}$$

where X_M and Y_M are respectively the height and the length of the image $\mathcal{I}$, (a_m, a_M) the minimum and the maximum major axis, (b_m, b_M) the minimum and the maximum minor axis, and $\theta \in [0, \pi[$ the orientation of our objects, as shown in Fig. (1). In practice we set $a_m = b_m$ and $a_M = b_M$. One object is still described by two sets of parameters in $\mathcal{K}$ because $(a, b, \theta) = (b, a, \pi - \theta)$. That is why we set $b > a$ to avoid this redundancy.

In the following, an object $x \in S$ will be noted $x = (p, k)$, where $p \in \mathcal{P}$ is the position of its center and $k = (a, b, \theta) \in \mathcal{K}$ its marks.

▹ **D**ensity of the marked point process

This part is not much affected by the change of geometry of the object. The only difference will occur in the repulsive term of the prior density, for which the overlapping rule has to be generalized.

Definition | ***Overlapping objects***

Two objects x_1 and x_2 are overlapping, and we write $x_1 \sim_r x_2$, if the intersection of their silhouettes is not empty on $\mathcal{P}$.

The overlapping coefficient is generalized too.

Definition | ***Overlapping coefficient***

Let $x_1 = (p_1, k_1)$ and $x_2 = (p_2, k_2)$ be two overlapping objects ($x_1 \sim_r x_2$). The overlapping coefficient, $\mathcal{A}(x_1, x_2) \in [0,1]$, is defined as :

$$\mathcal{A}(x_1, x_2) = \frac{\lambda_{\mathcal{P}}(x_1 \cap x_2)}{\max\left(\lambda_{\mathcal{P}}(x_1), \lambda_{\mathcal{P}}(x_2)\right)}$$

where $\lambda_{\mathcal{P}}(x_1 \cap x_2)$ is the Lebesgue measure on $\mathcal{P}$ of the intersection of the silhouettes.

The prior energy is unchanged, as for the likelihood.

4.1.2 Simulation algorithm

The RJMCMC algorithm is the same as the one previously used in the disc model. In this part, we review the different proposition kernels and tackle the points that are changing in this new model.

▹ **U**niform birth and death

The uniform birth and death kernel still proposes to the current configuration of ellipses $\mathbf{x}$ a birth of an object u uniformly generated in S, or the death of an object

$u \in \mathbf{x}$. The Green's ratios are the same as before :

$$R_{BD}(\mathbf{x}, \mathbf{y} = \mathbf{x} \cup \{u\}) = \frac{p_D}{p_B} \frac{f(\mathbf{y})}{f(\mathbf{x})} \frac{\nu(S)}{n(\mathbf{x}) + 1} \tag{32}$$

and for a death

$$R_{BD}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{u\}) = \frac{p_B}{p_D} \frac{f(\mathbf{y})}{f(\mathbf{x})} \frac{n(\mathbf{x})}{\nu(S)} \tag{33}$$

▷ **N**on jumping transformations

The non-jumping transformations are adapted to ellipses, which means that the dilation now proposes to change the major and the minor axis. Moreover, a rotation is added. This move proposes to change the angle θ of the ellipse. For every non-jumping transformation, the associated Green's ratios are :

$$R_{NJ}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{u\} \cup \{v\}) = \frac{f(\mathbf{y})}{f(\mathbf{x})} \tag{34}$$

▷ **B**irth and Death in a neighbourhood

As the previous kernels this one remains unchanged because the neighbourhood relationship $\sim_a$ is the same. The Green's ratios can be found in (24) and (25).

▷ **S**plit and Merge

The split and merge kernel is slightly modified. First, the definition of two mergeable objects is adapted to the new geometric shape.

Definition | ***Mergeable objects***

Two objects $x_1 = (p_1, k_1)$ and $x_2 = (p_2, k_2)$ are mergeable, and we write $x_1 \sim_m x_2$, if $d(p_1, p_2) \leq a_1 + a_2 + \rho$, where a_i is the major axis of the ellipse x_i.

The merge transformation map $T^M_{x_i \sim_m x_j}(z_m)$, which gives us the new object $y \in S$, is now :

$$y = T^M_{x_i \sim_m x_j}(z_a, z_b) = \left(\frac{p_i + p_j}{2}, \frac{a_i + a_j}{2} + z_a, \frac{b_i + b_j}{2} + z_b, \frac{\theta_i + \theta_j}{2} \right)$$

where z_a is a random variable uniformly distributed in $Z_a = [0, a_M - \frac{a_i+a_j}{2}]$, and z_b the equivalent for the minor axis in $Z_b = [0, b_M - \frac{b_i+b_j}{2}]$. We fix $M_{Z_a} = \lambda(Z_a)$ and $M_{Z_b} = \lambda(Z_b)$. Note that an additional rotation of $\frac{\pi}{2}$ and a swap between a_y and b_y are proposed if $b_y > a_y$ with the choice of random variables.

