INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Building Extraction from Digital Elevation Model

Mathias Ortner — Xavier Descombes — Josiane Zerubia

N° 4517

Juillet 2002

THÈME 3

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4517--FR+ENG

Building Extraction from Digital Elevation Model

Mathias Ortner , Xavier Descombes , Josiane Zerubia

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Ariana

Rapport de recherche n° 4517 — Juillet 2002 — 48 pages

Abstract: We aim to extract buildings from Digital Elevation Models.

To achieve this goal, we define a point process whose points represent buildings. We then define a density for this point process which is split into two parts. When written as an energy this density consists of two fields :

- the first one is an "internal field" that allows us to model the prior knowledge we have on patterns of buildings in urban areas. For instance, we avoid overlapping buildings.
- the second one is an "external field" that makes the point process fit the data, ie. the Digital Elevation Model.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Once we have defined this artificial likehood, we use a Metropolis Hastings Green sampler, which is an extension of Geyer and Møller algorithm to sample point processes. This gives an estimate of the observed urban area.

We present results on real data provided by the French Mapping Institute (IGN).

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

Extraction de Bâtiments sur des Modèles Numériques d'Elévation

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

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

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

Nous calculons ensuite une estimée de la zone urbaine à partir de cette densité en utilisant une simulation de Monte Carlo par Chaine de Markov et, en particulier, un algorithme de Metropolis Hastings Green, qui est une extension de l'algorithme de simulation de processus ponctuels proposé par Geyer et Møller.

Nous proposons des résultats sur des données réelles fournies par l'IGN.

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

Contents

1 Introduction

Detecting buildings from aerial images and automatic reconstruction of urban scenes have become of deep interest in many applications : cartography, flight simulations, etc...
The third dimension is now of first importance. A lot of people need to deal with 3D maps of towns. For this reason, automatic 3D analysis of urban areas is now an important issue.

However, high density of urban areas and complexity of human made objects make it difficult to achieve, and automatic 3D urban area cartography is still an open problem.
Some work has been done on this kind of task (see [3, 13] for general overviews) :

- A lot of works focus on buildings and especially on roof modeling (see [3] to get a good overview of models and methods). Usually, these works are restrained to sparse areas, with just a few buildings, and the main goal is to describe complex buildings.
- The other half of the works focus on building detection, that means working on dense areas and trying to catch all buildings.

Herein, we are interested by this second objective.

Several methods and ideas have been proposed since a couple of years.

- A lot of them are based on primitive detection on two or more views (see [21] to have an overview of this kind of techniques). They rely mostly on line or corner detections and hypothesis testing.
- Other works rely on DEM [1] construction using two or more views given by aerial images (see [4, 5, 9], for instance). The main idea is to detect buildings and to construct a DEM of the scene simultaneously. This is mixed with steps that use geometric and radiometric information to distinguish ground from buildings and buildings from vegetation. This

[1]DEM : Digital Elevation Model. It is an image of the altimetry of an area, including ground and above-ground objects.

kind of methods seems to be very efficient, but do not give any object description of a scene.

- Some tentatives have been made by using only DEM (see [25] for instance, or [11]). The main problems encountered here come from high density of buildings or vegetation.

1.1 Object recognition

We are interested in an object approach because we want to obtain a vector representation of a dense urban area. A few works have been developed within this framework. People working on dense urban areas are mainly interested in segmentation and classification.

An object representation gives a vector data, which is better than a raster data for compression. Moreover, an object representation is close to a semantic interpretation of the scene.

In [6], Garcin et al. use a point process approach to construct an object representation of an urban area.

This kind of framework has been used in image processing by Van Lieshout and Baddeley in [1] to detect an unknown number of objects. Rue and Hurn, in [16] have also used this approach with more complex objects.

Point processes allow to have an object-oriented approach : points of a point process can be buildings, and it is possible to add interactions between buildings. Thus, it is possible to model the prior knowledge we have on the behavior of buildings in urban areas, since buildings are seen in the point process context as particles.

1.2 Digital Elevation Model

1.2.1 Description

A DEM is a raster-data. Value at each location (pixel) represents a height. A DEM is thus a description of the altimetry of an area. The difference with a Surface Model comes from the kind of area described and the resolution used.

With low resolution, surface models describe natural topologies like mountains or valleys. Elevation models are focused on areas consisting of man-made objects and describe their height. Hence, they need to use high resolution.

1.2.2 Construction and artifacts

Usually, DEMs are built from aerial images. The data used in this work has been provided by French Mapping Institute (IGN).

To built a DEM, one requires two (or more) images of a scene. Knowing the position of the cameras that have taken these images, it is possible to built an image of the height of the scene, as does the brain with the two images taken by the two eyes to built a 3D representation of the environment.

Figure 1 a) shows two images of the French town of Amiens. With images of this type, the French Mapping Institute (IGN) has built the DEM shown on Figure 1 b). On this DEM, white pixels represent high points, while black ones stand for low points.

The DEM is of medium resolution (20 cm by 20 cm in the horizontal directions, and 15 cm vertically). The noise is not a geometrical noise due to the algorithms used to construct the DEM.

Other problems come from the density of the urban area. Some buildings are not visible on one of the images due to occlusions. Thus, for this kind of buildings, it is impossible to construct a 3D representation since some information is missing.

1.2.3 Goals and Interest

Our first goal is to refine the DEM. We want to eliminate geometrical noise. To achieve this, we have to use geometrical constraints.

As mentioned previously, we also want to have an object description of the scene. This is useful for semantic interpretation and for compression (vector data is better than raster data).

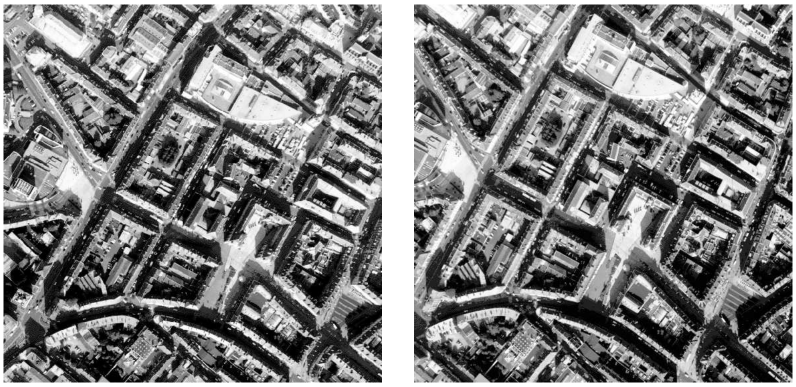
We also want to deal with occlusions. So we need to use a prior knowledge on urban areas in order to complete the missing information. Stochastic models fit well this objective, since a Bayesian framework is made of prior models and data likelihood.

These three points have led us to use point processes to refine DEMs. They actually give us a stochastic framework combined with geometrical objects.

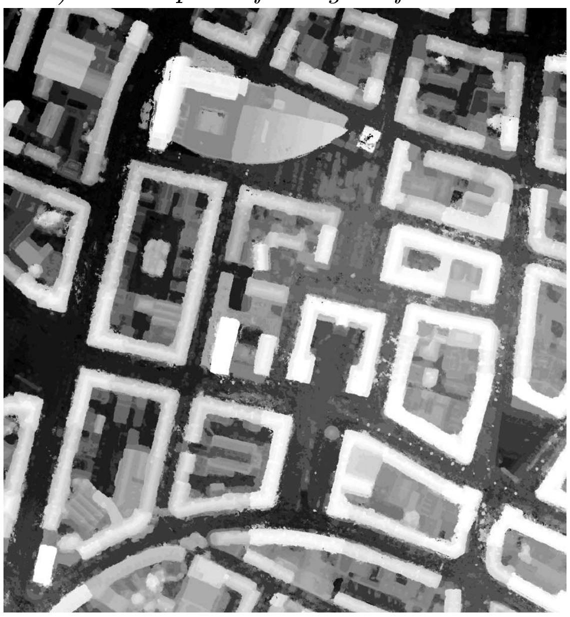
These geometrical objects can be silhouettes of buildings. The first advantage of using a DEM to extract buildings comes from the following : silhouettes are geometrical objects living in a plane, while buildings are living in a 3D space. Thus, if we first detect silhouettes of buildings and then estimate their roofs, we can work in a smaller space.

Of course, DEMs contain less information than radiometric images which are used in some other works on the topic. However, DEMs differ only with respect to resolution and noise. By using DEM, We avoid problems sewn while using radiometric information from images, since we do not need a lightening model of the scene. That is the second main advantage of using DEM.

Finally, it is worth to point out that refining DEMs is also of great interest in itself, since crude DEMs can be obtained not only by using stereo vision on radiometric images, but also by SAR interferometry or LASER measurement.



a) Stereo pair of images of Amiens.



b) Digital Elevation Model of Amiens (20 cm of resolution). Pixels in white are high points, while dark pixels are low points.

