INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Automatic 3D land register extraction from altimetric data in dense urban areas

Mathias Ortner — Xavier Descombes — Josiane Zerubia

N° 4919

August 2003

THÈME 3

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4919--FR+ENG

Automatic 3D land register extraction from altimetric data in dense urban areas

Mathias Ortner , Xavier Descombes , Josiane Zerubia

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

Rapport de recherche n° 4919 — August 2003 — 66 pages

Abstract: This work present an automatic algorithm that extract 3D land register from altimetric data in dense urban areas.

Altimetry of a town is a data which is easily available yet difficult to exploit. For instance, we present here results on two kind of measurements : the first one consists in a Digital Elevation Model (DEM) built using a correlation algorithm and some optical data, while the second one consists in a DEM obtained by Laser measurments.

Our main objective is to design an entirely automatic method that is able to deal with this kind of data in very dense urban areas.

We thus focus on elementary shape extraction and propose an algorithm that extracts rectangular buildings. The result provided consists in a kind of vectorial land register map that can be used, for instance, to perform precise roof shape estimation.

The proposed algorithm uses our previous work. Using a point process framework, we model towns as configuration of rectangles. An energy is defined, that takes into account both a low level information provided by the altimetry of the scene, and some geometric knowledge of the disposition of buildings in towns.

The estimation is done by minimizing the energy using a simulated annealing.

We use a MCMC sampler that is a combination of general Metropolis Hastings Green techniques and Geyer and Møller algorithm of sampling of point processes. We use some original proposition kernels, such as birth or death in a neighborhood and define the energy with respect to an inhomogeneous Poisson point process.

We present results on real data provided by IGN (French Mapping Institute). Results were automatically obtained, on areas that are 200m by 200m large. These results consist in configurations of around 100 rectangles describing considered areas with an error of 15% of missclassification.

Key-words: Automatic feature extraction, image processing, dense urban area, Digital Elevation Models, Laser data, land register, building detection, point process, inhomogeneous Poisson point process, MCMC, RJMCMC, simulated annealing.

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

Extraction automatique de cadastre 3D sous forme vecteur à partir d'une description altimétrique de zones urbaines denses.

Résumé : Ce travail présente un algorithme qui extrait automatiquement un plan cadastral de la description altimétrique (relief) d'une zone urbaine dense.

L'altimétrie d'une ville est une donnée qui est maintenant facilement accessible. Dans ce rapport, nous présentons par exemple des résultats sur deux types de données altimétriques : le premier consiste en un Modèle Numérique d'Elévation (MNE) obtenu par corrélation d'images optiques, le second correspond à un MNE obtenu par mesure LASER.

Notre objectif principal est de définir un algorithme entièrement automatique capable d'extraire un grand nombre de bâtiments dans des zones urbaines denses.

Nous nous intéressons donc plus particulièrement à l'extraction de formes élémentaires et proposons un algorithme qui modélise les bâtiments par des formes rectangulaires. Le résultat obtenu consiste en une carte cadastrale qui peut être utilisée pour faire une estimation précise des formes de toits, par exemple.

L'algorithme proposé ici repose sur nos travaux précédents. Nous modélisons des villes par des configurations de rectangles auxquelles nous associons une énergie définie de manière à tenir compte aussi bien d'une information de bas niveau provenant des données utilisées que d'une connaissance géometrique de l'agencement des bâtiments dans les zones urbaines.

L'estimation est ensuite faite en minimisant l'énergie définie grace à un recuit-simulé.

Nous utilisons un échantilloneur MCMC qui est une combinaison de techniques générales de type Metropolis Hastings Green et de l'algorithme de simulation de processus ponctuel proposé par Geyer et Møller. Nous utilisons en particulier des noyaux de proposition originaux comme la naissance ou mort dans un voisinage, et nous définissons l'énergie par rapport à un processus ponctuel de Poisson non-homogène, ce qui permet d'améliorer le comportement dynamique de l'algorithme.

Les resultats que nous présentons sont obtenus sur des donnée réelles fournies par l'IGN. Nous extrayons automatiquement des configurations composées d'une centaine de bâtiments sur des zones dont la taille est en moyenne de 200m sur 200m. L'erreur commise est en moyenne de 15 % par rapport a la vérité terrain.

Mots-clés : Extraction automatique de forme, traitement d'image, zone urbaine dense, Modèle Numérique d'Elévation, detection de bâtiments, donnée LASER, cadastre, processus ponctuel, processus ponctuel de Poisson inhomogène, MCMC, RJMCMC, recuit-simulé

Contents

1 Introduction

1.1 Urban areas, third dimension and remote sensing

A huge amount of applications require to be able to automatically build 3D precise maps of towns. Such applications can be related to cartography, urban planning, simulations, military intelligence, etc...

The third dimension is now an important issue. There is a real need of 3D precise maps of towns. Of course, there are several ways of representing a town in 3 dimension. The most simple is the raster data : each pixel of an image stands for a height, the most complex is the precise vectorial description: each building is described by a set of 3D primitives.

High resolution data in remote sensing have made aerial and satellitar images very useful to analyze urban areas. A lot of sensors are now available, at different costs. For instance, the commercial US satellite Ikonos gives images at 1m of resolution, European SPOT5 gives images at 2.5m, while aerial imagery gives centimetric resolution. LASER or SAR imaging give specific data, and finally there is a wide variety of available data that can be used to process dense urban areas.

In this work, we focus on the French town of Amiens, for which a set of data has been provided by the French Mapping Institute (IGN). Figure 1 presents an aerial shot of the center of the town, figure 2 shows a Digital Elevation Model of this area obtained by a correlation algorithm on optical data and figure 3 exposes a Laser data.

Figure 1: Aerial image (25cm) of Amiens provided by the French Mapping Institute (IGN).

1.2 Data, automaticity and problematics

Depending on the application, on the data provided and on the precision required, several methods and algorithms have already be proposed to deal with dense urban areas.

General overviews can be found in [2, 11]. It is important to notice the different approaches that exist on this subject .

- a first topic of importance is related to **segmentation and classification**. Intra urban analysis is thus done by classical image analysis (like texture analysis), see [23] for instance. SAR or optical data at medium resolution are used (from 1m to 5m).
- a second topic concerns **surface reconstruction**. Proposed solutions usually try to introduce semantic knowledge coupled with some low level processing. See [4] for instance

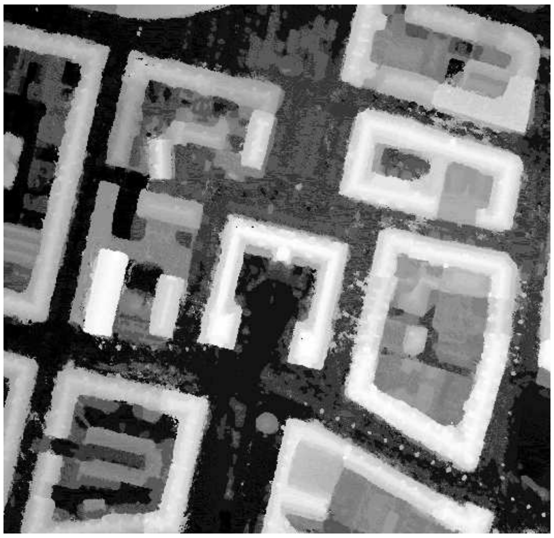


Figure 2: Digital Elevation Model (25cm,10cm) of Amiens

Figure 3: Laser Data (50cm,10cm) of Amiens

- a last topic deals with **object detection and reconstruction**. In [10] a very interesting work is proposed to reconstruct polyhedrical models of buildings, using a vectorial 2D land register map, and high resolution images. In [9] and [12] an automatic and a semi-automatic methods are described.

Some research has been conducted, trying to fuse all kinds of data and problematics like in [2, 3, 7].

The proposed method focuses on the last topic: we are interested in building detection and modeling.

1.3 Building detection and modeling

Our main goal is to obtain some vectorial representation of a town. It means that somewhere a correlation has to be made between a low level information (image analysis) and a high level knowledge (buildings) (see [10] to have an excellent overview of the different methods used on the topic).

Relationship between low level signal and high level interpretation is usually drawn using a bottom-up coupled with a top-down procedure. Primitives are detected on the data, aggregated using some rules and constraints (bottom up) while results of these procedures are compared with a library of given building models (top down). These procedures are complex, so in order to improve the quality of the result and the speed of the processing steps, the following points are explored nowadays:

- improvement of the quality of detected primitives, by using a large variety of data such as: multiple color images, Laser data, or external data like register maps or hyper-spectral images,
- simplification of buildings models in order to ease top-down procedures,
- introduction of hierarchical approaches to use several layers of primitives,
- development of other methods like grammar on graphs (see [5] for instance).

Usually, these methods have to face a combinatorial problem because they mostly rely on fusion of primitives.

The goal of this work is:

- to use as few data as possible possible (actually, only altimetric measurements are needed),
- to be able to deal with large and dense urban areas(hundreds of buildings),
- to build a vector representation of a town using a simple model of buildings (prismatic),
- to design an entirely automatic method.

This work can be seen as a medium-level extractor. The results are presented as a vectorial 3D land register map. This work can be used to ease or improve algorithms presented in [10] (see introduction of [10] in particular).

S. Vinson and L. D. Cohen did some work on the same topic. They extract rectangles from DEMs using an ortho-image and a pre-segmentation of the DEM. Their approach and results are described in [21] and [22].

1.4 Digital Elevation Model

First, we choose to use only a crude DEM, since this kind of data is available quite easily, using stereovision on optical images or laser measurements.

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

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 build 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 2 shows a DEM of the French town of Amiens (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 10 cm vertically). The noise is a geometrically correlated noise due to the algorithms used to construct the DEM.

1.5 Point processes

Spatial point processes were introduced in image processing by Van Lieshout and Baddeley in [1] to detect an unknown number of objects. Rue and Hurn in [17] 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 them. 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.

This kind of models gives another advantage since the algorithms used are entirely automatic.

In [14, 15] we already presented such a model to detect buildings on DEM. We present here a better model, that allows to be almost exhaustive, to be precise and efficient.

1.6 Goals and interest

In our first model ([14]), buildings have been detected, but some of them where missing, especially in very dense urban areas. Another problem was related to the precision of the result.

As we will see later, it is hard to meaningfully improve the precision of the results using stochastic exploration.

In [16], we have proposed some amelioration to our usual point process sampler. The work presented here uses these improvements. It also presents an elegant way of coupling local low level filters with high level prior knowledge.