The split does the opposite, which means that starting from one object x_i in the configuration $\mathbf{x}$, we propose to replace it by two mergeable objects y_1 and y_2 according to $T^S_{x_i}(z_{xy}, z_{a1}, z_{b1}, z_{a2}, z_{b2}, z_\theta)$:

$$\begin{array}{rcl} [y_1, y_2] & = & T^S_{x_i}(z_{xy}, z_{a1}, z_{b1}, z_{a2}, z_{b2}, z_\theta) \\ y_1 & = & (p_i + z_{xy}, a_i - z_{a1} + z_{a2}, b_i - z_{b1} + z_{b2}, \theta_i + z_\theta) \\ y_2 & = & (p_i - z_{xy}, a_i - z_{a1} + z_{a2}, b_i - z_{b1} + z_{b2}, \theta_i - z_\theta) \end{array}$$

where z_{a1} is a random variable uniformly distributed in $Z_{a1} = [0, a_i - a_m]$, and equivalently z_{b1} in $Z_{b1} = [0, b_i - b_m]$. Here again, after the first dilation with z_{a1} and z_{b1}, we could add a rotation of $\frac{\pi}{2}$ if $a_i - z_{a1} < b_i - z_{b1}$. Moreover, z_{a2} and z_{b2} are random variables distributed in a subset Z_{ab} of $Z_{a2} \times Z_{b2}$, where $Z_{a2} = [0, \min(a_i - z_{a1} - a_m, a_M - a_i + z_{a1})]$ and $Z_{b2} = [0, \min(b_i - z_{b1} - b_m, b_M - b_i + z_{b1})]$. Indeed, after this second dilation, we should still respect $a > b$ for both ellipses. Eventually, $z_{xy} = (z_x, z_y)$ is the same as before, and z_{theta} is distributed in $Z_\theta = [-\frac{\pi}{4}, \frac{\pi}{4}]$. We fix $M_{Z_{a1}} = \lambda(Z_{a1})$, $M_{Z_{b1}} = \lambda(Z_{b1})$, $M_{Z_{ab}} = \lambda(Z_{ab})$, $M_{Z_\theta} = \lambda(Z_\theta)$, and $M_{Z_{xy}} = \lambda_{\mathcal{P}}(Z_{xy})$.

We do detail neither the proposition kernels, nor the symmetric measure, which are very close to the ones of the disc process. We just give the Green's ratios of these moves :

$$R_{SM}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{x_i, x_j\} \cup \{y\}) = \frac{n_{\sim_m}(\mathbf{x}) + n(\mathbf{x})}{n_{\sim_m}(\mathbf{y}) + n(\mathbf{y})} \frac{f(\mathbf{y})}{f(\mathbf{x})} \frac{M_{Z_a} M_{Z_b}}{32\beta M_{Z_{a1}} M_{Z_{b1}} M_{Z_{ab}} M_{Z_\theta} M_{Z_{xy}}} \tag{35}$$

and

$$R_{SM}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{x_i\} \cup \{y_1, y_2\}) = \frac{n_{\sim_m}(\mathbf{x}) + n(\mathbf{x})}{n_{\sim_m}(\mathbf{y}) + n(\mathbf{y})} \frac{f(\mathbf{y})}{f(\mathbf{x})} \frac{32\beta M_{Z_{a1}} M_{Z_{b1}} M_{Z_{ab}} M_{Z_\theta} M_{Z_{xy}}}{M_{Z_a} M_{Z_b}} \tag{36}$$

4.2 Results on aerial images

We present in Fig. (31) one result obtained with the ellipse model on an image of 3 plantations simulated with AMAP. This software (see Appendix A.3) gives us the possibility to model our stands and offer a rendering which is very close to natural images. We included in our model an additional mark that represents the label related to the type of plantation for each object. The image is 403 × 410, and we ran the algorithme during 10 millions iterations.

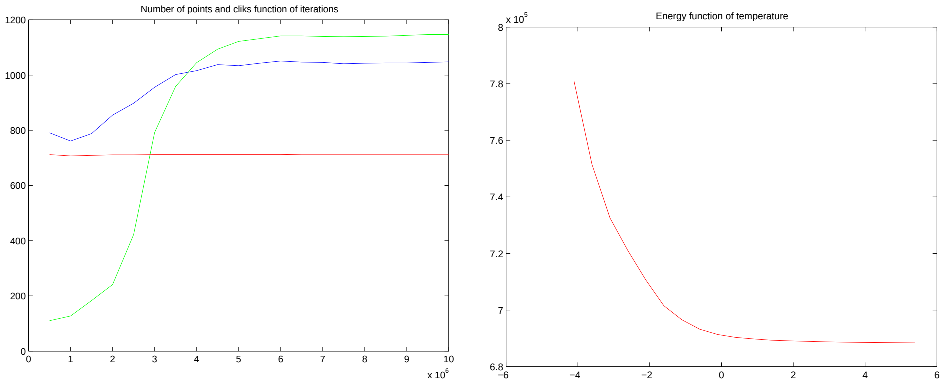

Figure 30: Left : Evolution of the number of objects (red), of mergeable objects (blue) and of aligned pairs of objects (green) during the simulation with respect to the number of iterations. Right : Decrease of the energy with respect to the negative logarithm of the temperature.