Figure 1: Aerial images and Digital Elevation Model provided by the French Mapping Institute (IGN).

2 A model for urban areas

As previously mentioned, the aim of this work is to extract 3D maps of urban areas from Digital Elevation Models.

We have seen that the marked point process framework fit this application well. This framework provides a way of modeling urban areas. But first of all, a model of building is needed. Of course such a model can not be independent of the data or of the algorithm used to complete the processing.

2.1 Building model

There are several way of modeling and detecting buildings.
We are interested herein in dealing with dense urban areas. Furthermore, we try to have an object based approach. As we focus on extraction and not on description, we choose a simple model of building. Figure 2 shows one of these types of model. This is the model you can find in [6].
Using this kind of model, the user needs to pay attention to two points :

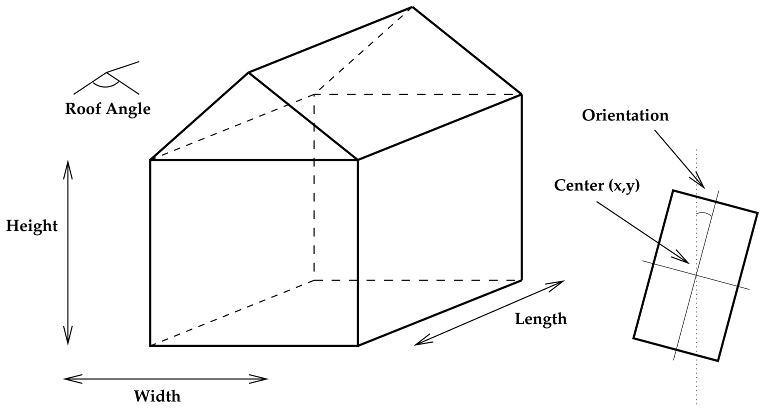


Figure 2: Example of a model of a building and its parameters.

- A cost function measuring how far a structure picked up in the data is from the model. Using radiometric information, the lightening model is involved here.

- An algorithm that allows to find parameters optimizing cost function for each building.

The difficulty comes from the fact that this model is a 3D model. First, the cost function has to compare 3D proposed objects to the associated data, using DEMs, which means comparing the volume of the proposed building to the volume of the data where the building stands. It can be very long from a computational point of view.
Moreover, the algorithm needs to optimize the cost function on a big space. The previously presented 3D model relies on 7 real parameters.

Because of these drawbacks, we choose another approach. We distinguish 2D silhouette detection and 3D shape estimation. It is worth to point out that this is easy only because our data is a DEM.
Our model consists in :

- first, a geometrical model of the silhouette of a building,

- secondly, a cost function measuring if proposing a building (ie a silhouette) somewhere on the data is relevant or not,

- third, a roof estimation function giving an estimation of the roof of a building, when its silhouette and the data where this silhouette stands are known.

Figure 3 explains how this approach works.

- Fig 3 a shows some propositions of a silhouette on the DEM,

- Fig 3 b presents the associated reward, given by the cost function. This cost function should be almost convex with respect to the silhouette parameters, since an optimization has to be done,
- Fig 3 c gives an example of possible estimated roofs. The first column should not lead to any roof estimation since the cost of the proposed silhouette is to high (ie. the reward is too low).

2.1.1 Silhouette

We choose to model the silhouette of buildings by rectangles. Obviously, rectangles can be described by elements of a 5 dimensional space, using a center, a length, a width and an orientation. The space is denoted by S :

$$S = [0, X_{max}] \times [0, Y_{max}] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times [L_{min}\ ,\ L_{max}] \times [l_{min}\ ,\ l_{max}]$$

More complex models could be used, like polygonal ones.

2.1.2 Cost function

The cost function is a mapping from the space of rectangles to $\mathbb{R}$.

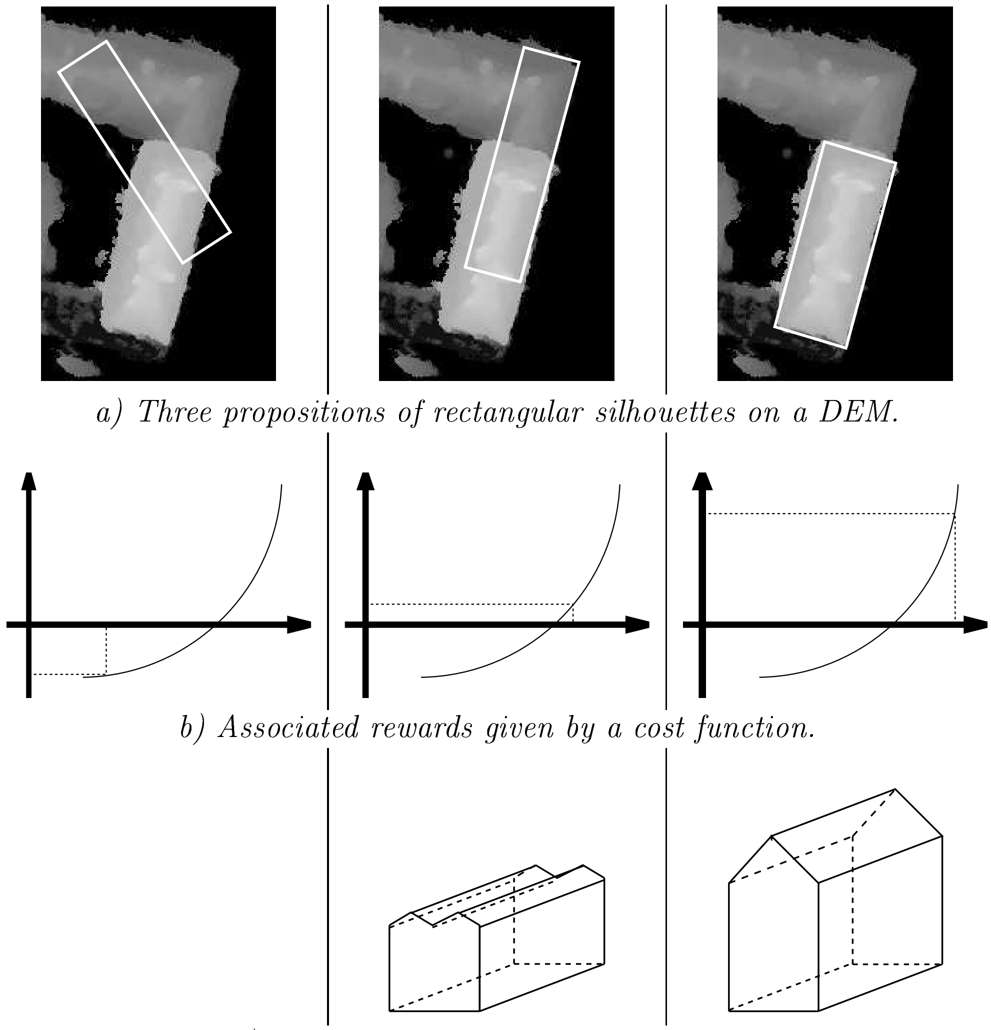
Assuming the Digital Elevation Model is defined on $S = [0, X_{max}] \times [0, Y_{max}] \subset \mathbb{R}^2$ we note h the mapping from points of S to $\mathbb{R}^+$ describing the DEM. For a point p, $h(p)$ is the the height in meters given by the DEM.

Given a set $s = \{x_1, \ldots, x_n\}$ of points of S, we note $\bar{s}$ its mean, that is :

$$\bar{s} = \frac{\sum_{i=1}^{i=n} h(x_i)}{n}$$

Having a rectangle, we build a mask of points as shown on Figure 4. This mask of points is composed of 5 different areas :

- four lines around the rectangle $(g_1, \ldots, g_4)$ that are used to compute a ground height estimate $\hat{h}_g$,



a) Three propositions of rectangular silhouettes on a DEM.

b) Associated rewards given by a cost function.

c) When it is relevant, estimated roofs.

Figure 3: Our approach is based on a silhouette model, a reward function, and roof-estimation.

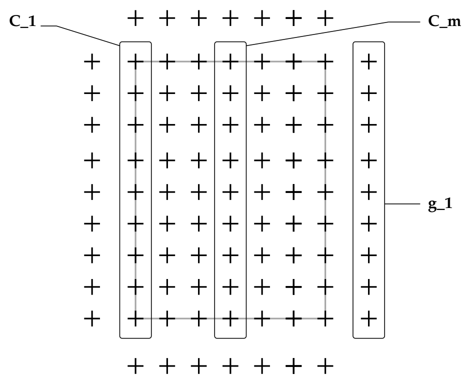


Figure 4: Points used on a silhouette to compute the cost function and roof estimation.