With respect to our previous research on point processes, the main originality of this work is the use of a non-homogeneous reference point process, with an impressive improvements of the quality of results.

2 Model definition

2.1 Spatial point process

2.1.1 Introduction

Spatial point process

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

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

Since K is bounded and included in $\mathbb{R}^2$, this mapping defines a point process (see [20] for details).

Basically, a point process is a random variable whose realizations are random configurations of points.

Note that K can be a torus.

Poisson point process

The simplest point process is the Poisson point process. Let $\nu(.)$ be a positive measure on S. We consider a **Poisson point process** X with **intensity measure** $\nu(.)$ on K.

If we consider the discrete Poisson distribution with mean $\nu(K)$:

$$p_n = e^{-\nu(K)} \frac{\nu(K)^n}{n!}$$

a way of generating a realization of X is the following:

1. generate $N(K) = n$, number of points in the configuration following the previously described discrete distribution,
2. generate $\{x_1, \ldots, x_n\}$ by n independent simulations of the distribution given by $(\frac{\nu}{\nu(K)})(.)$ and Lebesgue measure on K.

Figure 4 shows realizations of two Poisson point processes, the first one using a uniform measure $\nu(.)$ and the second one using an intensity measure that gives three times more weight to the upper left corner of K.

For a point process on K, the **intensity measure** gives the average number of points falling in every Borelian set of K. Let note $N_A(X)$ the random number of points of X falling in $A \subset K$. For a Poisson point process defined by its intensity measure $\nu(.)$, the following property holds:

$$\mathbb{E}[N_A(X)] = \nu(A)$$

Marked Point process

Point processes were introduced in image processing because they easily allow to model scenes of objects. A marked point adds some marks (parameters) to each point. Let take $S = K \times M$ with $M \subseteq \mathbb{R}^{d-2}$. For instance, to describe rectangles, we use the following M:

$$M = [-\frac{\pi}{2}, \frac{\pi}{2}] \times [L_{min}, L_{max}] \times [l_{min}, l_{max}]$$

Figure 5 presents a realization of a point process of rectangles.

We note $\mathcal{C}$ the set of all finite configurations of points of S:

$$\mathbf{x} \in \mathcal{C} \qquad \text{iff} \quad \begin{cases} \exists n \in \mathbb{N} \\ \mathbf{x} = \{x_1, \ldots x_n\} \\ \forall i \quad x_i \in S \end{cases}$$

Density of a spatial point process

Let consider the distribution $\mu(.)$ of a given Poisson point process which we call **reference Poisson point process**. Under some conditions (detailed in [20]) it is possible to define a point process X by specifying its probability distribution by a density with respect to the dominating distribution $\mu(.)$.

Let consider a mapping $h(.)$ from the space of configuration of points $\mathcal{C}$ to $\mathbb{R}$, and a real parameter Z such that:

$$\int_{\mathcal{C}} \frac{1}{Z} h(\mathbf{x}) d\mu(\mathbf{x}) = 1$$

For instance, let consider:

$$h(\mathbf{x}) = \Pi_{i=1}^{n(\mathbf{x})} \beta(x_i)$$

where $\beta(.)$ is an **intensity** function defined on S. A point process X specified by such a density is a Poisson point process with intensity

$$\nu'(A) = \int_A \beta(u) d\nu(u)$$

$\nu(.)$ being the intensity measure of the reference Poisson point process.

This example belongs to a more general class of densities defined by exponential families: suppose a mapping $t(.)$ from $\mathcal{C}$ to $\mathbb{R}^t$ is defined, it is possible to describe the class of densities using a parameter $\theta \in \mathbb{R}^t$ and the scalar product $< .,. >$:

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

This leads to the second example of point process defined by a density, which is given by the **Strauss process**. Let introduce the following two mappings: $n(\mathbf{x})$ that counts the number of points in the configuration $\mathbf{x}$, and $s(\mathbf{x})$ that counts the number of pairs of points in $\mathbf{x}$ that are closer than some parameter value d_{max}:

$$s(\{x_1, \ldots, x_n\}) = \text{card}\,\{\{i,j\} \quad : \quad i \neq j \;\; \|x_i - x_j\| \leq d_{max}\}$$

and define

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

e^{θ_1} can be seen as an intensity parameter that tunes the average number of points of the point process, while θ_2 is an interaction parameter. If $\theta_2 > 0$, then interacting pairs of points are favored. In this case, $h(.)$ is not integrable. If $\theta_2 < 0$, the point process X with density $h(.)$ is well defined and configurations with less interacting pairs are made more probable.

2.1.2 Estimator and MCMC

If we are given a point process X defined by

- an unnormalized density $h(.)$,
- a intensity measure $\nu(.)$ defining a reference Poisson point process distribution,

it is possible to build a Markov Chain that converges ergodically to the distribution of X. In [16] we detailed an algorithm producing a Markov Chain $(X_t)_{t\geq 0}$ with following properties:

- $(X_t)_{t\geq 0}$ ergodically converges to the distribution of X,

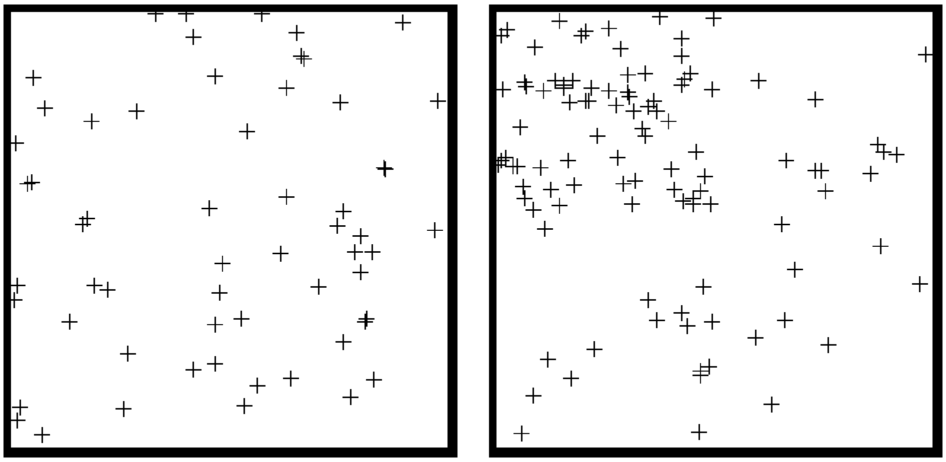


Figure 4: Realizations of two different Poisson point processes ; left : with a uniform intensity measure, right: with an intensity measure giving more weight to the upper left corner.

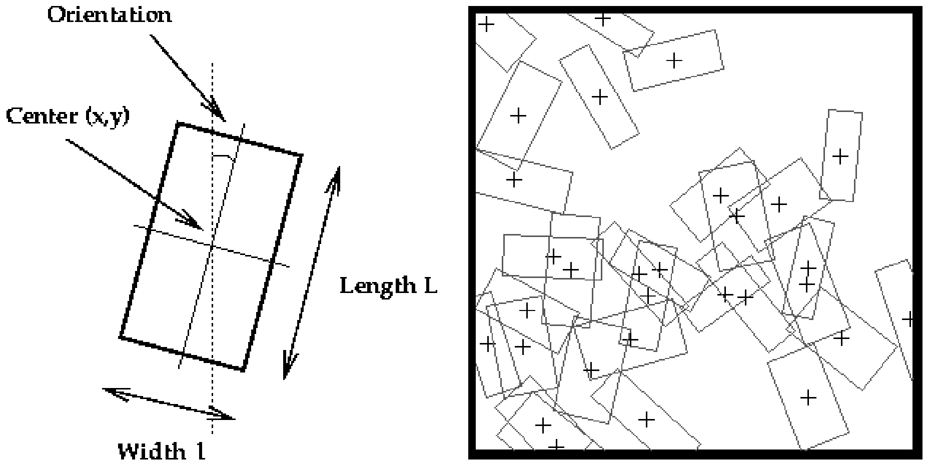


Figure 5: Parameterization of a rectangle and realization of a uniform Poisson point process of rectangles with mean 20.

- $(X_t)_{t\geq 0}$ is Harris recurrent (every starting point is suitable),
- the ergodic convergence is geometric.

Once such a sampler is defined, any Monte Carlo value is computable. It includes mean values, variances etc...

Another way of using this sampler is to use it within a **simulated annealing** framework which gives a global maximum of the density $h(.)$ as detailed in [24]. It means the estimator we obtain is given by:

$$\hat{\mathbf{x}} = Argmax \ \ h(.)$$

2.2 Point processes and buildings

Our main idea is to use marked point processes to model urban areas. It is important first to point out that the model we use is very simple since we look at configurations of rectangles.

Our goal is indeed to obtain a simple land register 3D map of a dense urban area described by its Digital Elevation Model.

The originality of this work is to combine a low level information given by a local filter applied on the DEM with a high level information described by interactions between rectangles of the designed point process.

2.2.1 Silhouette and third dimension

The first choice we have to make concerns the kind of objects we want to detect. It could have been lines or polygons, but we have chosen rectangles for several reasons:

- rectangles have few parameters. Limiting the dimension of S limits the size of $\mathcal{C}$ and makes the optimization easier,
- tests done with lines revealed two difficulties: there are a lot of lines of interest in a DEM, and a non closed line is hard to interpret from a semantic point of view. For instance a line detection in three dimension on a DEM gives results which are difficult to be interpreted in a reconstruction procedure,
- land register maps are often presented as a collection of polygons which appear to be close to rectangles. It makes the rectangular shape seem to be a natural pattern to be detected on a DEM.

The third dimension is still an open issue. Roof shapes could be estimated jointly to the silhouette of a building by adding one or more parameters to the five ones describing a two dimensional rectangle. However, it is easier

- first to detect rectangular shapes on the data,
- then to estimate roof shapes using any suitable method.

2.2.2 Configurations of objects, image and energy

The first question arising from the point process model is related to the density $h(.)$. In image processing, two main types of models are generally used.

Bayesian modeling

A Bayesian modeling requires to be able to exhibit likelihood function describing the distribution of a DEM I being given a configuration of rectangles $\mathbf{x}$.

$$L(I/X = \mathbf{x}) = L(\mathbf{x}, I) \qquad h(\mathbf{x}) \propto L(\mathbf{x}, I)h_{prior}(\mathbf{x})$$

In this case the optimization is done over the **a posteriori** density obtained by multiplying the **likelihood** giving the correspondence between data and the underlying probabilist model and some **a priori** density, chosen either for statistical properties fitting well the application or for mathematical good behavior.