From that extraction, the good point is that we have access to many statistics such as the number of trees, the diameter distribution and the orientation of the trees in each plantation. It is then interesting to look at the charts in Fig. (30), which show us the evolution of some statistics during the simulation. We can see that the total number of objects is quite fast to reach, perhaps because the parameter β was well chosen. But obtaining the position and the marks of the trees takes time (see Fig. (30), lefthandside), when we see the number of aligned cliks which gradually increases.

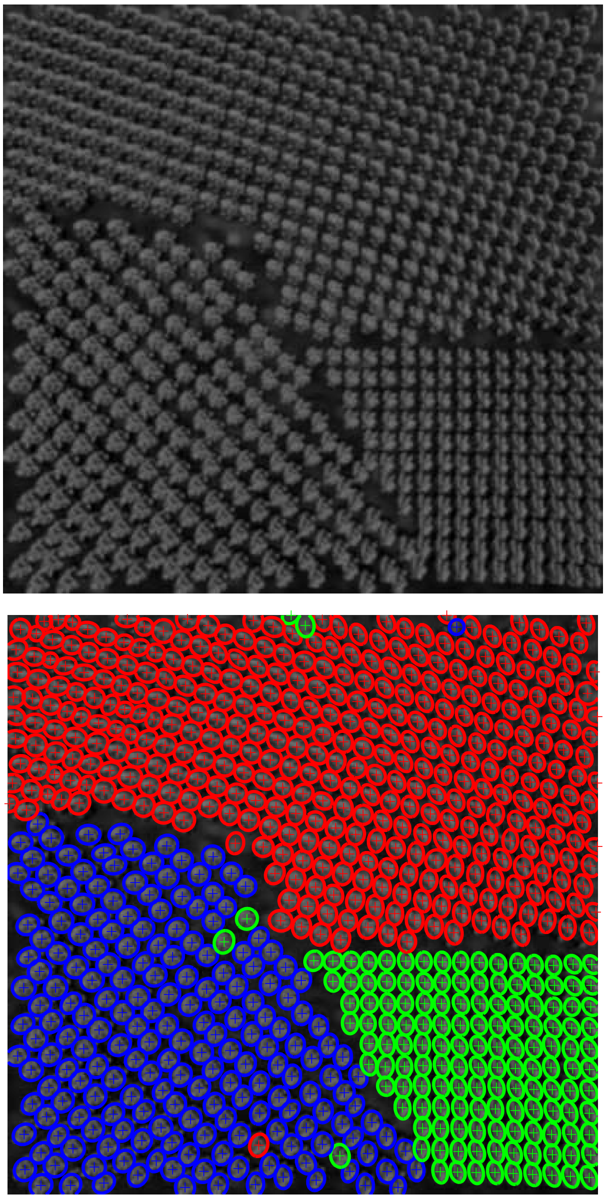

Figure 31: Top : the original AMAP image, with 3 plantations of different difficulty levels (bottom right one at nadir with perfect alignments, top one not at nadir, bottom left with looser alignments). Bottom : the extraction result, each ellipse color stands for the mark of plantation that was found.

5 Conclusion

In this report we present an algorithm to extract tree crowns in high resolution aerial images of forests using a marked point process of discs or ellipses. This approach is automatic, and gives precious statistics like the number of tree crowns and their diameter distribution that are helpful in forestry resource management. The parameters can all be calibrated or estimated via an MCMCML procedure, and the obtained results on images of poplars are quite satisfying.

In the future, we will design more accurate models for the data term of the density, and think about a non Bayesian model in order to tackle more complicated images. Texture informations could be added in order to extract and classify the trees at the same time. Optimization techniques will also be studied to improve the computational time.

6 Acknowledgments

The work of the first author has been partly supported by a grant from the MAS Laboratory of Ecole Centrale Paris. The authors would like to thank the French Forest Inventory (IFN) for providing the data and for interesting discussions. A special thanks to C. Saguez from the MAS Laboratory of ECP, P.H. Cournede and P. De Reffye from Digiplante (INRIA Rocquencourt / CIRAD), M. Jaeger, T. Fourcaud and V. Prinet from LIAMA (Beijing), for their collaboration within the ARC “Mode de Vie”.

A Appendix : convergence of algorithms

A.1 Disc model

We detail here the calculations that were omitted in the chapter addressing the disc model.