- the central area c , which is made of N lines along the length $c_1, \dots c_N$. In order to compute roof models, we make from these N lines m couples of lines $\{l_1, \dots, l_m\}$:

$$m = \left(\frac{N+1}{2}\right), \quad \forall i \in \{1, \dots, m\} \quad l_i = c_i \cup c_{N+1-i}$$

  So, these couples of lines are made of lines that are symmetric with respect to the length-way axis of the rectangle.

For the **ground estimate** $\widehat{h}_g$, we choose the lowest mean of the sets of points g_i :

$$\widehat{h}_g = \min_{i \in \{1,\dots,4\}} \bar{g}_i$$

From the central area, we first define $v \in [0,1]$ being the **volume rate** :

$$v = \frac{\text{card}\,\{p \in c \ \text{ s. t. } \ (h(p) - \widehat{h}_g) \geq h_{min}\}}{\text{card}\ c}$$

where h_{min} is a parameter of the model giving the minimal height we want to allow for a detected building.

The m couples of lines give us m means :

$$\forall i \in \{1, \dots, m\} \qquad m_i = \bar{l_i}$$

And we define the **homogeneity rate** $t \in [0, 1]$ being :

$$t = \frac{1}{m} \sum_{i=1}^{i=m} \frac{\text{card } \{p \in l_i \quad \text{s. t.} \quad |h(p) - m_i| < \sigma\}}{\text{card } l_i}$$

Here σ is a parameter of the model related to homogeneity.

The last rate we use is a **surface rate** : assuming that the length and the width (L, l) of a rectangle are in $[L_{min}, L_{max}] \times [l_{min}, l_{max}]$, we define $s \in [0, 1]$ being :

$$s = \frac{l * L}{l_{max} * L_{max}}$$

Finally, we can define the cost function, obtained by trial and error, as a weighted product of these several rates. For a rectangle R and a DEM h, we define the cost function J as :

$$\boxed{J(R, h) = s * t^2 * v^3}$$

This cost function has two intrinsic parameters : h_{min} which is a physical parameter, and σ that has to be tuned. We will see later in this report that this cost function can not be used directly, but need to be adapted for several reasons.

2.1.3 Roof estimation

There are quite a lot of possible roof models. The one we choose is simple and fits the cost function well : we model roofs with lines along the length, using the estimates of the cost function $m_0, \dots, m_m$ (cf. figure 5 to see the roof shape we have chosen). This shape is symmetric, since m_i stands for the mean estimate on symmetric lines along the length axis.

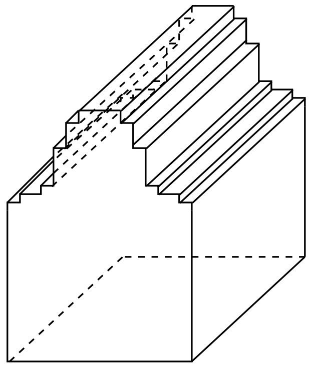


Figure 5: Model of roof used

2.2 Interactions between buildings

As we have a knowledge of the configuration of buildings in urban areas, we decide to use it in the prior. We know for instance that buildings in towns do not overlap.

The marked point process approach allows us to deal with this knowledge, and we do not need to know the number of buildings to be detected.

2.2.1 Point processes

First, we consider a point process X living in $K = [0, X_{max}] \times [0, Y_{max}]$. X is a mapping from a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ to **configurations of points** of K :

$$\forall \omega \in \Omega \quad X(\omega) = \{x_1, \ldots, x_n, \ldots\} \quad x_i \in K$$

Since K is bounded and included in $\mathbb{R}^2$, this mapping verifies the conditions defining a point process (see [24] for details) :

Conditions

1. *For all bounded Borel set $A \subseteq K$, and all $\omega \in \Omega$, the number of points of the configuration $X(\omega)$ falling in A is finite. Such configurations of points are said* **locally finite** *and the family of all locally finite configurations is denoted by N^{lf} .*

2. *For all Borel set $A \subseteq K$, the mapping $N_X(A)$ from $(\Omega, \mathcal{A}, \mathbb{P})$ to $\mathbb{N}$ giving the number of points of X falling in A is measurable, that means is a (finite) random variable.*

So, since a natural σ-algebra on N^{lf} is missing, a point process is not directly measurable, and its measurability is defined through the measurability of applications $N_X(A)$. We call $\mathcal{N}^{lf}$ the smallest σ-algebra from N^{lf} such that these applications are measurable. X takes its values in $(N^{lf}, \mathcal{N}^{lf})$. We call **distribution of** X the probability measure induced by X on $\mathcal{N}^{lf}$. A theorem given in [24] completes the previous definitions :

Theorem *The distribution of a point process X on a complete, separable, metric space χ is characterized by its* **finite-dimensional distributions** *: the collection of joint distribution of $(N(A_1), \ldots, N(A_m))$ for all finite vectors $(A_1, \ldots, A_m)$ of bounded Borel sets $A_i \subseteq \chi$, $i = 1, \ldots, m$ of any length $m \in \mathbb{N}$.*

The most famous point process is the Poisson point process, defined as follow :

Definition *Let $\nu(.)$ be a Borel measure on K, then a point process X on K is a* **Poisson process** *with* **intensity** ν *if :*

1. *$N(A)$ is Poisson distributed with mean $\nu(A)$ for every bounded Borel set $A \subseteq \chi$;*
2. *For any k disjoint bounded Borel set $A_1, \ldots, A_k$ the random variables $N(A_1). \ldots, N(A_k)$ are independent.*

The distribution of such a point process will be denoted by $\mu_\nu(.)$. Under some assumptions, it is possible to construct point processes whose laws are defined by densities against the distribution of a Poisson process.

Since we are working on K, our framework verifies these assumptions.

2.2.2 Marked point processes

We want to deal with rectangles : the space we are working on should not be K but $S = K \times M$ where $M = [-\frac{\pi}{2}, \frac{\pi}{2}] \times [L_{min}, L_{max}] \times [l_{min}, l_{max}]$. However, we do not use a point process defined on S, but a marked point process on K with it marks living in $[-\frac{\pi}{2}, \frac{\pi}{2}] \times [L_{min}, L_{max}] \times [l_{min}, l_{max}]$. That means that the point processes we are looking at are both point processes on K and on S : to each point of K we attach a mark made of an orientation, a length and a width.

2.2.3 Energy

Working on S, we first consider a marked Poisson process on K and S, with intensity measure $(\lambda_K \times \mathbb{P}_M)(.)$ where $\lambda_K(.)$ is the Lebesgue measure on K and $\mathbb{P}_M(.)$ is a probability distribution on M. We note $\mu(.)$ the distribution of this Poisson process.

It is known that the expectation of the random variable $N_X(S)$ under $\mu(.)$

is :

$$\mathbb{E}_\mu \left[N_X(S)\right] = (\lambda_K \times \mathbb{P}_M)(S) = \lambda_K(K)$$

This process will be our reference process. We are going to define the distribution $\mathbb{P}_X(.)$ of our point process of interest X using the Radon-Nikodym derivative :

$$\frac{d\mathbb{P}_X}{d\mu}(\mathbf{x}) = \beta^{n(\mathbf{x})} f(\mathbf{x})$$

Here, β is a scale factor. Such a point process X as the same distribution than a point process defined by $\beta = 1$, the same function $f(.)$ and an underlying Poisson process of intensity $\beta * (\lambda_K \times \mathbb{P}_M)(.)$. It is then equivalent to multiply either β by any factor or the underlying intensity measure by the same factor.

The usual way of defining densities of point process is to write the density under its Gibbs form :

$$\mathbf{x} = \{u_1, \ldots, u_{n(\mathbf{x})}\} \qquad u_i \in S$$

$$f(\mathbf{x}) = \frac{1}{Z} e^{-U(\mathbf{x})}$$

where $U(\mathbf{x})$ is the energy of the set of particles $\mathbf{x}$. The lower the energy, the more probable the configuration. Therefore, the final goal of our framework consists in first building an energy, then finding configurations that minimize it.

2.2.4 Overlapping

We are dealing with points that represent the shape of a building. We want to introduce in our model the knowledge we have on the behavior of buildings in urban areas.

To do so, we use a Strauss model (see [22] for more details).

First of all, we know that buildings do not overlap. That is why we introduce in our model a **soft core** term, as employed in [1] and [16]:

$$U_{inter}(\mathbf{x}) = V_{inter} * s(\mathbf{x})$$

where $s(\mathbf{x})$ is an integer counting in the configuration $\mathbf{x}$ the number of pairs of rectangles that intersect. V_{inter} is a real parameter, called **potential of intersection**.

- If $V_{inter} > 0$, intersections are made repulsive since each new intersection increases the energy of the system.
- If $V_{inter} < 0$, intersections are made attractive. But f is not normalisable, since for configurations with a large number n of points, the energy could evolve in n^2.