The nice property of such a model is given by Bayes' risk: the **a posteriori** maximizer corresponds to the value minimizing the expected quadratic Bayes risk under the **a priori** distribution.

In [17], H. Rue does Bayesian inference on images using a point process as the underlying object of interest. Since he tries to detect clear objects on a dark background, an obvious model for the distribution of gray level is given by a mixture model.

In our setup, such a Bayesian model would require to accurately describe the distribution of heights in every area of the DEM that is not a building. Since it involves rivers or trees, roads and cars or just the simple non-flat ground, it appears to be very hard to model the background of a DEM[1].

The foreground model would also be hard to obtain, since if rectangles describe silhouettes of buildings, their surface stand for roofs which can be very different from one house to a neighboring one.

That is why we have focused in this report on the other class of models.

Energy definition

As usually done when working for instance active contours, it is easier and better to define an energy term that is coherent with the goal of our application. We define a density under its energetic form:

$$h(.) \propto e^{-U(\mathbf{x})} \quad U(\mathbf{x}) = U_{int}(\mathbf{x}) + U_{ext}(\mathbf{x})$$

Here, $U_{int}(\mathbf{x})$ stands for an internal energy giving a spatial structure to the configuration $\mathbf{x}$, while $U_{ext}(\mathbf{x})$ is the external field quantifying the quality of a configuration given the data. The simplest way of defining such a data term is to define a sum over points in a configuration:

$$U_{ext}(\mathbf{x}) = \sum_{u \in \mathbf{x}} U_d(u) \tag{1}$$

$U_d(.)$ is thus a mapping from S to $\mathbb{R}$ quantifying the relevance of an object with respect to the DEM.

If $U_d(u) \leq 0$, the object u is attractive: apart from the internal energy of the configuration, if u is added to any configuration $\mathbf{x}$ of points, $\mathbf{x} \cup u$ is more probable than $\mathbf{x}$. Since the chosen estimator is the maximizer of probability density $h(.)$, U_d needs to be correctly defined. A good solution is to use a binary test giving a support on which U_d is negative.

Another problem is related to superposition of points. In equation 1 it is obvious that if $U_d(u) \leq 0$, then successive additions of u in the configuration x increases the probability of the current configuration: $h(\mathbf{x} \cup u \cup u) \geq h(\mathbf{x} \cup u) \geq h(\mathbf{x})$. We need an exclusive term avoiding such a superposition, while in Bayesian modeling such a term is generally not useful since a second u adds nothing to the distribution of I given $\mathbf{x} \cup u \cup u$.

The prior model we use also needs some exclusive term, and finally we propose to define a model with the following structure:

$$U(\mathbf{x}) = \sum_{u \in \mathbf{x}} U_d(u) + U_{int}(\mathbf{x}) + U_{excl}(\mathbf{x}) \tag{2}$$

Where $U_{excl}(\mathbf{x})$ is such that:

$$U(\mathbf{x} \cup u \cup u) > U(\mathbf{x} \cup u) \quad \forall\, (u, \mathbf{x}) \in S \times \mathcal{C}$$

2.3 Low level filter

In this section, we focus on the analysis of the altimetric data. In particular, we present how the data term $U_d(.)$, mapping from S to $\mathbb{R}$, has been designed.

[1]The background of a DEM in our context is everything that is not a building.

2.3.1 Introduction

Slices

The filter we need should recognize elements of buildings on a DEM. Since objects have local properties, the filter should involve some local detector. In [14] we made two hypotheses on gray level distribution in a building's silhouette. However, this filter appeared to be inefficient when trying to detect more complex structures.

Here we propose here to use slices that are measured on the DEM orthogonally to a rectangle. Figure 6 shows a part of a DEM, a rectangle on it and slices evenly disposed. Figure 7 shows some examples of profiles measured on the DEM.

Slices presented on Figure 8 shows that some interesting points can be extracted.

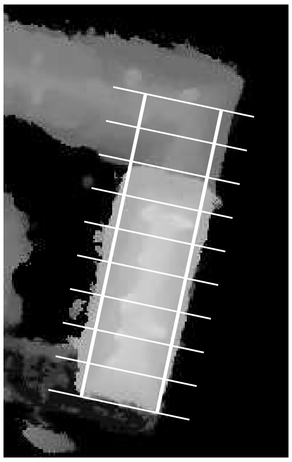


Figure 6: A DEM, a rectangle and slices of interest.

Points of Interest and geometrical structure

We propose to extract points of interest on profiles and to verify that the set of such points detected on the set of profiles is coherent with the geometrical structure of the rectangle.

The points of interest we propose to use are points where there is a discontinuity on the DEM. Being given a direction, we select a point if the gradient at this location and in this direction is larger than a threshold.

Basically, our goal is to detect rectangles with high gradients along its sides in the orthogonal direction to the rectangle, as introduced by Figure 8 . To compute $U_d(u)$, we propose to use a low level algorithm that relies on the following procedure:

For a given rectangle $u \in S$:

1. compute all slices related to this rectangle,
2. for each slice detect points of interest,
3. using these points compute:
   - a binary test deciding if the object is relevant (see discussion following equation (1), page 13): this gives the support of U_d,
   - a cost function corresponding to values of U_d.

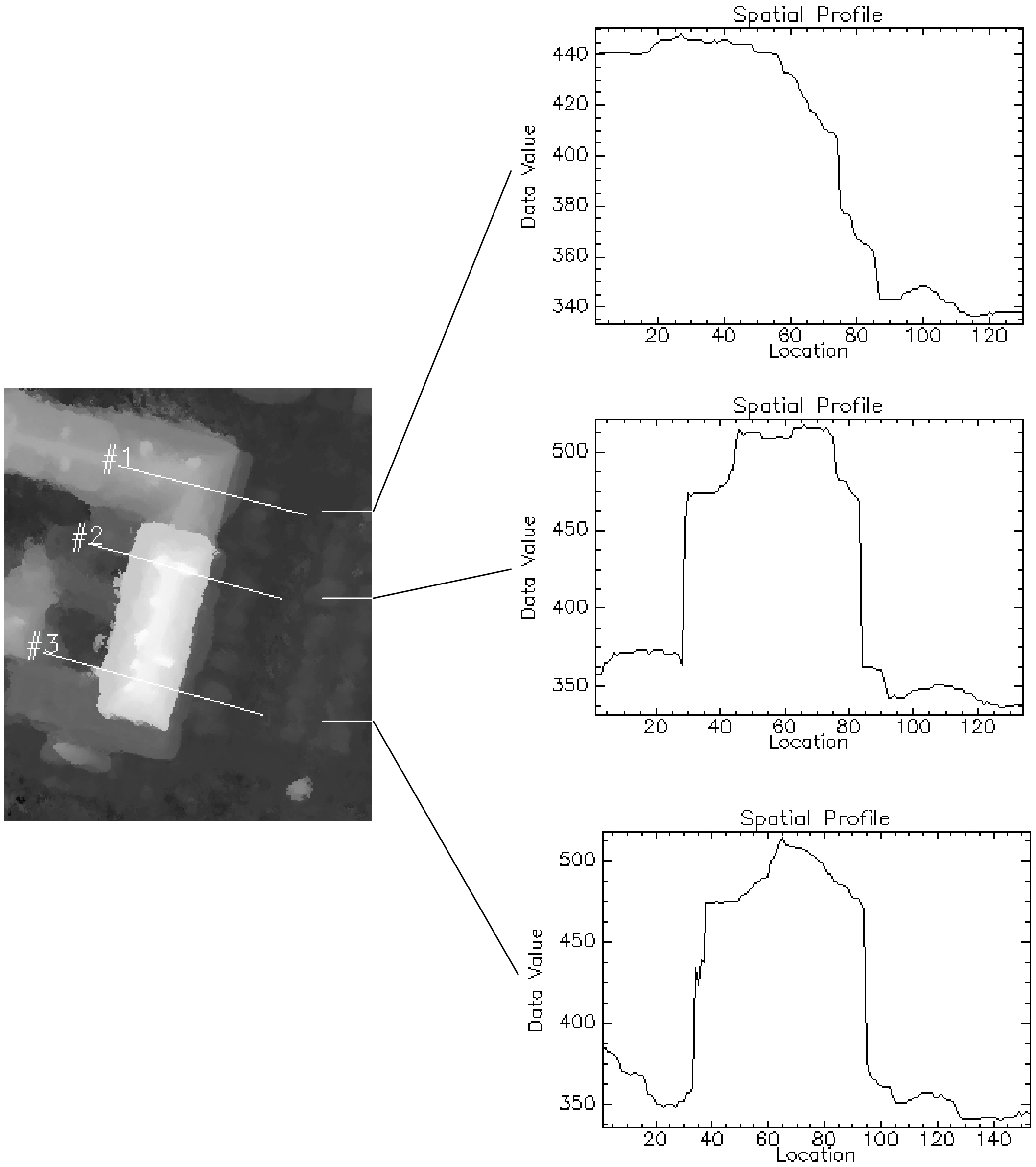


Figure 7: Example of measured slices on a part of a DEM.

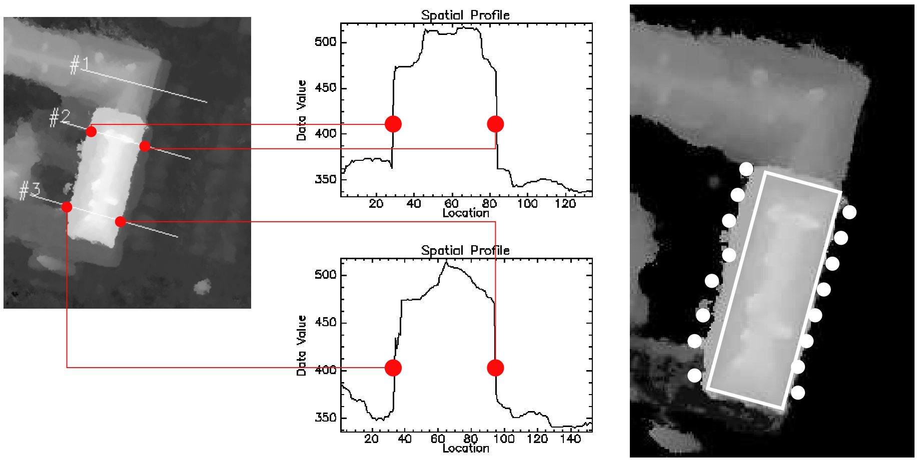


Figure 8: Slices, points of interest and rectangles.