A.1.1 Split and Merge kernel

The split kernel proposes to split one selected object of the configuration $\mathbf{x}$ in two objects that are mergeable with respect to $\sim_m$, and the merge kernel proposes to merge two mergeable objects.

$\triangleright$ **P**roposition kernel $Q_{SM}(.,.)$

The global split and merge kernel can be seen as a sum of two proposition kernels, that deal with the merge and with the split :

$$Q_{SM}(\mathbf{x},.) = p_M(\mathbf{x})Q_M(\mathbf{x},.) + p_S(\mathbf{x})Q_S(\mathbf{x},.)$$

We have seen that both proposition kernels could be written as : $\forall B \in \mathcal{B}(\mathbf{\Psi})$,

$$Q_M(\mathbf{x}, B) = \frac{1}{n_{\sim_M}(\mathbf{x})}\frac{1}{M_{Z_m}}\sum_{x_i \sim_m x_j}\int_{Z_m} \mathbf{1}_B(\mathbf{y} = \mathbf{x} \setminus \{x_i, x_j\} \cup \{y\})\lambda(dz_m)$$

and

$$Q_S(\mathbf{x}, B) = \frac{1}{n(\mathbf{x})}\sum_{x_i \in \mathbf{x}}\frac{1}{M_{Z_{xy}}M_{Z_r}M_{Z_s}}\int_{Z_{xy}\times Z_r \times Z_s} \mathbf{1}_B(\mathbf{y} = \mathbf{x}\setminus\{x_i\}\cup\{y_1, y_2\})\lambda^2(dz_{xy})\lambda(dz_r)\lambda(dz_s)$$

So, for any $\mathbf{x} \in \mathbf{\Psi}$, and well chosen $d\mathbf{y} \in \mathcal{B}(\mathbf{\Psi})$ these kernels become :

$$Q_M(\mathbf{x}, d\mathbf{y} = \mathbf{x} \setminus \{x_i, x_j\} \cup \{y\}) = \frac{1}{n_{\sim_M}(\mathbf{x})}\frac{1}{M_{Z_m}}\lambda(dz_m)$$

and

$$Q_S(\mathbf{x}, d\mathbf{y} = \mathbf{x} \setminus \{x_i\} \cup \{y_1, y_2\}) = \frac{1}{n(\mathbf{x})}\frac{1}{M_{Z_{xy}}M_{Z_r}M_{Z_s}}\lambda^2(dz_{xy})\lambda(dz_r)\lambda(dz_s)$$

We should now find a symmetric measure that dominates $\mathcal{P}_X(d\mathbf{x})Q_{SM}(\mathbf{x}, d\mathbf{y})$.

▷ **Symmetric measure** $\xi_{SM}(.,.)$

The symmetric function $\xi_{SM}(A,B)$ on $\mathcal{B}(\mathbf{\Psi}) \times \mathcal{B}(\mathbf{\Psi})$ should always give an empty measure to any Borel sets $(A,B) \in \mathcal{B}(\mathbf{\Psi}_j) \times \mathcal{B}(\mathbf{\Psi}_k)$ if $j \neq k+1$ or $j+1 \neq k$. Indeed, it has to dominate $\mathcal{P}_X(d\mathbf{x})Q_{SM}(A,B)$, and $Q_{SM}(A,B) = 0$ in the opposite case. We can also write :

$$\xi_{SM}(A,B) = \sum_n \xi_{SM}(A_{n+1},B_n) + \xi_{SM}(A_n,B_{n+1})$$

where $A_n = A \cap \mathcal{B}(\mathbf{\Psi}_n)$, and $B_n = B \cap \mathcal{B}(\mathbf{\Psi}_n)$. Then, we propose :

$$\xi_{SM}(A_{n+1},B_n) = \int_{\mathbf{\Psi}} \mathbf{1}_{A_{n+1}}(\mathbf{x}) \int_{\mathbb{R}} \sum_{i,j} \mathbf{1}_{B_n} \left(\mathbf{x} \setminus \{x_i, x_j\} \cup \{T^M_{x_i \sim_m x_j}(z_m)\}\right) \lambda(dz_m)\mu(d\mathbf{x})$$

and

$$\xi_{SM}(A_n,B_{n+1}) = \int_{\mathbf{\Psi}} \mathbf{1}_{A_n}(\mathbf{x}) \int_{\mathbb{R}^4} \sum_{i} \mathbf{1}_{B_{n+1}} \left(\mathbf{x} \setminus \{x_i\} \cup \{T^S_{x_i}(z_{xy}, z_r, z_s)\}\right) 8\beta\lambda^2(dz_{xy})\lambda(dz_r)\lambda(dz_s)\mu(d\mathbf{x})$$

The functions $T^M_{x_i \sim_m x_j}(z_m)$ and $T^S_{x_i}(z_{xy}, z_r, z_s)$ are the ones that were used in the split and merge description before. This measure can be easily proven to be symmetric, ie $\xi_{SM}(A_{n+1},B_n) = \xi_{SM}(B_n,A_{n+1})$. Actually, the coefficient 8β appears in the change of variables when we want to prove the symmetry.