2.3 Data term

The aim here is to use the cost function described before and the DEM in an energy term. This energy is given by :

$$U_{data}(\mathbf{x}) = \sum_{u_i \in \mathbf{x}} V_d(u_i)$$

The data term V_d has to take into account the following points :

1. the minimums of V_d should be relevant houses on the DEM. This is ensured by the construction of the cost function.
2. V_d should be quite smooth in order to ease the optimization.

That is why, using three positive real numbers b_1, b_2, b_3, we propose the following function V_d living in $[-(b_3 + b_1), b_2]$:

$$V_d(R,h) = \begin{cases} b_2 - (b_2 + b_1) * \frac{v(R,h)}{v_{min}}^2 & \text{if } v(R,h) \leq v_{min} \\ -b_1 - b_3 * J(R,h) & \text{if } v(R,h) \geq v_{min} \end{cases}$$

for a given rectangle $R \in S$ and a height function$h(.)$. This data term has been obtained by trial and error and is quite robust. In practice, we use the following parameters :

$$\begin{array}{lcl} b_3 & = & 10 \\ b_2 & = & 0.05 \\ b_1 & = & 0 \\ \\ v_{min} & = & 0.8 \\ h_{min} & = & 3 \text{ m} \\ \sigma & = & 1.5 \text{ m} \end{array}$$

For a low volume rate (ie. less than v_{min}) the rectangle is repulsive. But the closer v is from v_{min}, the less repulsive it is (smooth term in v^2). Then, when the volume rate is large enough, the homogeneity rate and the surface rate are involved in the data term.

The surface rate makes bigger silhouettes be more valuable.

The only important parameters are b_1 that should always be equal to zero, v_{min} since it gives the authorized error, h_{min} and σ. b_2 and b_3 allow to bound the energy of an object. This is important for mathematical reasons as described later in this report.

2.4 External Field, Internal Field and Temperature

The last step of the construction of the proposed model consists in using interactions and the data term in one energy. We call

- **internal field** the part of the energy related to the interactions (ie. the prior),
- **external field** the energy related to the data term,
- and **temperature** the coefficient T :

$$U(\mathbf{x}) = \frac{1}{T}\left(\rho.U_{data} + (1-\rho).U_{inter}\right)$$

where ρ is the smoothing factor that tunes how important the data term is versus the internal field. This kind of weighting is very common in image processing.

3 Optimization

Once we have defined the model, the next step consists in defining a procedure that allows us to find the configuration minimizing the energy.
Here, this energy is related to a density of a point process, so the optimal configuration is the one that maximizes this density.
This has been down several times, especially in image processing (see [1, 16, 17]).
Most of times, proposed algorithms are Monte-Carlo samplers coupled with simulated annealing. We have decided to follow this idea.

3.1 Basic Monte Carlo Sampler for Point Process

The basic idea of such a sampler has been developed and justified by Geyer and Moller (see [8]), based on an extension of Hasting-Metropolis to point processes. The heart of the algorithm consists in birth or death of points.

3.1.1 Properties

We assume we want to simulate a point process X of density $h(\mathbf{x}) = \beta^{n(\mathbf{x})} f(\mathbf{x})$ on S, with respect to the measure $\mu(.)$ whose intensity measure is $(\lambda_K \times \mathbb{P}_M)(.)$. Then, under some assumptions, the algorithm described below builds a Markov Chain (X_t) on the space of configurations that verifies the following properties :

- The produced Markov Chain (X_t) has an invariant measure $\pi(.)$, and this measure is the law of the point process defined by $h(.)$ and $\mu(.)$, that is the law we have defined to solve our problem.
- For any initialization, (X_t) converge to $\pi(.)$. Moreover (X_t) verifies geometric ergodicity, and so the law of (X_t) converges in total variation to $\pi(.)$ with a geometric speed.

3.1.2 The algorithm

The algorithm, as all Hasting-Metropolis samplers, is made of two steps : one of proposition and one of rejection.

Algorithm

Given $X_t = \mathbf{x} = \{u_1, \ldots, u_{n(x)}\}$,

1. - *with probability* p_b*, we propose a* **birth** *and define* $\mathbf{y} = \mathbf{x} \cup u$*, where* u *is uniformly generated on* S *using*

$$\frac{\lambda_K \times \mathbb{P}_M}{\lambda_K(K)}(.)$$

   Then we define :

$$R(\mathbf{x}, \mathbf{y}) = \frac{p_d}{p_b} \frac{\lambda_K(K)}{n(\mathbf{x}) + 1}$$

   - *with probability* p_d*, we propose a* **death** *and define* $\mathbf{y} = \mathbf{x} \setminus u$*, where* u *is uniformly chosen in* $\mathbf{x}$*.*
   This case gives :

$$R(\mathbf{x}, \mathbf{y}) = \frac{p_b}{p_d} \frac{n(\mathbf{x})}{\lambda_K(K)}$$

2. *The rejection step begins with the computation of the acceptance rate* α *as :*

$$\alpha = \min(1, \frac{h(\mathbf{y})}{h(\mathbf{x})} R(\mathbf{x}, \mathbf{y}))$$

   and,

   - *with probability* α *the proposition is accepted :* $X_{t+1} = \mathbf{y}$
   - *with probability* $1 - \alpha$ *the proposition is rejected :* $X_{t+1} = \mathbf{x}$

3.1.3 Hypothesis

The main assumption concerns the density we want to simulate. The point process has to be stable. This notion of stability has been introduced by Ruelle in [18, 19] for statistical physics on set of particles. Definition of **Ruelle's stability** is :

definition | *A point process of density $h(.)$ with respect to a Poisson process is* **stable**, *if there exits a real M s. t. :*

$$\frac{h(\mathbf{x} \cup u)}{h(\mathbf{x})} \leq M$$

for all $\mathbf{x}$ and $u \in S$.

Note that we only need to know $h(.)$ up to a normalizing constant to test this condition.

The model proposed in this report verifies this condition.

3.2 Improved sampler

However, this algorithm produces a highly correlated Markov chain. As shown in [2, 12] while working with MCMC, one has to be careful with the proposition kernel involved in the sampler. [20] provides a good overview of other techniques that can be used to improve Markov Chain behavior.

So, in order to improve the results, we use the work of Green in [10]. This work has been analyzed for point processes by Geyer in [7].

The idea is to add other transformations to the proposition kernel and to have a mixture of kernels. The 'birth or death' transformation has to remain, since it gives the mathematical properties we need. We can add any transformation $q_m(.,.)$, assuming we can compute its Green's ratio $R_m(\mathbf{x},\mathbf{y})$ (see [10]). This gives the following algorithm :

Algorithm *Given* $X_t = \mathbf{x} = \{u_1, \ldots, u_{n(x)}\}$,

1. *we randomly select a transition kernel* q_m *according to the mixture distribution.*
2. *we generate* $\mathbf{y}$ *according to the selected kernel :*
$$\mathbf{y} \sim q_m(\mathbf{x}, .)$$
3. *we compute the Green ratio* $R_m(\mathbf{x},\mathbf{y})$
4. *we compute the acceptance rate* α *as :*
$$a = \min(1, \frac{h(\mathbf{y})}{h(\mathbf{x})} R_m(\mathbf{x},\mathbf{y}))$$
and,
   - *with probability* α *the proposition is accepted :* $X_{t+1} = \mathbf{y}$
   - *with probability* $1-\alpha$ *the proposition is rejected :* $X_{t+1} = \mathbf{x}$

We use a general form of transformation : these transformations are symmetric (ie. $q_m(\mathbf{x},\mathbf{y}) = q_m(\mathbf{y},\mathbf{x})$) and their Green ratio is always equal to 1. Moreover, all these transformation first choose a point u uniformly in $\mathbf{x}$, and set $\mathbf{y} = \mathbf{x} \setminus u \cup v$, where v is generated with a law depending on u (for a more general framework, see the annex).

3.2.1 Translation

We generate uniformly (δ_x, δ_y) in $[-d_x, +d_x] \times [-d_y, +d_y]$ and, if $u = (x, y, \theta, L, l)$ we set $v = (x + \delta_x, y + \delta_y, \theta, L, l)$, considering for this transformation that K is a torus, in order to obtain the symmetry of the transformation.

3.2.2 Rotation

We generate uniformly δ_θ in $[-d_\theta, +d_\theta]$ and if $u = (x, y, \theta, L, l)$ we set $v = (x, y, \theta + \delta_\theta, L, l)$.

3.2.3 Dilation

The main idea is the same : we want to randomly modify the length or the width of a rectangle. However, for dynamics reasons, it appears that it is better to modify both the length or the width and the center of a rectangle, in order to let one side of the rectangle be constant under the transformation, as shown by figure 6. The mathematical description of this transformation is presented in the annex.

3.3 Simulated annealing

Once a sampler has been defined, it is possible to optimize the density, using simulated annealing, as described by Winkler in [26]. A good overview on simulated annealing techniques can be found in [23].