2.3.2 Profile simplification

Profiles acquisition

For a given rectangle u, we use three parameters that define a mask of points:

- a resolution e standing for the distance between two successive profiles,
- a resolution r corresponding to the sampling resolution of the elevation on this profile,
- a parameter l_{ext} corresponding to the length to be explored in the neighboring of the rectangle.

Figure 9 shows the mask of points that are used to compute the profiles on the DEM. (e and r are two parameters of sub-sampling).

Now, let present below the algorithm that is used to extract points of interests.

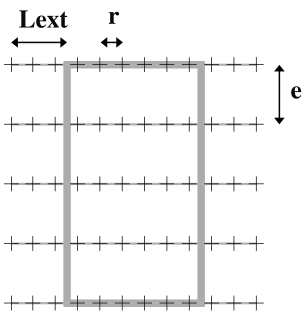


Figure 9: Construction of profiles using two resolution parameters and an external length.

Extraction of discontinuities and selection

Let consider one line standing for one profile in Figure 9. It is easy to compute gradients on the profile and select the ones which are larger than a threshold σ.

However, DEMs are usually noisy, and thus a pre-processing step is often needed. This pre-processing step may be adapted to the kind of data used.

The filter is based on the following ideas:

1. on a profile, detect gradients higher than a low threshold σ_l. This allows to first detect parts of the profile where the slope is larger than some threshold.
2. do some summing up of detected gradients, and in particular, accumulate them.
3. once regularization is done, select discontinuities larger than a high threshold σ_h.

Let define $p(.)$ as the profile of interest: $\forall i \in \{1, \ldots, N\}$, $p(i)$ represents a corresponding height (in meter) on the DEM. Let g be the corresponding gradient vector and for each location i, let define $sel(i)$ as a characteristic value set to 1 or -1 if the subscript i is selected (i.e. if a discontinuity has been detected at this location) and 0 otherwise. The sign of $sel(i)$ depends on the sign of the corresponding discontinuity.

Algorithm

1 `Compute the vector` g `of gradients:`

$$\forall i \in \{1, \ldots, N-1\} \quad g(i) = p(i+1) - p(i)$$

2 `Select the gradients larger than` σ_l`:`

$$\forall i \in \{1, \ldots, N-1\} \quad sel(i) = \begin{cases} 1 & \text{if } g(i) \geq \sigma_l \\ -1 & \text{if } g(i) \leq -\sigma_l \\ 0 & \text{otherwise} \end{cases}$$

The first idea was to take σ_l large enough and thus have a direct detector of discontinuities.

However, three main problems appeared:

- A large discontinuity can be badly localized. For instance, on Laser data, some smoothing has been done. Thus, the discontinuity is really soft. In this case, between two successive points, the difference is not so large.

  Moreover, it is sometimes important to take into account the slope of the roofs.
- Putting σ_l small enough to detect 'smooth' discontinuities gives large sequences of selected subscripts, as illustrated by Figure 10. A solution is to sum up a sequence of selected subscripts in one location.
- Some of the large gradients are false alarms either because they correspond to small artifacts on the DEM (chimney, tree) or due to the correlation algorithm.

Accumulation

The first issue is described by Figure 10. On this figure, a fictive profile is drawn under which gradients are presented. Selected gradients (larger than σ_l) are boxed. Last line presents gradients we should consider to detect buildings. A simple solution consists in reading the profile, accumulating successive selected gradients depending on their sign, and finally selecting accumulated gradients larger than σ_h.

Two detected gradients are accumulated by the following algorithm if there are less than l_{regul} far from each other.

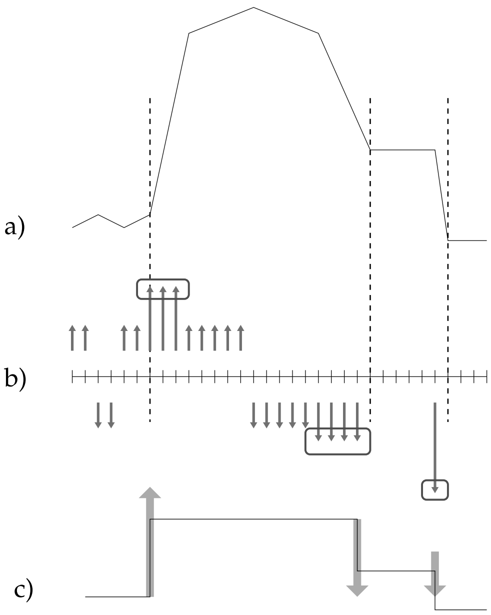


Figure 10: A fictive example of a profile: a) profile and rectangle side (dotted lines), b) computed and selected gradients (boxed) and c) accumulated gradients that should be used for building detection.

Algorithm

3 `Accumulate positive gradients:`

```
tem=0
last_pos=0
last_acc=0
```

For i=1 to i=N-1
  if sel(i)=1
    if tem=1 and d(last_pos,i)$\leq l_{regul}$
```
      sel(i)=0
      last_pos=i
      g(last_acc)=g(last_acc)+g(i)
```
    else
```
      last_pos=i
      last_acc=i
      tem=1
```
  if sel(i)=-1
```
   tem=0
```

4 `Accumulate negative gradients:`

```
tem=0
last_neg=0
```

For i=1 to i=N-1
  if sel(i)=-1
    if tem=1 and d(last_neg,i)$\leq l_{regul}$
```
      sel(last_neg)=0
      g(i)=g(last_neg)+g(i)
      last_neg=i
```
    else
```
      last_neg=i
      tem=1
```
  if sel(i)=1
```
   tem=0
```

Opening

Figure 11 presents an example related to the second issue. It corresponds to the case where two following large gradients ($\geq \sigma_h$), one positive and one negative compensate each other.

Such artifacts can be removed by using a morphological operator of opening. However, this is time consuming, since an opening relies on an erosion followed by a dilatation operation (see [18]).

That is why we use a heuristic method which consists in subtracting two following accumulated gradients and selecting the result if it is larger in absolute value than σ_h.

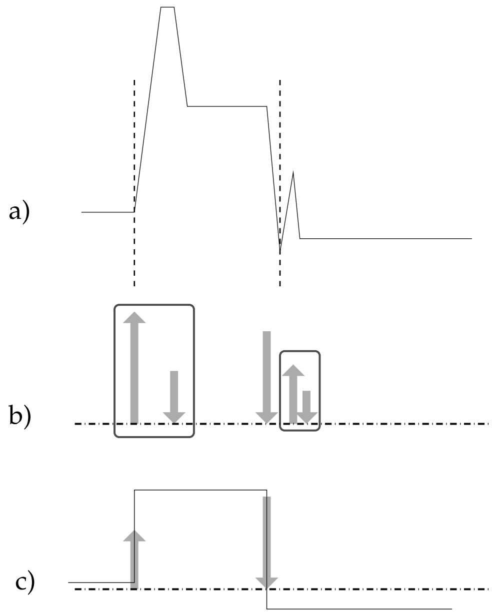


Figure 11: Fictive example of a profile with two artifacts : a) fictive profile, b) selected and accumulated gradients, and c) gradients that should be subtracted for building detection.

Algorithm

```
5   Opening step:

    last_pos=0

    For i=1 to i=N-1
         if sel(i)=1
              last_pos=i
         if sel(i)=-1 and d(last_pos,i)≤ l_regul
              g=g(i)+g(last_pos)

              if g≥ 0
                   sel(i)=0
                   g(last_pos)=g
              else
                   sel(last_pos)=0
                   g(i)=g
```

Of course the last step consists in selecting discontinuities larger than σ_h:

Algorithm

6 `Select gradients larger than` σ_h`:`

For i=1 to i=N-1
  if sel(i)=-1 or sel(i)=1
    if g(i)$\leq \sigma_h$ **and g(i)** $\geq -\sigma_h$
      `sel(i)=0`

Comments

The simple algorithm described above simplify the profile by selecting interesting gradients on it. It should be noticed that steps 1 to 4 can be compacted in a single loop, the same can be done for steps 5 and 6, minimizing the number of elementary operations needed to process a profile. The whole filter only needs two loops over the subscripts to select discontinuities of interest.

The filter relies on 4 parameters: r, σ_l, σ_h and l_{regul} (all defined in meters). This is useful since these parameters have a physical meaning. In practice they appeared to be quite robust.

Experimental results

The first Annex gives some experimental results provided by this filter on true data. It presents the influence of parameters r, σ_l, σ_h and l_{regul}, and results on the two types data used (optical and LASER).

2.3.3 Values of interest

The low level filter gives a simplified representation of the concerned profile. We propose to use this simple representation to compute some useful values that are to be used in our cost function.

Let introduce some notations, illustrated by Figure 12:

- i stands for an integer subscript allowing to iterate on a profile,
- $sel(.)$ indicates selected subscripts on a profile provided by the filter,
- $sp(.)$ describes the simplified profile provided by the filter
- and i_1 and i_2 stand for indices corresponding to rectangle sides.

These notations concern one simplified profile. We now assume that a rectangle gives Ne simplified profiles , denoted by $sp^1, \ldots, sp^{Ne}$.

Let present the 4 steps used to sum up observations provided by simplified profiles.

Successive points

The first step consists in selecting two points among points of interest given by the low level filter. We actually try to detect extruded objects on the DEM. So we need to focus on two successive discontinuities.

To do this, the closest discontinuity from one of the two rectangle sides is first found, and then the next is also selected.

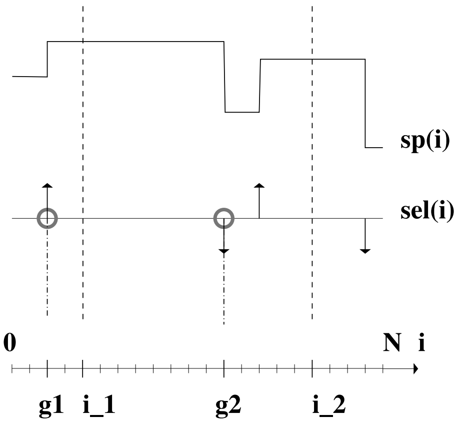


Figure 12: Result of low level filter: a simplified profile, corresponding selected gradients (circles) and subscript notations. Dotted lines correspond to rectangle sides.

Definition `First selected discontinuity:`

$$g_1 = \text{Argmin}_{i\in\{1,N\}\ \text{s.t.}\ \ sel(i)\neq 0} \min(|i-i_1|,|i-i_2|)$$

Once the first discontinuity is chosen, the second one is selected as the successive one in the good direction. For sake of simplicity, we assume in the following that the first selected gradient ($i = g_1$) is closer to i_1 than to i_2.