The reduced measure in the case of a merge between $(\mathbf{x},\mathbf{y}) \in \mathbf{\Psi} \times \mathbf{\Psi}$ is also :

$$\xi_{SM}(d\mathbf{x}, d\mathbf{y} = \mathbf{x} \setminus \{x_i, x_j\} \cup \{y\}) = \mu(d\mathbf{x})\lambda(dz_m)$$

and for the split :

$$\xi_{SM}(d\mathbf{x}, d\mathbf{y} = \mathbf{x} \setminus \{x_i\} \cup \{y_1, y_2\}) = 8\beta\mu(d\mathbf{x})\lambda^2(dz_{xy})\lambda(dz_r)\lambda(dz_s)$$

▷ **Green's ratios of the split and merge kernel**

Eventually, we can calculate the Radon Nikodym derivatives. In the case of a merge, we have :

$$\begin{aligned}
f_{SM}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{x_i, x_j\} \cup \{y\}) &= \frac{\mathcal{P}_X(d\mathbf{x}) Q_M(\mathbf{x}, d\mathbf{y})}{\xi_{SM}(d\mathbf{x}, d\mathbf{y})} \\
&= \frac{f(\mathbf{x})\mu(d\mathbf{x}) \frac{\lambda(dz_m)}{n_{\sim_M}(\mathbf{x}) M_{Z_m}}}{\mu(d\mathbf{x})\lambda(dz_m)} \\
&= \frac{f(\mathbf{x})}{n_{\sim_M}(\mathbf{x}) M_{Z_m}}
\end{aligned}$$

and for the split :

$$\begin{aligned}
f_{SM}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{x_i\} \cup \{y_1, y_2\}) &= \frac{\mathcal{P}_X(d\mathbf{x}) Q_S(\mathbf{x}, d\mathbf{y})}{\xi_{SM}(d\mathbf{x}, d\mathbf{y})} \\
&= \frac{f(\mathbf{x})\mu(d\mathbf{x}) \frac{\lambda(dz_{xy})\lambda(dz_r)\lambda(dz_s)}{n(\mathbf{x}) M_{Z_{xy}} M_{Z_r} M_{Z_s}}}{8\beta\mu(d\mathbf{x})\lambda^2(dz_{xy})\lambda(dz_r)\lambda(dz_s)} \\
&= \frac{f(\mathbf{x})}{8\beta n(\mathbf{x}) M_{Z_{xy}} M_{Z_r} M_{Z_s}}
\end{aligned}$$

The Green's ratios presented in (35) and (36) can be deduced from these two derivatives.

A.1.2 Birth and Death in a neighbourhood

The birth and death in a neighbourhood kernel proposes to add or to remove some objects in the alignment neighbourhood of existing objects of the configuration $\mathbf{x}$.

▷ **P**roposition kernel $Q_{BDN}(.,.)$

The global birth and death in a neighbourhood kernel can be seen as a sum of eight sub-kernels that propose up to four births and up to four deaths in the neighbourhood of an object of the configuration :

$$Q_{BDN}(\mathbf{x}, B) = \frac{1}{n(\mathbf{x})} \sum_{x \in \mathbf{x}} \sum_{i=1}^{4} p_{B_i}(x, \mathbf{x}) Q_{B_i}(x, \mathbf{x}, B) + \sum_{i=1}^{4} p_{D_i}(x, \mathbf{x}) Q_{D_i}(x, \mathbf{x}, B)$$

We have seen that these 8 proposition kernels could be written as : $\forall B \in \mathcal{B}(\mathbf{\Psi})$,

$$\begin{aligned} Q_{B_i}(x, \mathbf{x}, B) \quad = \quad & \frac{1}{C^i_{n_f(x,\mathbf{x})}} \sum_{\mathcal{J} \in \mathcal{C}(i, n_f(x,\mathbf{x}))} \frac{1}{M_{k_1} \dots M_{k_i}} \int_{\mathcal{B}_{k_1} \times \mathcal{K} \times \dots \times \mathcal{B}_{k_i} \times \mathcal{K}} \mathbf{1}_B(\mathbf{y} = \mathbf{x} \cup \{y_{j_1}, \dots, y_{j_i}\}) \\ & \lambda(dz_{x_{k_1}}) \lambda(dz_{y_{k_1}}) \lambda(dz_{r_{k_1}}) \dots \lambda(dz_{x_{k_i}}) \lambda(dz_{y_{k_i}}) \lambda(dz_{r_{k_i}}) \end{aligned}$$

and

$$Q_{D_i}(x, \mathbf{x}, B) = \frac{1}{C^i_{n_u(x,\mathbf{x})}} \sum_{\mathcal{J} \in \mathcal{C}(i, n_u(x,\mathbf{x}))} \mathbf{1}_B(\mathbf{y} = \mathbf{x} \setminus \{y_{k_1}, \dots, y_{k_i}\})$$