In the previous algorithm, we replace $h(\mathbf{x}) = \beta^{n(\mathbf{x})} f(\mathbf{x})$ by :

$$h_t(\mathbf{x}) = \frac{\beta^{n(\mathbf{x})}}{Z} \exp\left(\frac{-U(\mathbf{x})}{T_t}\right)$$

with temperature depending on the time of the Markov chain.

In theory, if we use the above sampler and make simultaneously decrease the temperature T_t from T_{init} to 0, (using a logarithmic law), the chain converges in total variation to a Dirac measure, whose mass is equally distributed on the global minimas of $U(.)$.

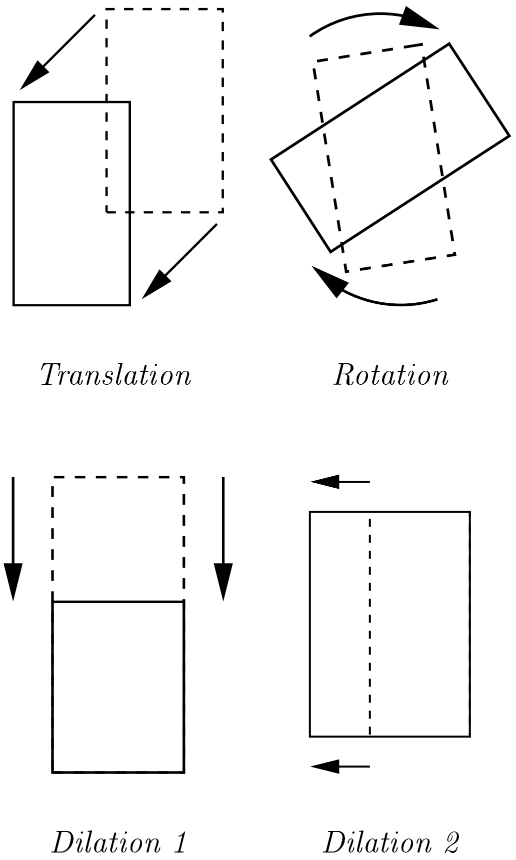


Figure 6: Transformations used in the proposition kernel.

Hence, using the previous sampler with such a decreasing law allows to find the minima of the energy defined in section 2.
Of course, since this kind of decrease is slow, we have to use in practice a geometric one and we loose the theoretical properties of global optimization.

4 Results

4.1 Description

We present below a result obtained on a part of the DEM of the city of Amiens (France). Here, small intersections are allowed, for 3D visualization reasons.

The proposed algorithm gives a list of buildings in a text file. Figure 7 shows the result of the detection. A few comments can be made :

- there are some border effects : in the top left corner of Figure 7 a building is irrelevant. This is due to the ground estimate, which is made with only 2 points, since other points of the upper side are out of the area. These kind of phenomena can be regulated by imposing a minimum number of points while estimating values like ground height.
- some small and low buildings are missed, because of the structure of the data which is too smooth where buildings are low.

Figure 8 (top) shows the results of roof estimation and the ground truth (bottom) provided by the French Mapping Institute (IGN). This ground truth is precise since it has been built by hand using very high resolution aerial images. As shown by Figure 8, some buildings are missing, and details of the shape of buildings are missed, due to the rectangular silhouette model we have chosen.

Figure 9 shows the data and the result in 3D, and allows to see that mistakes are due to artifacts on the data.

4.2 Comments

The results presented here have been obtained in 40 minutes with a `SUN-blade 2` (500 MHz, 250 MB). The data image size is 1060 by 1024.

As shown by the ground truth, the quality of the results obtained by the proposed model is hard to improve. We should be able to deal with even

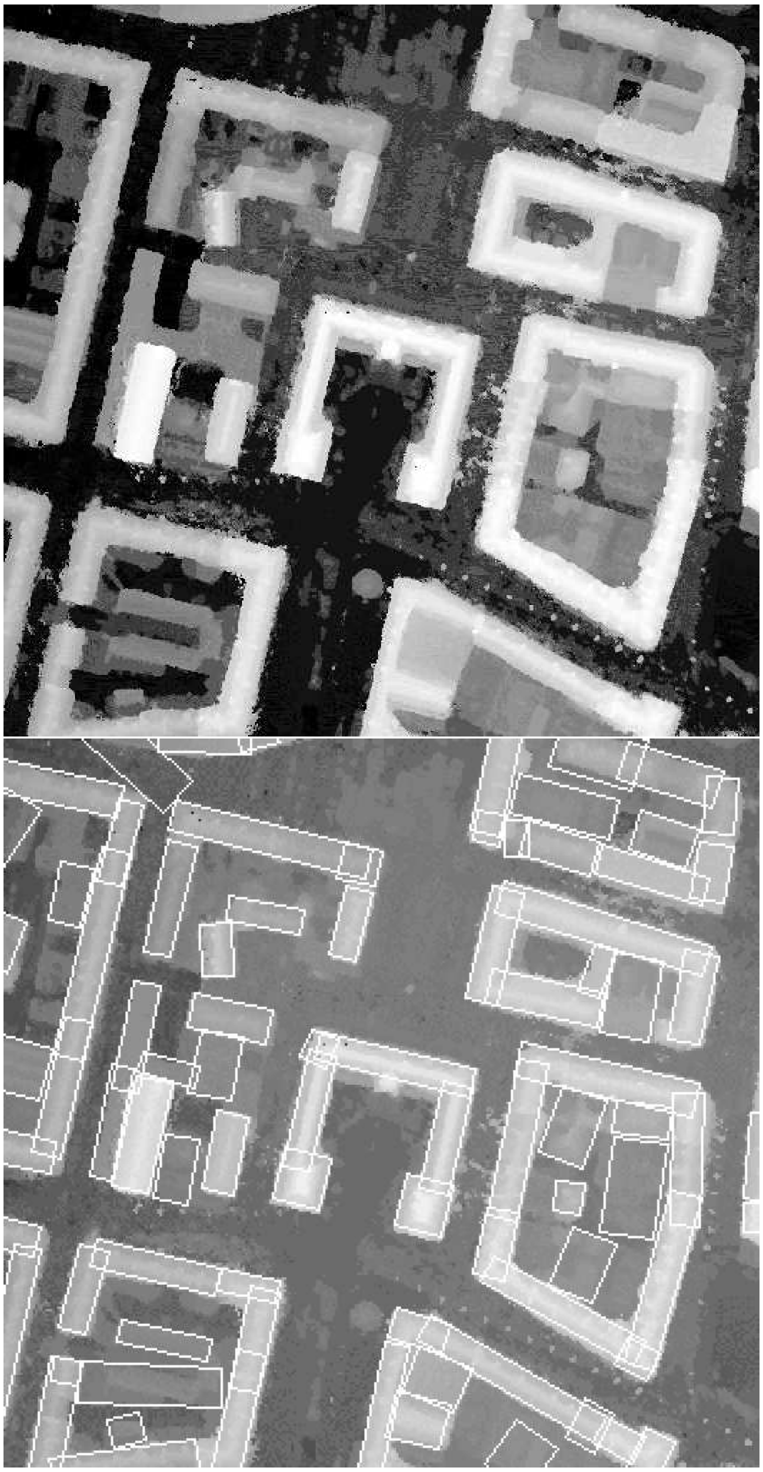


Figure 7: Result on the DEM. (Top : Crude DEM, bottom : result of the building extraction)