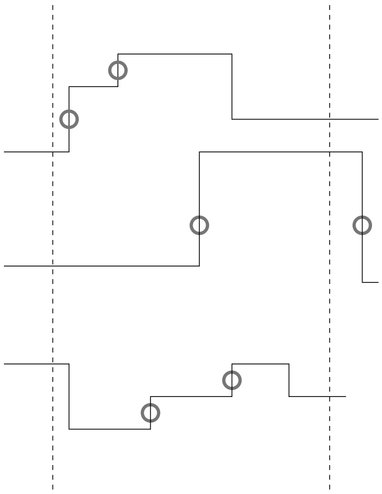
An exception appears when the first discontinuity corresponds to a negative gradient ($sel(g_1) = -1$) and the second one to a positive gradient ($sel(g_2) = 1$). In this case the selected couple of points corresponds to a hole on the DEM and thus the selected couple is the next suitable one.

Figure 13a) shows three examples. Each line stands for a simplified profile obtained using the filter. Vertical dotted lines stand for the location of the two rectangle sides. For each profile, the two selected discontinuities are marked using a circle.

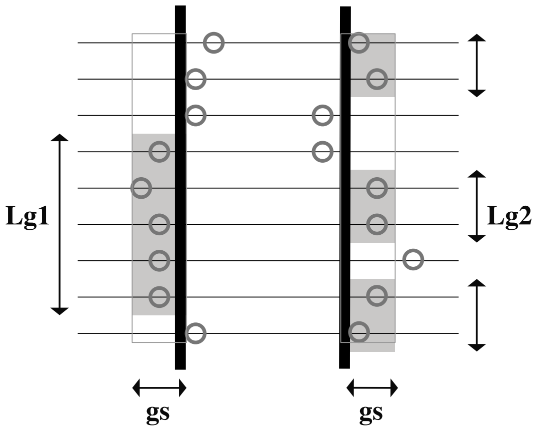
- the first line presents a trivial example: the closest discontinuity is the first one starting from left,
- the second line presents another example where the closest discontinuity is on the right,
- and the third line provides an example of the pathological case where the closest discontinuity corresponds to a hole in the DEM, and thus is ignored.

Gradient length

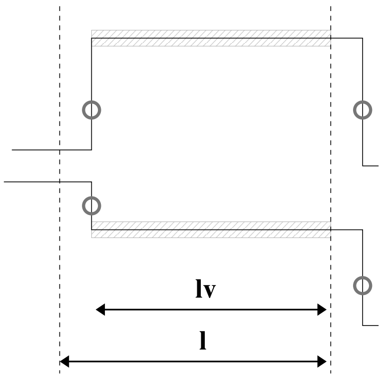
The most important value corresponds to the length of gradient effectively detected: using a parameter sg, we define the length of detected gradient as follow and as illustrated by figure 13b).



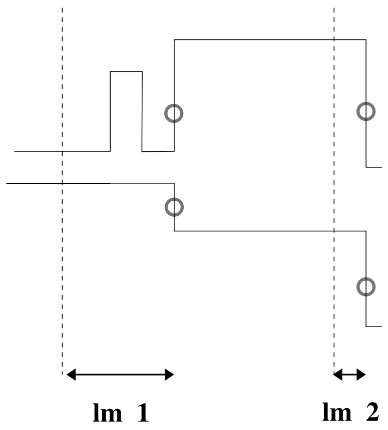
a) Three simplified profiles (continuous lines), location of the sides of the rectangle (dotted lines) and selected discontinuities (circles).



b) Definition of the lateral bands where discontinuities should stand and associated detected discontinuity length.



c) Two profiles and the corresponding volume segments whose length are used to compute the volume rate of a profile.



d) Two profiles and the corresponding moment segments whose length are used to compute the moment rate of a profile.

Figure 13: Illustration on different fictive profiles of the different values of interest.

Definition

`Indicators for each profile:`

$$lg_1(sp) = \mathbf{1}(ri_1 \in [rg_1 - sg, rg_1])$$
$$lg_2(sp) = \mathbf{1}(ri_2 \in [rg_2, rg_2 + sg])$$

`Indicator for both profile:`

$$lg_{and}(sp) = \mathbf{1}\left(ri_1 \in [rg_1 - sg, rg_1] \text{ and } ri_2 \in [rg_2, rg_2 + sg]\right)$$

`Detected gradient length:`

$$\tilde{Lg} = \frac{L}{Ne} * \sum_{k=1}^{Ne} lg_1(sp^k) + lg_2(sp^k)$$

This detected gradient length will be part of the chosen reward function (see section 2.3.4).

However, optimizing such a term over rectangles proved in practice to be hard. We thus add two other values that allow us to order propositions, and ease the optimization.

Volume rate

Figure 13c) shows on two simplified profiles how we define the length lv : as the length of the segment results of the intersection between the interior of the rectangle and the length between the two selected points. The global volume rate given by a rectangle is obtained by averaging volume length over simplified profiles.

Definition

`volume length:`

$$lv = r * (\min(i_2, g_2) - \max(i_1, g_1))$$

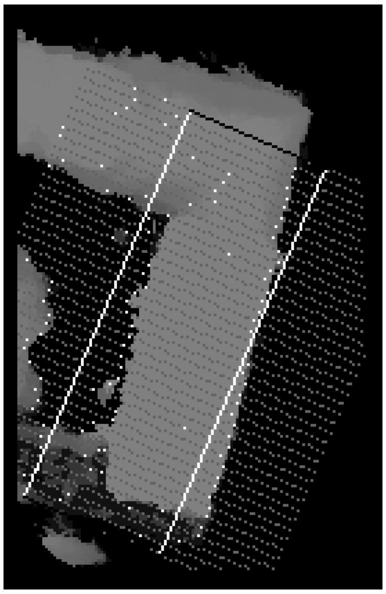
`volume rate:`

$$\tilde{v} = \frac{1}{Ne} \sum_{k=1}^{Ne} \frac{lv(sp^k)}{l}$$

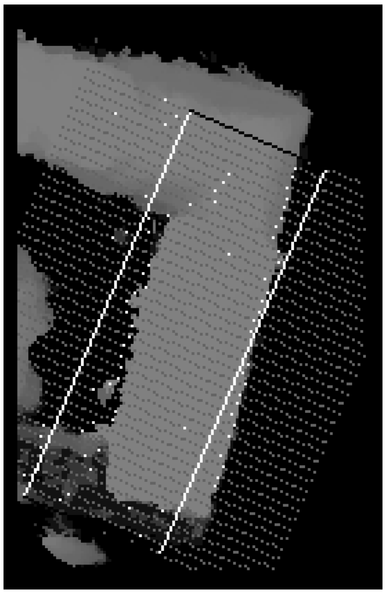
Maximization of the volume rate can be seen as an intermediary step that allows to focus on specific parts of the DEM where buildings should stand. The volume rate is used as a translation potential.

Moment rate

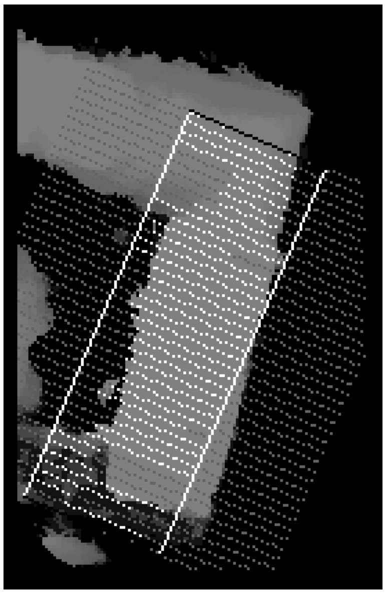
Figure 13d) presents the length used to compute the moment of a rectangle with respect to the DEM.



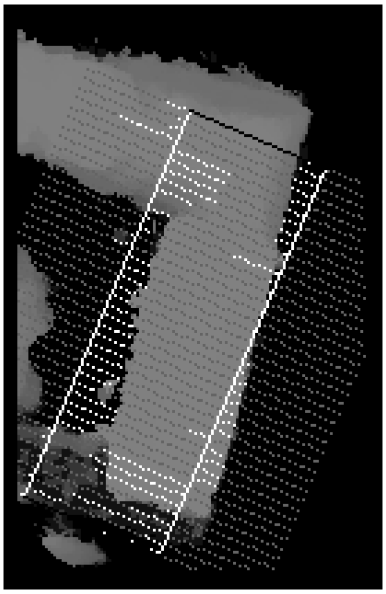
Points selected by the low level filter using $\sigma_h = \sigma_l$ =2m, l_{regul}=2m,r =0.5m and e =1m.



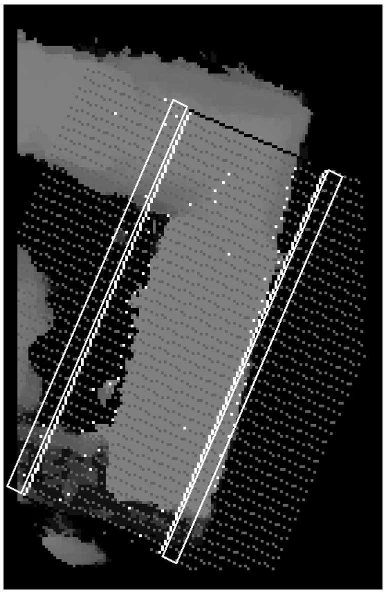
For each profile, couple of points selected among the points of interest. points



Segments used for each profile to compute the volume rate



Segments used for each profile to compute the moment rate



Areas used to compute the length of well positioned discontinuities along a rectangle.

Figure 14: Example of values computed for a given rectangle on real data.

Definition

`moment length:`

$$lm^2 = r^2 * (|i_1 - g_1|^2 + |i_2 - g_2|^2)$$

`moment rate:`

$$\tilde{m} = \frac{1}{Ne} \sum_{k=1}^{Ne} \frac{lm^2(sp^k)}{(l + l_{ext})^2}$$

The moment rate is a value that is minimized. Once a plausible location is found, moment rate minimization allows to put rectangle sides along the edges of the DEM. Of course, moment rate can be seen as a rotation potential.

2.3.4 Thresholding, cost functions and energy

As already stated, $U_d(u)$ will be defined as the detected gradient length of a given rectangle, provided that this length is large enough.