So, for any $\mathbf{x} \in \mathbf{\Psi}$ and appropriate $d\mathbf{y} \in \mathcal{B}(\mathbf{\Psi})$, they become :

$$\begin{aligned} Q_{B_i}(x, \mathbf{x}, d\mathbf{y} = \mathbf{x} \cup \{y_{k_1}, \dots, y_{k_i}\}) \quad = \quad & \frac{1}{C^i_{n_f(x,\mathbf{x})}} \frac{1}{M_{k_1} \dots M_{k_i}} \\ & \lambda(dz_{x_{k_1}}) \lambda(dz_{y_{k_1}}) \lambda(dz_{r_{k_1}}) \dots \lambda(dz_{x_{k_i}}) \lambda(dz_{y_{k_i}}) \lambda(dz_{r_{k_i}}) \end{aligned}$$

in the case of a number i of births, and

$$Q_{D_i}(x, \mathbf{x}, d\mathbf{y} = \mathbf{x} \setminus \{x_{k_1}, \dots, x_{k_i}\}) = \frac{1}{C^i_{n_u(x,\mathbf{x})}}$$

in the case of a number i of deaths.

▷ **S**ymmetric measure $\xi_{BDN}(.,.)$

We should find a symmetric measure which gives an empty measure to any Borel sets $(A, B) \in \mathcal{B}(\mathbf{\Psi}_j) \times \mathcal{B}(\mathbf{\Psi}_k)$ that do not verify $j = k \pm i$, with $i = \{1, \dots, 4\}$. We can also write :

$$\xi_{BDN}(A, B) = \sum_n \sum_{i=\{1,\dots,4\}} \xi_{BDN}(A_{n+i}, B_n) + \xi_{BDN}(A_n, B_{n+i})$$

Then, we propose :

$$\xi_{BDN}(A_n, B_{n+i}) = \int_{\mathbf{\Psi}} \int_{\{u_1,\dots,u_i\} \in S} \mathbf{1}_{A_n}(\mathbf{x}) \mathbf{1}_{B_{n+i}}(\mathbf{x} \cup \{u\}) \beta^i \lambda(du_1) \dots \lambda(du_i) \mu(d\mathbf{x})$$

and

$$\xi_{BDN}(A_{n+i}, B_n) = \int_{\mathbf{\Psi}} \mathbf{1}_{A_{n+i}}(\mathbf{x}) \sum_{\{x_1,\dots,x_i\} \in \mathbf{x}} \mathbf{1}_{B_n}(\mathbf{x} \setminus \{x_1, \dots, x_i\}) \mu(d\mathbf{x})$$

The reduced measure in the case of a birth in a neighbourhood between $(\mathbf{x}, \mathbf{y}) \in \mathbf{\Psi} \times \mathbf{\Psi}$ is also :

$$\xi_{BDN}(d\mathbf{x}, d\mathbf{y} = \mathbf{x} \cup \{y_{k_1}, \dots, y_{k_i}\}) = \beta^i \lambda(du_1) \dots \lambda(du_i) \mu(d\mathbf{x})$$

and for the death in a neighbourhood :

$$\xi_{BDN}(d\mathbf{x}, d\mathbf{y} = \mathbf{x} \setminus \{x_{k_1}, \dots, x_{k_i}) = \mu(d\mathbf{x})$$

▷ **Green's ratios of the birth and death in a neighbourhood kernel**

Eventually, we can calculate the Radon Nikodym derivatives. In the case of a birth of i objects :

$$\begin{aligned} f_{BDN}(\mathbf{x}, \mathbf{y} = \mathbf{x} \cup \{y_{k_1}, \dots, y_{k_i}\}) &= \frac{\mathcal{P}_X(d\mathbf{x}) Q_{BDN}(\mathbf{x}, d\mathbf{y})}{\xi_{BDN}(d\mathbf{x}, d\mathbf{y})} \\ &= \frac{f(\mathbf{x}) \mu(d\mathbf{x}) \frac{1}{n(\mathbf{x})} \sum_{x \in \mathcal{G}(\mathbf{x} \to \mathbf{y})} \dots \lambda(dy_{k_1}) \dots \lambda(dy_{k_i})}{\beta^i \lambda(dy_{k_1}) \dots \lambda(dy_{k_i}) \mu(d\mathbf{x})} \\ &= \frac{f(\mathbf{x})}{n(\mathbf{x})} \frac{1}{\beta^i} \sum_{x \in \mathcal{G}(\mathbf{x} \to \mathbf{y})} \frac{1}{n_{fu}(x, \mathbf{x})} \frac{1}{C^i_{n_f(x, \mathbf{x})}} \frac{1}{M_{k_1} \dots M_{k_i}} \end{aligned}$$