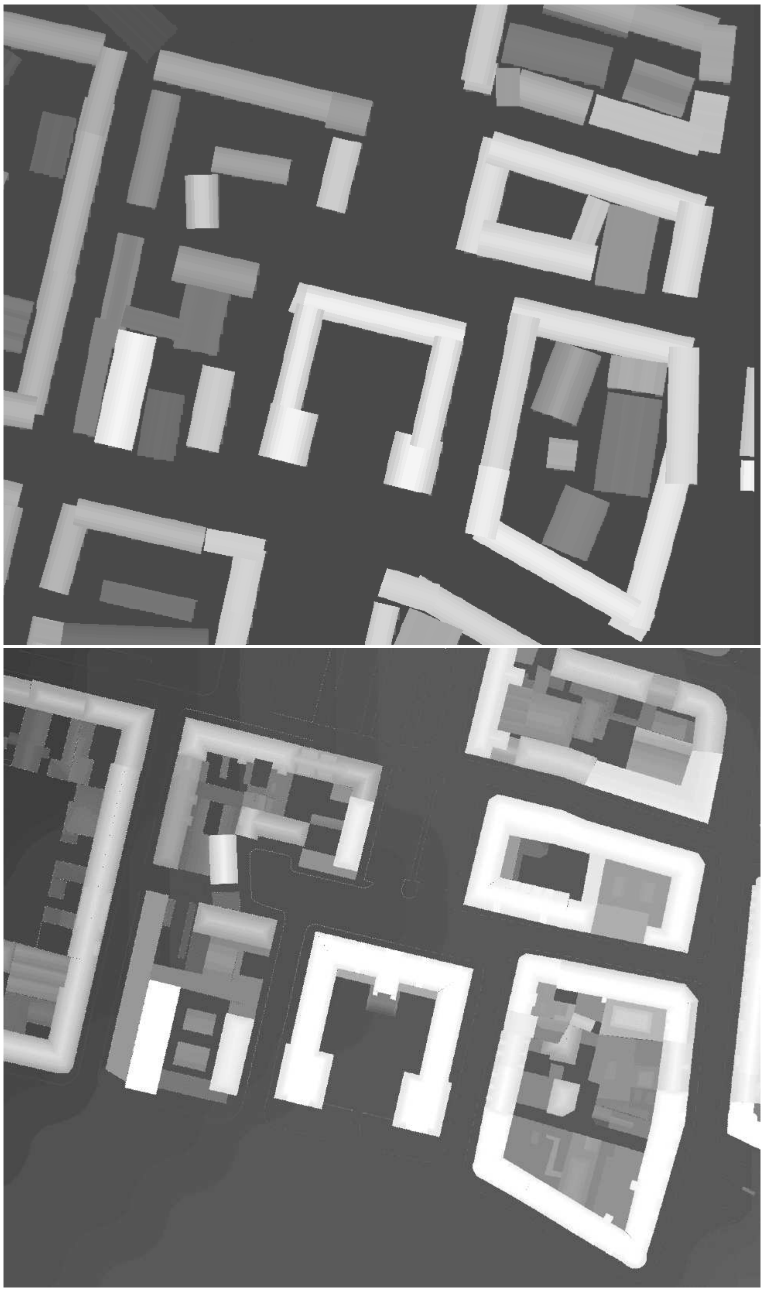


Figure 8: Validation. Top : estimated roofs, bottom : ground truth manually obtained by IGN.

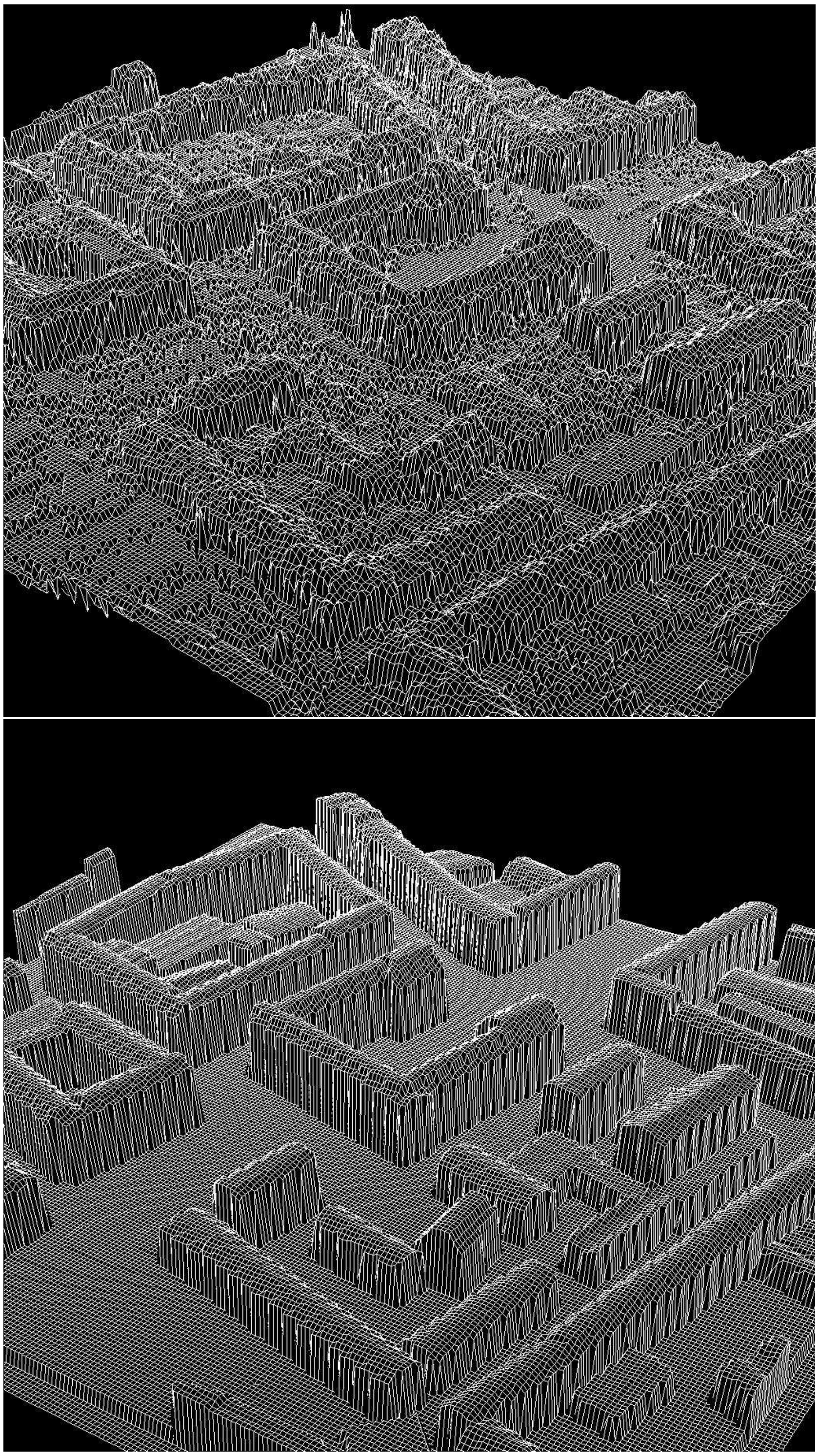


Figure 9: Visualization in 3 dimensions. Top : crude DEM, bottom : refined DEM obtained by the proposed method.

worst data, if we could optimize more complex energies. Tests have been done (see [14]) with more complex prior models, using relations such as alignment or orthogonality between buildings.

Results are not improved with a more complex prior model due to the optimization step. Simulated annealing looses its efficiency if the energy exhibits too many local minimas. Designing a good prior model and estimating its parameters is then hard to achieve, since we cannot obtain optimal configurations.

Therefore, one way to improve the proposed technique is to work on a more efficient optimization algorithm. This will allow to improve the model and use more complex silhouettes and a bigger set of possible roofs. This will be done in a near future.

Conclusion and future work

Automatic construction of 3D maps of towns is a difficult problem since urban areas can be very dense. Usual techniques to achieve this goal rely on radiometric information, and use very complex procedures involving a lot of parameters.

We have proposed in this report a new method to refine DEM.

This method actually gives an automatic way of extracting buildings from a dense urban area by using a crude DEM as initial condition.

A first advantage comes from the kind of results given by the procedure . Given a raster data, we obtain a vector which is useful for semantic interpretation and compression goals.

Moreover this extraction is fast and efficient. It also provides robust estimation of roofs.

The originality of this method can be appreciated as follows :

- First, the object oriented approach we have chosen gives an elegant way of adding geometric constraints during the extraction.
- Second, the way of adding a prior knowledge as interactions between buildings gives a nice framework to deal with poor quality data.
- Finally, since the data term is built using a projection of a proposed silhouette on the data, it is easy either to change the kind of data used in the energy term, or to add a complementary energy term using another data.

However, this work can be improved by :

- testing the algorithm on more DEMs (optical, radar, laser etc...),
- improving the algorithm to be faster and to deal with larger areas,
- adding more complex models of buildings and roofs,

- improving the optimization step in order to be able to add a more complex prior. Encountered problems are related to the high number of local minimas of energy which appear when we add a better a priori knowledge in order to deal with cruder data.

A Annex : RJMCMC for point processes

A.1 General framework

Reversible Jump Monte Carlo sampling has been proposed by Green in [10]. He gives a general framework to ensure ergodic convergence of the generated Markov chain to a desired distribution, in the case where the Markov chain is living on an union of vectorial spaces of different dimensions. Below is presented a way to apply this result to our point process framework.

We assume we are working on a space $\mathcal{C}$. In our case $\mathcal{C} = N_S^{lf}$ that is the set of locally finite configurations of points of S.

We want to simulate a measure $\pi(.)$. In our case, this measure is defined by a density $h(.)$ against a Poisson process with law $\mu(.)$ and a intensity measure $\nu(.) = (\lambda_K \times \mathbb{P}_M)(.)$

We want to build a Markov chain $(X_t)_{t\in\mathbb{N}}$ on $\mathcal{C}$ which is :

- **aperiodic**,
- **irreducible**,
- with **stationary law** $\pi(.)$.

These conditions, as shown in [15], give ergodic convergence of (X_t) to measure $\pi(.)$. We note K the transition kernel of the Markov chain.

Aperiodicity is easy to obtain, since it is sufficient to insure that K allows to stay in a state with positive probability.