Level 0

As pointed out in section 2.2.2, the sign of function $u_d(.)$ is important. We propose to use a test on the length of detected gradients along the side of a rectangle: a rectangle is attractive if it has enough discontinuities along its sides. Let γ_0 be the set of attractive rectangles:

Definition

`Let` th_1 `and` th_2 `be two parameters living in` $[0,1]$`:`

$$\begin{array}{rcl} \gamma_0 : \; S & \rightarrow & \{0,1\} \\ u & \rightarrow & \begin{cases} 1 \quad \text{if} \begin{cases} \max(Lg_1(u), Lg_2(u)) \geq th_1 * L(u) \\ \text{and} \\ Lg_{and}(u) \geq th_2 * L(u) \end{cases} \\ 0 \quad \text{otherwise} \end{cases} \end{array}$$

A rectangle belonging to γ_0 should have a least a rate of th_1 discontinuity along one of its side and th_2 discontinuity along both sides. We now define the reward function associated to the level 0:

Definition

`Let define` $J_0 \; : \; S \; \rightarrow \; [0,1]$ `reward function of the level 0 as:`

$$J_0(u) = \gamma_0(u) \left[\frac{1}{2} \frac{Lg(u)}{L_{max}} + \frac{1}{2} \frac{l(u)}{l_{max}} \right]$$

$U_d(.)$ will be defined as $-J_0(.)$ on γ_0. Basically the reward function is the detected discontinuity length, except that a width term is added. This term is useful, because of the degrees of freedom given by the width of the localization band sg.

It is very importance to notice that, without any interactions, the energy term essentially corresponds to the length of detected discontinuities.

Levels 1,2 and 3

Since the test associated with level 0 is very restrictive, we decompose $S \setminus \gamma_0$ into other levels. We aim at favoring rectangles 'close' to γ_0.

Definitions

`Level 1:`

$$u \in \gamma_1 \text{ iff } \begin{cases} u \notin \gamma_0 \\ \tilde{v}(u) \geq v_{min} \\ \tilde{m}(u) \leq m_{max} \\ max(Lg_1(u), Lg_2(u)) \geq th_1 \end{cases} \qquad J_1(u) = \frac{1}{10}Lg(u) + \frac{9}{10}(1 - \frac{\tilde{m}(u)}{m_{max}})$$

`Level 2:`

$$u \in \gamma_2 \text{ iff } \begin{cases} u \notin \gamma_1 \\ \tilde{v}(u) \geq v_{min} \text{ or } \tilde{m}(u) \leq m_{max} \end{cases} \qquad J_2(u) = \frac{1}{2}(1 - \tilde{m}(u)) + \frac{1}{2}\tilde{v}(u)$$

`Level 3:`

$$u \in \gamma_3 \text{ iff } u \notin \gamma_2 \qquad J_3(u) = \frac{1}{2}(1 - \tilde{m}(u)) + \frac{1}{2}\tilde{v}(u)$$

These levels were designed such that a gradient descent using for each level the associated reward function and starting from a level 3 object[2] drive it through level 2,1 and finally reaches the level 0.

In section 3.4.3 we present how to use such levels and the convexity property described above in order to guide evolution of a rectangle in the Poisson point process framework. These levels can not be used properly in the data energy U_d.

Final Energy

The energy $U_d(u)$ is defined as:

$$U_d(u) = -a_i - b_i J_i(u) \quad \text{for } u \in \gamma_i \tag{3}$$

Of course, since the only desirable objects are those living in γ_0.

We take $a_0 = 0$ and $b_0 = 1$. Other levels are penalized, as described in section 3.4.3.

It is worth pointing out that with the conventions:

$$t_d(\mathbf{x}) = \sum_{u \in \mathbf{x}} \begin{pmatrix} \mathbf{1}_{\gamma_0}(u) \\ \mathbf{1}_{\gamma_0}(u) J_0(u) \\ \vdots \\ \mathbf{1}_{\gamma_3}(u) \\ \mathbf{1}_{\gamma_3}(u) J_3(u) \end{pmatrix} \qquad \theta_d = \begin{pmatrix} a_0 \\ b_0 \\ \vdots \\ a_3 \\ b_3 \end{pmatrix}$$

the data energy term can be re-written as:

$$U_d(\mathbf{x}) = - < \theta_d, t_d(\mathbf{x}) > \tag{4}$$

2.4 Reference measure

We kept silent the kind of reference measure we use. This measure denoted by ν should be defined over S. Usually a Lebesgue measure is taken. This means that all u are equivalent.

[2]A level 3 object is an object belonging to level 3

Under a homogeneous Poisson process, the expected number of points falling in a subset directly depends on the size of this set. Taking $\nu(.)$ proportional to Lebesgue measure on S: $\lambda|.|_S$, we obtain that under the distribution $\mu(.)$ the expected number of points falling in γ_i is given by:

$$\mathbb{E}_\mu\left[N_{\gamma_i}(X)\right] = \lambda|\gamma_i|$$

In this work, we propose to use a non-homogeneous Poisson point process as the reference process against which $h(.)$ density of the process of interest is defined. Since we optimize the density $h(.)$, the reference measure is theoretically no important.

However, to ease the optimization step, it is suitable to take a non homogeneous reference measure that rewards interesting γ_i's. We will see in section 3 how it is possible to use such an non-homogeneous reference measure.

This non-uniform reference measure was the main reason for introducing levels 1 to 3.

2.5 Prior Model

2.5.1 Goal and constraints

The goal of the prior model is to favor some geometric patterns of objects. We have introduced three kinds of attractive interactions.

Let define an interaction as a symmetric relation between objects. For instance, let be $\sim$ be a relation of proximity, acting on objects of γ_0.

$$\text{for } (u,v) \in S \times S \quad u \sim v \iff \begin{cases} u \in \gamma_0 \\ v \in \gamma_0 \\ d(u,v) \leq d_{max} \end{cases}$$

It is easy to penalize such an interaction, by adding a prior term in the energy:

$$U_{int}(\mathbf{x}) = \sum_{\substack{u,v \in \mathbf{x} \\ u \neq v}} W\mathbf{1}(u \sim v)$$

where $W \geq 0$ is a real parameter. This energy can be rewritten as

$$U_{int}(\mathbf{x}) = Ws(\mathbf{x})$$

$s(\mathbf{x})$ being the number of pairs of interacting points in $\mathbf{x}$. Thus for each added pair, the energy grows of W and the interaction is repulsive. If we put $W < 0$, the interaction is made attractive, but with no limit, and the density $h(.)$ is then not normalisable anymore.

We proposed a general model in [13] to use attractive interactions between points. Here we propose another class of model defined locally.

2.5.2 A general model for attractive interactions

Neighbors

For a given symmetric relation $\sim$, let define by $\mathcal{R}(\mathbf{x})$ the set of interacting pairs of $\mathbf{x}$:

$$\mathcal{R}(\mathbf{x}) = \{\{u,v\} \ : \ u \in \mathbf{x},\ v \in \mathbf{x},\ u \neq v \ \ u \sim v\}$$

We also define the neighborhood $V(\mathbf{x},u)$ of a point u in $\mathbf{x}$ as the set of points in $\mathbf{x}$ in relation with u:

$$u \in \mathbf{x} \;\; V(\mathbf{x},u) = \{v \in x \; : \; u \sim v\}$$

First we consider the mapping I:

$$I(\mathbf{x},u) = \mathbf{1}(V(\mathbf{x},u) \neq \emptyset)$$

$I(\mathbf{x},u)$ is null only if u has no neighbor in $\mathbf{x}$.

Reward function

We now suppose that a mapping $J(.,.)$ from $S \times S$ to $[-1,1]$ is defined on interacting objects. This function should quantify the quality of a pair of interacting points (examples are given later):

$$\begin{array}{rcl} J: \quad S \times S & \rightarrow & [-1,1] \\ (u,v) & \rightarrow & \begin{cases} J(u,v) = J(v,u) & \text{if } u \sim v \\ 0 & \text{otherwise} \end{cases} \end{array}$$

Another value that can be computed over a non empty set $V(\mathbf{x},u)$ is the following maxima $L(\mathbf{x},u)$:

$$L(\mathbf{x},u) = \begin{cases} \max_{v \in x \; v \sim u} J(u,v) & \text{if } V(\mathbf{x},u) \neq \emptyset \\ \\ 0 & \text{otherwise} \end{cases}$$

Local energies

It is now possible to define an energy that is the sum of some local energies describing the state of each object$u \in \mathbf{x}$ with respect to the relation $\sim$

Let define the local energy of an object $u \in \mathbf{x}$ as':

$$u \in \mathbf{x} \qquad U_{loc}^{\sim}(\mathbf{x},u) = -\left(aI^{\sim}(\mathbf{x},u) + bL^{\sim}(\mathbf{x},u)\right)$$

and write the total energy of the configuration $\mathbf{x}$:

$$U_{int}^{\sim}(\mathbf{x}) = \sum_{u \in \mathbf{x}} U_{loc}^{\sim}(\mathbf{x},u)$$

Conclusion

When using several interactions $\sim^1, \dots, \sim^k$, the above described model becomes:

$$\begin{aligned} U_{int}(\mathbf{x}) &= \sum_{i=1}^{i=k} U_{int}^{\sim^i}(\mathbf{x}) \\ &= \sum_{i=1}^{k} \sum_{u \in \mathbf{x}} U_{loc}^{\sim^i}(\mathbf{x},u) \\ &= \sum_{u \in \mathbf{x}} \sum_{i=1}^{k} U_{loc}^{\sim^i}(\mathbf{x},u) \\ &= -\sum_{u \in \mathbf{x}} \sum_{i=1}^{k} [a^{\sim^i} I^{\sim^i}(\mathbf{x},u) + b^{\sim^i} L^{\sim^i}(\mathbf{x},u)] \end{aligned}$$

Expressed this way, it is easy to verify that the global energy of the configuration evolves linearly with the number of points in the configuration. Thus combinatorial problems disappear and balance of the different terms is possible to achieve since they are homogeneous. This is important because since the data energy U_{ext} evolves linearly with the number of points in the configuration, the interaction energy U_{int} should have the same behavior.