and for the death of i objects :

$$\begin{aligned} f_{BDN}(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{x_{k_1}, \dots, x_{k_i}) &= \frac{\mathcal{P}_X(d\mathbf{x}) Q_{BDN}(\mathbf{x}, d\mathbf{y})}{\xi_{BDN}(d\mathbf{x}, d\mathbf{y})} \\ &= \frac{f(\mathbf{x}) \mu(d\mathbf{x}) \frac{1}{n(\mathbf{x})} \sum_{x \in \mathcal{G}(\mathbf{y} \to \mathbf{x})} \frac{1}{n_{fu}(x, \mathbf{y})} \frac{1}{C^i_{n_u(x, \mathbf{y})}}}{\mu(d\mathbf{x})} \\ &= \frac{f(\mathbf{x})}{n(\mathbf{x})} \sum_{x \in \mathcal{G}(\mathbf{y} \to \mathbf{x})} \frac{1}{n_{fu}(x, \mathbf{y})} \frac{1}{C^i_{n_u(x, \mathbf{y})}} \end{aligned}$$

We then obtain the Green's ratios (24) and (25).

A.1.3 Convergence of the Markov Chain

We need to prove that the Markov chain X_t ergodically converges to the target distribution $\mathcal{P}_X(.)$ in the RJMCMC algorithm, with our choice of proposition kernels.

▷ **R**eversibility

The reversibility is obtained thanks to the choice of the acceptance ratio at each step (see [GM98]).

▷ **I**rreducibility, small sets

The irreducibility and small sets properties is a consequence of the stability condition (4).

Proposition | ***Irreducibility and small sets***

If the unnormalized density $f(\mathbf{x})$ satisfies the condition (4), then the RJMCMC algorithm with our choice of proposition kernels simulates an irreducible Markov chain and every bounded set is small.

Demonstration :
First, we note $\nu_0(.)$ the measure that is empty for every set of $\mathbf{\Psi}$, except $\mathbf{\Psi}_0 = \{\emptyset\}$. The probability to accept a death in the uniform birth and death kernel is :

$$\alpha_D(\mathbf{x}, \mathbf{y} = \mathbf{x} \setminus \{u\}) \geq \frac{p_B}{p_D} \frac{1}{M\nu(S)} \tag{37}$$

where $M > 0$ is the coefficient introduced in (4), chosen large enough so that the righthandside of (37) is less than one. As the probability to propose such a death is $p_{BD} p_D$, we can express the probability $\mathcal{P}_X^m(\mathbf{x}, \mathbf{\Psi}_0)$ to go from a configuration $\mathbf{x}$ to the set space $\mathbf{\Psi}_0$, in $m \geq n(\mathbf{x})$ steps :

$$\begin{aligned} \mathcal{P}_X^m(\mathbf{x}, \mathbf{\Psi}_0) &\geq \mathcal{P}_X^{n(\mathbf{x})}(\mathbf{x}, \mathbf{\Psi}_0) \mathcal{P}_X^{m-n(\mathbf{x})}(\{\emptyset\}, \mathbf{\Psi}_0) \\ &\geq \left(\frac{p_B}{p_D} \frac{1}{M\nu(S)} \right)^{n(\mathbf{x})} (p_{BD} p_D)^{m-n(\mathbf{x})} \\ &\geq \left(\frac{p_{BD} p_B}{M\nu(S)} \right)^m . \end{aligned}$$

This calculation shows that the Markov chain is ν_0-irreducible, since the only ν_0-positive set is $\mathbf{\Psi}_0 = \{\emptyset\}$, and $\mathcal{P}_X^m(\mathbf{x}, \mathbf{\Psi}_0) > 0$ whenever $m \geq n(\mathbf{x})$.

Then, let $\mathcal{C}_m = \{\mathbf{x} \in \mathbf{\Psi} : n(\mathbf{x}) \leq m\}$, *and* $c = \left(\frac{p_{BD} p_B}{M\nu(S)}\right)^m$. *For any* m, $\mathcal{C}_m$ *is small with respect to the measure* $c\nu_0(.)$ *because* $\nu_0(\mathbf{\Psi}_0) = 1$ *and so :*

$$\mathcal{P}_X^m(\mathbf{x}, \mathbf{\Psi}_0) \geq c\nu_0(\mathbf{\Psi}_0)\,, \forall \mathbf{x} \in \mathcal{C}_m.$$

Thus, every bounded set is small.

▹ **H**arris recurrence and geometric ergodicity

The Harris recurrence and the geometric ergodicity of the Markov chain are a consequence of the stability condition (4) too.