Irreducibility is given, in our case, by Geyer and Moller algorithm of birth or death for point processes, if stability condition is verified. We obtain more (see [7]) since Harris recurrence and geometric ergodicity follows from this condition.

At this point, only the stationary distribution is missing. The work of Green [10] is involved here.

We consider a **proposition kernel** $Q(.,.)$ on $\mathcal{C} \times \mathcal{C}$ made of a mixture of kernels $q_m(.,.)$:

$$\forall \mathbf{x} \in \chi \quad Q(\mathbf{x},.) = \sum_m p_m(\mathbf{x}) q_m(\mathbf{x},.) \tag{1}$$

While in state $\mathbf{x}$, proposition of a state $\mathbf{y}$ consists in :

- first, choosing a kernel q_m with probability $p_m(\mathbf{x})$,
- then, generating $\mathbf{y}$ with distribution $q_m(\mathbf{x},.)$.

To be sure that the Markov chain is $\pi(.)$ invariant, it is sufficient to ensure that :

$$\begin{array}{ll} \forall m, & \pi(d\mathbf{x}) q_m(\mathbf{x}, d\mathbf{x}') \text{ has a finite density } f_m(\mathbf{x}, \mathbf{x}') \\ & \text{with respect to a symmetric measure } \quad \xi_m \text{ on } \mathcal{C} \times \mathcal{C} \end{array} \tag{2}$$

And to define as Green's ratio :

$$R_m(\mathbf{x}, \mathbf{x}') = \frac{f_m(\mathbf{x}', \mathbf{x})}{f_m(\mathbf{x}, \mathbf{x}')} \tag{3}$$

If this condition is insured, then the Markov chain built by the following algorithm is $\pi(.)$ invariant.

Algorithm | *If state of the Markov chain at time t is $X_t = \mathbf{x}$,*

1. *Choose a kernel $q_m(\ .\ ,\ .\ )$ with probability $p_m(\mathbf{x})$.*
2. *Sample $\mathbf{y}$ according to the chosen kernel $q_m(\mathbf{x},\ .\ )$*
3. *Compute Green's ratio $R_m(\mathbf{x}, \mathbf{y})$ and acceptance probability :*

$$\alpha = \min(1, R_m(\mathbf{x}, \mathbf{y}))$$

4.
   - *with probability α, accept the proposition : $X_{t+1} = \mathbf{y}$,*
   - *with probability $1 - \alpha$, reject the proposition : $X_{t+1} = \mathbf{x}$*

A.2 Heart of the sampler : birth or death

The birth or death sub-kernel is the most important one, since it ensures geometric ergodicity of the Markov chain.

Here, we aim at showing that the condition (2) is verified and calculating Green's ratio. For the latter, we need to describe the density f_m of the 'birth or death' sub-kernel.

We suppose that birth generates a point in S according to the probability distribution $\frac{\nu(.)}{\nu(S)}$ and that death uniformly choose a point in the current configuration. We first write this kernel as :

$$q = p_b Q_b + p_d Q_d \tag{4}$$

with the two kernels of birth and death being :

$$Q_b(\mathbf{x}, A) = \int_{u \in S} \mathbf{1}_A(\mathbf{x} \cup u) \frac{\nu(du)}{\nu(S)} \tag{5}$$

and

$$Q_d(\mathbf{x}, A) = \sum_{u \in \mathbf{x}} \mathbf{1}_A(\mathbf{x} \setminus u) \frac{1}{n(\mathbf{x})} \tag{6}$$

To verify condition (2), we first consider the following measure ξ , where A and B are Borel sets of S.

$$\xi(A \times B) = \int_{\mathcal{C}} \int_{u \in S} \mathbf{1}_A(\mathbf{x}) \mathbf{1}_B(\mathbf{x} \cup u) \nu(du) \mu(d\mathbf{x}) + \int_{\mathcal{C}} \mathbf{1}_A(\mathbf{x}) \sum_{u \in \mathbf{x}} \mathbf{1}_B(\mathbf{x} \setminus u) \mu(d\mathbf{x}) \tag{7}$$

We first need to show that this measure is symmetric. This comes from the fact that $\nu(.)$ is the intensity measure of the Poisson process law $\mu(.)$. Let us take $A_n = A \cap N_n^f$ where N_n^f is the subset of N_S^{lf} made of configurations of n points. Definition of a Poisson process gives (see [24] for details) :

$$\begin{aligned}
\xi(A_n \times B_{n-1}) &= \frac{e^{-\nu(S)}}{n!} \int_{S^n} \sum_{u \in S} \mathbf{1}_{A_n}(\mathbf{x}) \mathbf{1}_{B_{n-1}}(\mathbf{x} \setminus u) \nu^n(d\mathbf{x}) && \text{(8a)} \\
&= \frac{e^{-\nu(S)}}{n!} \int_{S^n} n \mathbf{1}_{A_n}(\{x_1, \ldots, x_n\}) \mathbf{1}_{B_{n-1}}(\{x_1, \ldots, x_n\}) \nu^n(d\mathbf{x}) && \text{(8b)} \\
&= \frac{e^{-\nu(S)}}{(n-1)!} \int_{\S^{n-1}} \int_S \mathbf{1}_{B_{n-1}}(\mathbf{y}) \mathbf{1}_{A_n}(\mathbf{y} \cup u) \nu^{n-1}(d\mathbf{y}) \nu(du) && \text{(8c)} \\
&= \xi(B_{n-1} \times A_n) && \text{(8d)}
\end{aligned}$$

From equation (8), follows the symmetry of ξ.

Now, we need to show that $\xi(d\mathbf{x}, d\mathbf{x}')$ dominates $\pi(d\mathbf{x}) Q(\mathbf{x}, d\mathbf{x}')$ and to calculate the Radon-Nikodym derivative.

It is useful to observe in equations (8) that the measure $\xi(A, B)$ is null when A is made of configurations that cannot be obtained by adding or removing a point from any configuration of B.

Hence, if a set $A \times\ B$ has a strictly positive $\pi(.)Q(.;.)$ measure, its ξ measure is also strictly positive.
Thus, we have two cases to consider :

1. if $\mathbf{y} = \mathbf{x} \cup u$, then expressions of π and Q give :

$$\pi(d\mathbf{x})Q(\mathbf{x}, d\mathbf{y}) = h(\mathbf{x})\mu(d\mathbf{x})p_b\frac{\nu(du)}{\nu(S)}$$

   And definition of ξ (cf. equation (7)) gives :

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

   from which follows absolute continuity and Radon-Nikodym derivative :

$$\boxed{f(\mathbf{x}, \mathbf{y}) = p_b\frac{h(\mathbf{x})}{\nu(S)}} \tag{9}$$

2. The other case consists in considering : $\mathbf{y} = \mathbf{x} \setminus u$.

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

   Measure ξ gives :

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

   and the derivative :

$$\boxed{f(\mathbf{x}, \mathbf{y}) = p_d\frac{h(\mathbf{x})}{n(\mathbf{y})}} \tag{10}$$

We may infer Green's ratio from equations (9) and (10). This coefficient has two different expressions, depending on how $\mathbf{y}$ is obtained (ie by adding or removing a point in $\mathbf{x}$) :

1. In case of a **birth**, $\mathbf{y} = \mathbf{x} \cup u$, and Green's Ration is given by :

$$\boxed{R(\mathbf{x}, \mathbf{y}) = \frac{f(\mathbf{y}, \mathbf{x})}{f(\mathbf{x}, \mathbf{y})} = \frac{p_d}{p_b}\frac{h(\mathbf{y})}{h(\mathbf{x})}\frac{\nu(S)}{n(\mathbf{y})}} \tag{11}$$

   where $n(\mathbf{y})$ can be replaced by $n(\mathbf{x}) + 1$.

2. In case of a **death**, $\mathbf{y} = \mathbf{x} \setminus u$, and Green's Ratio is given by :

$$\boxed{R(\mathbf{x},\mathbf{y}) = \frac{f(\mathbf{y},\mathbf{x})}{f(\mathbf{x},\mathbf{y})} = \frac{p_b}{p_d}\frac{h(\mathbf{y})}{h(\mathbf{x})}\frac{n(\mathbf{x})}{\nu(S)}} \tag{12}$$

This gives Green's ratio for the birth or death sub kernel.

A.3 Other transformations

We consider a random variable Z living on $\Sigma \subseteq \mathbb{R}^k$ where Σ is bounded.

We note $\mathbb{P}_Z(.)$ the law of Z, and we suppose that this law has a density $f_Z(.)$ with respect to the Lebesgue's measure $\lambda_\Sigma(.)$.

We then consider a function ζ:

$$\begin{array}{rlll} \zeta : & U \times \Sigma & \rightarrow & U \\ & (u,z) & \rightarrow & v \end{array} \tag{13}$$

The first constraint on ζ is that $\zeta(u,.)$ should be an **injection** for any u.