Finally, introducing $t_{int}(.)$, mapping from $\mathcal{C}$ to $\mathbb{R}^{2*k}$, and $\theta \in \mathbb{R}^{2*k}$ defined as:

$$t_{int}(\mathbf{x}) = \sum_{u \in \mathbf{x}} \begin{pmatrix} I^{\sim^1}(\mathbf{x}, u) \\ L^{\sim^1}(\mathbf{x}, u) \\ \vdots \\ I^{\sim^i}(\mathbf{x}, u) \\ L^{\sim^i}(\mathbf{x}, u) \\ \vdots \\ I^{\sim^k}(\mathbf{x}, u) \\ L^{\sim^k}(\mathbf{x}, u) \end{pmatrix} \qquad \theta = \begin{pmatrix} a^{\sim^1} \\ b^{\sim^1} \\ \vdots \\ a^{\sim^i} \\ b^{\sim^i} \\ \vdots \\ a^{\sim^k} \\ b^{\sim^k} \end{pmatrix}$$

we obtain the exponential expression:

$$U_{int}(\mathbf{x}) = - < \theta, t_{int}(\mathbf{x}) >$$

2.5.3 Alignment interactions

In a town buildings are usually aligned. Hence, we design an interaction that favors such alignments. However, we need to take care of what is an alignment within our framework. Since our data term detects discontinuities that can be seen as walls, we take into account the two kinds of alignments described by Figure 15.

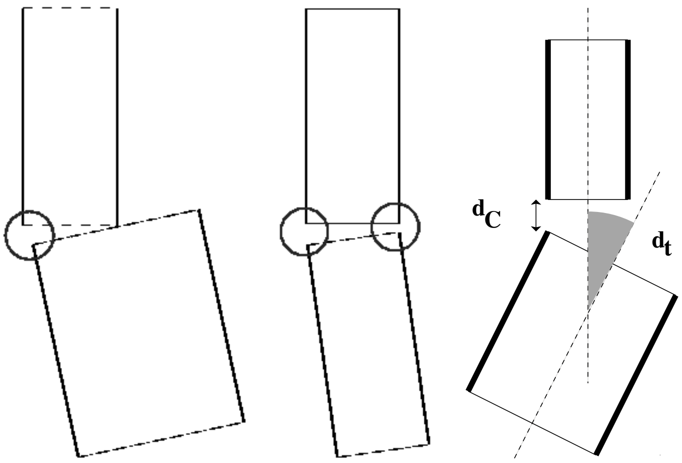


Figure 15: Left: single alignment between two rectangles, middle: double alignment and right: values used to define the interaction.

Denoting $d_{C1}(u, v)$ the distance between appropriate corners and $d_t(u, v)$ the angle difference between the two rectangles (modulo π), we only need two simple conditions to define one interaction:

$$u \sim^{al_1} v \quad \Longleftrightarrow \quad \begin{cases} d_{C1}(u, v) \leq d_{C\,max} \\ d_t(u, v) \leq d_{t\,max} \end{cases}$$

Since a rectangle has four corners, we define an alignment relation for each of them, and this gives how relations ^{al_1} to ^{al_4} are defined. An object can be aligned with another with respect to two relations. Figure 16 shows an example of a rectangle that is related to three other ones under the four relations.

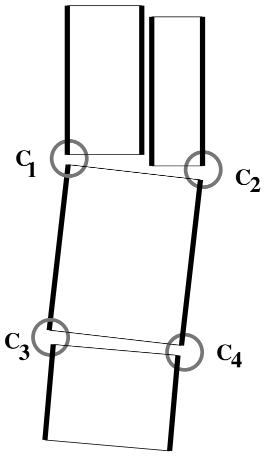


Figure 16: Example of a rectangle that is in relation with three other ones with respect to the 4 possible relations of alignment.

To define the reward function associated with the alignment relationship, we first introduce the following real valued function $\sigma(.)$:

$$\begin{array}{rlcl} \sigma : & \mathbb{R}^2 & \rightarrow & [0,1] \\ & (x, x_{max}) & \rightarrow & \frac{1}{x_{max}^2}\left(\frac{1+x_{max}^2}{1+x^2} - 1\right) \end{array}$$

σ was designed in [13] such that $\sigma(0, x_{max}) = 1$ and $\sigma(x_{max}, x_{max}) = 0$. The associated reward function evaluated on a pair of related points is:

$$J^{al_1}(u,v) = \frac{1}{2}\sigma(d_t(u,v), d_{t\,max}) + \frac{1}{2}\sigma(d_{C1}(u,v), d_{C\,max})$$

This reward function is important: the goal is not only to favor presence of alignments, but also to optimize the quality of alignments once they are detected.

2.5.4 Completion interactions

Simple buildings are usually made of four sides. Since our data term only detects two discontinuities, it seems useful to add a completion relationship.

We use the same type of conditions as in section 2.5.3 : related objects should both belong to γ_3, distance between suitable corner should be less than d_{C_max}, and the difference between angles should be less than $d_{t\,max}$ modulo $\pi/2$. Figure 17 presents examples of related rectangles. Once again, we actually define fout different relations, each of them being related to one of the corners.

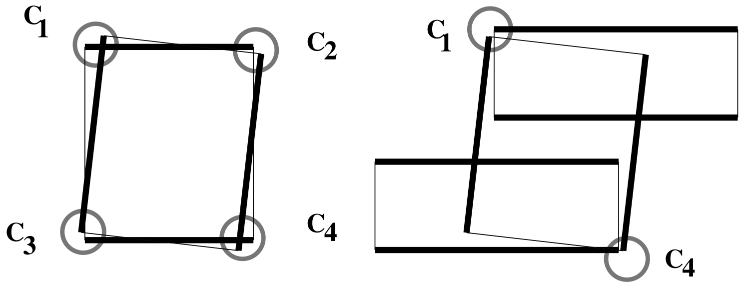


Figure 17: Illustrations of a completion relation

2.5.5 Paving interactions

The last relation we introduce favors parallel rectangles which are located side by side. Figure 18 shows what it means. The definition of the relation is similar to the previously defined ones. The function J is also defined in a similar way.

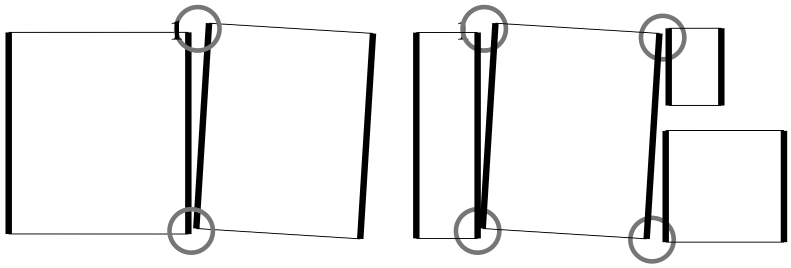


Figure 18: Illustrations of the paving relation.

2.5.6 Visual results

We present on figure 19 two results. The first one is a realization of point process whose density is defined by $U_{int}(.)/T$ with a medium temperature. The second one is a configuration minimizing $U_{int}(.)$ obtained by simulated annealing. These results show how our internal field constraint object positioning.

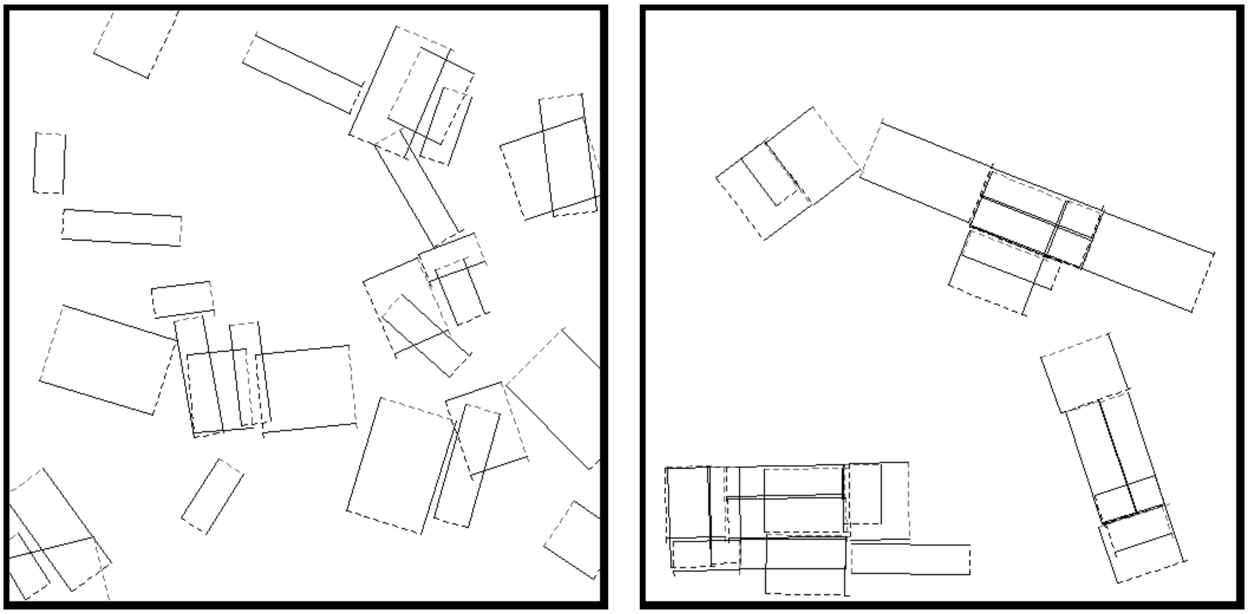


Figure 19: Illustration of the influence of the internal field: left hand side at medium temperature, right hand side at low temperature.

2.6 Exclusion Interaction

2.6.1 Definition

An exclusion term that avoid redundant objects is needed, because of 3 main reasons:

- we need to avoid redundant explanations of the data,

- we need to insure that the attractive interactions will not be too strong,
- a condition used to show the convergence of the algorithm requires that the variation of the energy, induced by adding a point to a given configuration, is bounded.

The simplest exclusion interaction we can use is the following intersection relation that acts only on rectangles which are parallel:

$$u \sim^{excl} v \text{ iff } \left\{ \begin{array}{c} d_t(u,v) \leq d_{tmax_{excl}} \\ Surf(u) \cap Surf(v) \neq \emptyset \end{array} \right.$$

This exclusion interaction has the following property with respect to the attractive interactions we previously defined:

For any attractive interaction i,

$$\forall (u, \mathbf{x}) \;\; \text{card} \left\{ v \in \mathbf{x} \text{ s.t. } \left\{ \begin{array}{cc} & u \sim^i v \\ \nexists w \in \mathbf{x} & v \sim^{excl} w \end{array} \right. \right\} \leq N_{over} \tag{5}$$

This result states that if u has N_{over} neighbors (with respect to one of the attractive relations $\sim^i$), it is impossible to add a new neighbor that does not intersect with one of the formers.