Proposition

Harris recurrence and geometric ergodicity

If the unnormalized density $f(\mathbf{x})$ *satisfies the condition (4), then the RJMCMC algorithm with our choice of proposition kernels simulates a Markov chain that is Harris recurrent and geometrically ergodic.*

Demonstration :
This demonstration is longer, but could be found in [PDZ03] or [Ort04]. Moreover, as we work on a space of configurations with a finite number of objects (because of the hard core term), this one is a lot easier.

A.2 Ellipse model

The demonstrations of the results obtained for the Green's ratios and for the convergence of the model are very close to the ones of the disc model. That's why we have chosen not to present them in this report.

A.3 AMAP

We used the AMAP Digiplante's Orchestra software in order to simulate tree stands. This software exploits the CIRAD's knowledge in vegetation and plant growth simulation. It integrates the vegetation procedural growth engine which is able to generate a 3D model of a plant at any age, taking into account its natural morphology. Moreover, this growth engine easily enables to simulate several individuals belonging to the same specy.

We also simulated some plantations of poplars using this method, which made us control the number of trees, their position in the image, ie. the ground truth. This was very useful for the calibration of our model as well as for the discussion of the results.

References

[And03] H.E. Andersen. *Estimation of critical forest structure metrics through the spatial analysis of airborne laser scanner data.* PhD thesis, College of Forestry Resources, University of Washington, Seattle, 2003.

[DvLSZ01] X. Descombes, M.N.M. van Lieshout, R. Stoica, and J. Zerubia. Parameter estimation by a Markov chain Monte Carlo technique for the Candy model. In *Proc. of the IEEE Workshop of Statistical Signal Processing*, pages 22–25, Singapore, August 2001.

[GM94] C.J. Geyer and J. Moller. Simulation and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics*, 21:359–373, 1994.

[GM98] C.J. Geyer and J. Moller. *Stochastic geometry, likelihood and computation*, chapter Likelihood inference for spatial point processes. Chapmann and Hall, London, 1998.

[Gou98] F.A. Gougeon. Automatic individual tree crown delineation using a valley-following algorithm and rule-based system. In D.A. Hill and D.G. Leckie, editors, *Proc. of the International Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, pages 11–23, Victoria, British Columbia, Canada, February 1998.

[Gre95] P.J. Green. Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. *Biometrika 82*, pages 711–7320, 1995.

[GT92] C.J. Geyer and E.A. Thompson. Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society*, 54:657–699, 1992.

[Lac04] C. Lacoste. *Extraction de réseaux linéiques à partir d'images satellitaires et aériennes par processus ponctuels marqués.* PhD thesis, University of Nice-Sophia Antipolis, December 2004. (in French).

[Lar99] M. Larsen. Individual tree top position estimation by template voting. In *Proc. of the Fourth International Airborne Remote Sensing Conference and Exhibition / 21^{st} Canadian Symposium on Remote Sensing*, volume 2, pages 83–90, Ottawa, Ontario, June 1999.

[LBN+99] D.G. Leckie, C. Burnett, T. Nelson, C. Jay, N. Walsworth, F.A. Gougeon, and E. Cloney. Forest parameter extraction through computer-based analysis of high resolution imagery. In *Proc. of the Fourth International Airborne Remote Sensing Conference and Exhibition / 21st Canadian Symposium on Remote Sensing*, Ottawa, Ontario, June 1999.

[ODZ03a] M. Ortner, X. Descombes, and J. Zerubia. Automatic 3D land register extraction from altimetric data in dense urban areas. Research Report 4919, INRIA, September 2003.

[ODZ03b] M. Ortner, X. Descombes, and J. Zerubia. Improved RJMCMC point process sampler for object detection on images by simulated annealing. Research Report 4900, INRIA, August 2003.

[Ort04] M. Ortner. *Processus ponctuels marqués pour l'extraction automatique de caricatures de bâtiments à partir de modèles numériques d'élévation.* PhD thesis, University of Nice-Sophia Antipolis, December 2004. (in French).

[PDZ03] G. Perrin, X. Descombes, and J. Zerubia. Extraction de houppiers par processus objet. Research Report 5037, INRIA, December 2003.

[PS00] A. Penttinen and D. Stoyan. Recent applications of point process methods in forestry statistics. *Statistical Science*, 15(1):61–78, 2000.

[SGM04] R. Stoica, P. Gregori, and J. Mateu. Simulated annealing and object point processes : tools for analysis of spatial patterns. Technical Report 69, University Jaume I, Castellon, Spain, 2004.

[SKM95] D. Stoyan, W.S. Kendall, and J. Mecke. *Stochastic Geometry and its Applications.* Wiley, 1995.

[Str75] D.J. Strauss. A model for clustering. *Biometrika*, 63:467–475, 1975.

[vL00] M.N.M. van Lieshout. *Markov point processes and their applications.* Imperial College Press, London, 2000.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399