The sub-kernel used can be described as follows :
Denoting : $\mathbf{x} = \{u_1, \ldots, u_{n(\mathbf{x})}\}$,

1. Choose one of the u_i with a discrete probability law $p(\mathbf{x}, u_i)$.
2. Generate z using the distribution of Z.
3. Propose $\mathbf{y} = \mathbf{x} \setminus u_i \cup \zeta(u_i, Z)$.

From a mathematical point of view such a kernel can be written as :

$$Q(\mathbf{x}, A) = \sum_{u\in\mathbf{x}} p(\mathbf{x}, u) \int_{z\in\Sigma} \mathbf{1}_A(\mathbf{x} \setminus u \cup \zeta(u,z)) f_Z(z)\lambda(dz) \tag{14}$$

We consider a measure ξ . For A and B in $\mathcal{C}$,

$$\xi(A \times B) = \int_A \sum_{u\in\mathbf{x}} \int_{z\in\Sigma} \mathbf{1}_B(\mathbf{x} \setminus u \cup \zeta(u,z))\lambda(dz)\mu(d\mathbf{x}) \tag{15}$$

We then impose symmetry to Σ and the following condition of **parity** to ζ :

$$v = \zeta(u,z) \Leftrightarrow u = \zeta(v,-z) \tag{16}$$

This condition makes ξ symmetric as required. We verify, using injectivity of $\zeta(u,.)$, that $\pi(d\mathbf{x})Q(\mathbf{x},d\mathbf{y})$ is dominated by ξ and that its Radon-Nikodym derivative is given by :

$$f(\mathbf{x},\mathbf{y}) = p_{\mathbf{x}}(u)h(\mathbf{x})f_Z(z) \tag{17}$$

if $u = \mathbf{x}\backslash(\mathbf{x}\cap\mathbf{y})$, and if z is the unique parameter defined by $\mathbf{x} = \mathbf{y}\backslash u \cup \zeta(u,x)$. We then can write the expression of Green's Ratio, using the same notations :

$$\boxed{R(\mathbf{x},\mathbf{y}) = \frac{f(\mathbf{y},\mathbf{x})}{f(\mathbf{x},\mathbf{y})} = \frac{p_{\mathbf{y}}(v)}{p_{\mathbf{x}}(u)}\frac{h(\mathbf{y})}{h(\mathbf{x})}\frac{f_Z(-z)}{f_Z(z)}} \tag{18}$$

This recalls the usual Hasting Metropolis update.

Note that the possibility to choose one point in a configuration with a discrete probability law that is not uniform could be extended to birth or death updates.

A.3.1 Translation

We consider ζ given by :

$$\zeta(.;z) : u = \begin{pmatrix} X \\ Y \\ l \\ L \\ \theta \end{pmatrix} \rightarrow v = \begin{pmatrix} X + z_X \\ Y + z_Y \\ l \\ L \\ \theta \end{pmatrix} \tag{19}$$

If $\zeta(u_i,z) \notin S$, we can either reject the proposition, or view S like a torus.

A.3.2 Rotation

For this transformation, we use : $z = \theta \in \Sigma = [-\delta_\theta; +\delta_\theta] \subseteq \mathbb{R}$ and ζ :

$$\zeta(.;z) : u = \begin{pmatrix} X \\ Y \\ l \\ L \\ \theta \end{pmatrix} \rightarrow v = \begin{pmatrix} X \\ Y \\ l \\ L \\ \theta + \delta_\theta \end{pmatrix} \tag{20}$$

A.3.3 Dilations

We use four kinds of dilations. We give an example below made of two of them (ζ_0 and ζ_1). These two are modifying the length of a rectangle while the two others dilations related to the width.

Let $k \in \{0, 1\}$, $z = z_L \in \Sigma = [-\delta_L; +\delta_L]$ and define ζ_k as :

$$\zeta_k(.;z) : u = \begin{pmatrix} X \\ Y \\ l \\ L \\ \theta \end{pmatrix} \rightarrow v = \begin{pmatrix} X + z_L \cos(\theta + k * \pi) \\ Y + z_L \sin(\theta + k * \pi) \\ l \\ L + z_L \\ \theta \end{pmatrix} \tag{21}$$

If $v \notin S$, we reject the proposition.

References

[1] A. Baddeley and M. N. M. van Lieshout. Stochastic geometry models in high-level vision. In K.V. Mardia, editor, *Statistics and Images*, volume 1, pages 233–258. Abingdon: Carfax, 1993.

[2] S.P. Brooks, P. Guidici, and G.O. Roberts. Efficient construction of reversible jump MCMC proposal distributions (with discussion). *Journal of the Royal Statistical Society, Series B*, To appear.

[3] A. Fischer, T. H. Kolbe, F. Lang, A. B. Cremers, W. Förstner, L. Plümer, and V. Steinhage. Extracting buildings from aerial images using hierarchical aggregation in 2D and 3D. *Computer Vision and Image Understanding: CVIU*, 72(2):185–203, 1998.

[4] M. Fradkin, M. Roux, and H. Maître. Building detection from multiple views. In *ISPRS Conference on Automatic Extraction of GIS Objects form Digital Imagery*, 1999.

[5] M. Fradkin, M. Roux, H. Maître, and U. Leloglu. Surface reconstruction from multiple aerial images in dense urban areas. In *Proc of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 262–267, Fort Collins, Colorado, USA, June 1999.

[6] L. Garcin, X. Descombes, J. Zerubia, and H. Le Men. Building detection by Markov object processes and a MCMC algorithm. *INRIA Research Report 4206*, June 2001.

[7] C. J. Geyer. *Stochastic Geometry Likehood and computation*, chapter Likehood Inference for Spatial point Processes. Chapman and Hall, 1999.

[8] C.J. Geyer and J. Møller. Simulation and likehood inference for spatial point process. *Scandinavian Journal of Statistics*, Series B, 21:359–373, 1994.

[9] S. Girard, P. Guerin, H. Maître, and M. Roux. Building detection from high resolution colour images. In S. Serpico, editor, *International Symposium on Remote Sensing, Europto'98*, volume 3500, pages 278–289, Barcelona, September 1998.

[10] P.J. Green. Reversible jump Markov chain Monte-Carlo computation and Bayesian model determination. *Biometrika*, 57:97–109, 1995.

[11] A. Gruen and R. Nevatia (eds). Special issue on automatic building extraction from aerial images. *Computer Vision and Image Understanding (CVIU)*, 72, 1998.

[12] P. Guidici and G.O. Roberts. On the automatic choice of reversible jumps. *Research Report, University di Pavia*, June 1998.

[13] H. Mayer. Automatic object extraction from aerial imagery-a survey focusing on buildings. *Computer Vision and Image Understanding (CVIU)*, 74(2):138–149, 1999.

[14] M. Ortner. Extraction de caricatures de bâtiments sur des modèles numériques d'élévation. Master Thesis (DEA, in French), August 2001. `http://www-sop.inria.fr/ariana/personnel/Mathias.Ortner/rapport_DEA_ortner.ps.gz.`

[15] C. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer-Verlag, New York, 1999.

[16] H. Rue and M. Hurn. Bayesian object identification. *Biometrika*, 3:649–660, 1999.

[17] H. Rue and A. R. Syverseen. Bayesian object recognition with Baddeley's delta loss. *Adv. Appl. Prob*, 30:64–84, 1998.

[18] D. Ruelle. *Statistical mechanics.* Benjamin, New York, 1969.

[19] D. Ruelle. *Superstable Interactions in Classical Statistical Mechanics*, chapter 18, pages 127–159. Commun. Math. Phys. , Springer-Verlag, 1970.

[20] J. Sénégas. *Méthodes de Monte-Carlo en vision stéréoscopique. Application à l'étude de Modèles Numériques de Terrain.* PhD thesis, Ecole des Mines de Paris, Sept. 2002. (in French).

[21] A. Stassopoulou and T. Caelli. Building detection using bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(6):715–733, 2000.

[22] D.J. Strauss. A model for clustering. *Biometrika*, 62:467–475, 1975.

[23] P.J.M. van Laarhoven and E.H.L. Aarts. *Simulated Annealing: Theory and Applications*. Reidel, Dordrecht, 1987.

[24] M. N. M. van Lieshout. *Markov Point Processes and their Applications*. Imperial College Press, London, 2000.

[25] U. Weidner. Building extraction from digital elevation models. Technical report, Institut für Photogrammetrie, Bonn, 1995.

[26] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Springer-Verlag, 1987.

Acknowledgments

The authors would like to thank the French Mapping Institute for providing the data, Marie Colette Van Lieshout and Radu Stoica from CWI for several interesting discussions. The work of the first author has been partially supported by the French Defense Agency (DGA) and CNRS.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis 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 - 38330 Montbonnot-St-Martin (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