This property is easily verified, since the number of non overlapping rectangles bigger than $l_{min} * L_{min}$ is limited because the surface of interest (K) is finite.

2.6.2 Associated energy

We use a simple model:

$$U_{excl}(\mathbf{x}) = -a_{excl} \sum_{u \in \mathbf{x}} I^{\sim_{excl}}(\mathbf{x}, u) \qquad a_{excl} < 0$$

a_{excl} is taken small enough, such that it is impossible to have redundant objects. This is called a hard core term.

It is possible to choose more relevant terms, that avoid to penalize too strongly intersecting rectangles.

For instance, one could imaging making the interior part of the energy of an object be null if it intersects another one. The hard core term nevertheless proved to be sufficient with respect to the quality of the results obtained while being more 'relevant' would require more computations for each iteration.

Once again, this term can be written under the exponential form as:

$$U_{excl} = - < \theta_{excl}, t_{excl}(\mathbf{x}) > \quad t_{excl}(\mathbf{x}) = \sum_{u \in x} I^{\sim_{excl}}(\mathbf{x}, u)$$

3 Algorithm

In this section, we present the algorithm used to optimize the energy defined in the previous section. A general presentation and proofs of convergence have been given in [16].

We have defined a point process X by its energy $U(.)$. This energy leads to a density h :

$$h(\mathbf{x}) = \frac{1}{Z} \exp(-U(\mathbf{x}))$$

where $\frac{1}{Z}$ is an unknown normalizing constant depending on the reference measure $\mu(.)$.

3.1 Generalities

3.1.1 MCMC for point process

Suppose we want to sample from $h(.)$ (i.e. to obtain some configurations of rectangles distributed according to $h(.)$). A solution is to use a Monte Carlo Markov Chain (MCMC) method. The idea is to build a Markov Chain $(X_t)_{t\geq 0}$ on $\mathcal{C}$, space of finite configurations of rectangles using :

- a starting point : in our case, the empty configuration $X_0 = \{\emptyset\}$,
- a Markovian transition kernel $K(\mathbf{x}, .)$ giving the distribution of $X_{t+1}|X_t = \mathbf{x}$.

Being given these two elements, it is possible to generate one (or more) trajectories of the Markov Chain. Of course, $K(.,.)$ is designed in order to make the Markov Chain converge ergodically to the desired distribution.

Ergodic convergence means that the final elements of any trajectory are distributed according to $\pi(.)$, defined by the density $h(.)$ and the reference measure $\mu(.)$:

$$\|K^n(\{\emptyset\}, .) - \pi(.)\|_{TV} \to 0$$

The Markov Chain generated by the following algorithm verifies this property. We actually have more accurate results, since we know that we can start from any configuration (Harris recurrence) and that the total variation tends to zero with a geometrical rate (geometric ergodicity), as proved in [16].

3.1.2 Geyer and Møller algorithm

The sampler we use is based on Geyer and Møller work (see [6]). It also uses the work of Green on Metropolis Hastings sampler on general state space (see [8]).

We suppose we want to sample a point process defined by its density $h(.)$ known up to a normalizing constant and a reference Poisson point process with intensity measure $\nu(.)$ on S.

The algorithm is based on Metropolis Hastings method. This technique uses some proposition kernels $Q_m(\mathbf{x}, .)$ that generate a new configuration $\mathbf{y}$ from an old configuration $\mathbf{x}$, and an acceptance rate $\alpha(\mathbf{x}, \mathbf{y})$ which tells if a proposed perturbed configuration $\mathbf{y}$ should replace $\mathbf{x}$ or not.

Let denote by $\mathbf{x} = \{x_1, \ldots, x_{n(\mathbf{x})}\}$ a configuration of points. We consider 3 kind of proposition kernels :

Birth or death : let denote by $Q_{BD}(\mathbf{x}, .)$ this kind of perturbation. It works as follows : first choose with probability 0.5 if a point should be removed, or if a point should be added to the current configuration.

- if **death** has been chosen, chose randomly one point u in $\mathbf{x}$ and propose $\mathbf{y} = \mathbf{x} \setminus u$,
- if **birth** has been chosen, generate a new point u **according to the intensity measure** $\nu(.)/\nu(S)$ and propose $\mathbf{y} = \mathbf{x} \cup u$.

Non jumping transformations : we denote by $Q_{NJ}(.,.)$ these kinds of transformations. Such transformations randomly select one point u in the current configuration $\mathbf{x}$, perturb it to obtain a new point v and propose $\mathbf{y} = \mathbf{x} \setminus u \cup v$. We present later all the perturbations used.

Birth of death in a neighborhood : we introduced this kind of transformation in [16]. The idea is to propose to remove or add interacting pairs of points, the interaction being one of the previously defined relations.

For each of these proposition kernels, a mapping $R(.,.)$ from $\mathcal{C} \times \mathcal{C}$ to $(0, \infty)$ is associated (see [16] for details on how to compute this value, which we call **Green's ratio**).

Algorithm A

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

1. *choose one of the previously described proposition kernels $Q_m(.,.)$ with probability $p_m(\mathbf{x})$,*
2. *sample $\mathbf{y}$ according to the chosen kernel : $\mathbf{y} \sim Q_m(\mathbf{x},.)$,*
3. *compute the associated Green's ratio $R_m(\mathbf{x},\mathbf{y})$ and the acceptance rate $\alpha_m(\mathbf{x},\mathbf{y})$,*
4. *accept the proposition $X_{t+1} = \mathbf{y}$ with probability $\alpha_m(\mathbf{x},\mathbf{y})$, and reject it otherwise.*

3.2 Reversibility and Green's ratio

The main point is that the designed Markov Chain converges to the desired distribution. Algorithm A actually build a $\pi(.)$ reversible Markov Chain. This comes from the expression of Green's ratio $R(\mathbf{x},\mathbf{y})$.

If we assume that the proposition kernel $Q(\mathbf{x},.)$ can be written as a mixture of distribution:

$$Q(\mathbf{x},.) = \sum_m p_m(\mathbf{x}) Q_m(\mathbf{x},.)$$

then, if y is obtained from $\mathbf{x}$ using $Q_m(\mathbf{x},.)$, the expression of the Green's ratio can be formally denoted by :

$$\text{if } \mathbf{y} \sim Q_m(\mathbf{x},.) \text{ then } R_m(\mathbf{x},\mathbf{y}) = \frac{\pi(d\mathbf{y}) Q_m(\mathbf{y}, d\mathbf{x})}{\pi(d\mathbf{x}) Q_m(\mathbf{x}.d\mathbf{y})}$$

where the division corresponds to a Radon Nikodym derivation. Such a notation supposes that if $\mathbf{y} \sim Q_m(\mathbf{x},.)$, then $\mathbf{x}$ can be obtained from $\mathbf{y}$ through $Q_m(\mathbf{y},.)$. For instance, a translation perturbation that can only translate a point to its left is not convenient, since it does not allows the 'coming back' move.

3.3 Simulated annealing

Our goal is to find a minimizer of energy $U(.)$, and to obtain it we use a simulated annealing.

Instead of simulating according to $h(.)$, we simulate according to $h^{\frac{1}{T_t}}(.)$. T_t is a temperature parameter and tends to zero as t tends to ∞.

These techniques have widely been used in image processing (see [24] for instance). If T_t decreases with a logarithmic rate, then X_t tends to one of the global maximizer of $h(.)$.

Of course, in practice, it is not possible to use a logarithmic decrease. Thus we use a geometrical one.

This last point makes the quality of the proposition kernels be important. Actually, a low correlated trajectory is much better since it insures that the space has been widely enough explored.

All that follows aims at building a trajectory that is not too much correlated.

3.4 Reference measure

3.4.1 Comments

Suppose we are given a reference non atomic measure $\nu(.)$ on S. Let define $\mu_\nu(.)$ the distribution of the Poisson point process whose intensity measure is $\nu(.)$.

The maximizer of $h(.)$ does not depend on the chosen reference measure. Usually, this framework is used with a Lebesgue measure on S, which makes the birth of a point easy. A normal distribution or any practically usable distribution may also be interesting.

However, at high temperatures, $h(.)^{\frac{1}{T}}$ is approximatively a uniform distribution. Hence, only the Poisson part given by $\mu_\nu(.)$ is important. Using a uniform Poisson point process is problematic, because points of interest are those living in γ_0 which is of small Lebesgue measure.

A solution is to use another reference measure that favors points of γ_0. However, since the birth or death kernel is required to generate new points according to $\nu(.)$ this it not possible since the γ_i sets are not known. That is why the following development is used.

3.4.2 Analysis

Let $p : \mathcal{C} \to [0, \infty)$ be a non negative measurable function on the family $\mathcal{C}$ of finite configurations of points such that:

$$\int_{\mathcal{C}} p(\mathbf{x}) d\mu(x) = 1 \tag{6}$$

$p(.)$ is a probability density and defines a point process X on S. Now, to explain the relation between $p(.)$ and $\nu(.)$, let decompose $\mathcal{C}$ into

$$\mathcal{C} = \sum_{n=0}^{\infty} C_n$$

C_n being for a given n the family of configurations of n points of S.

Thus, to describe $p(.)$, we can look at the distribution of total number of points $(p_n)_{n\geq 0}$ and at the conditional distributions $j_n(x_1, \ldots, x_n)$ of n random points $x_1, \ldots x_n$. According to [20] (see chapter 1, page 27), we obtain:

$$p_n = \frac{e^{-\nu(S)}}{n} \int_S \cdots \int_S p(\{x_1, \ldots, x_n\}) d\nu(x_1) \ldots d\nu(x_n) \tag{7}$$

and

$$j_n(x_1, \ldots, x_n) = \frac{p(\{x_1, \ldots, \mathbf{x}_n)}{\displaystyle\int_S \cdots \int_S p(\{x_1, \ldots, x_n\}) d\nu(x_1) \ldots d_\nu(x_n)} \quad \text{w.r.t. } (\nu^n)(., \ldots, .) \tag{8}$$

Now, let consider the following density defined with respect to the intensity measure $\nu(.)$:

$$p(\mathbf{x}) = \alpha \Pi_{u \in \mathbf{x}} \beta(u) p'(x)$$

where $\beta : \mathcal{C} \to (0, \infty)$ is a measurable function.

Let define $\nu'(.)$ as follows:

$$A \subseteq S \quad \nu'(A) = \int_S \mathbf{1}_A(u) \beta(u) d\nu(u)$$

We immediately obtain that (p'_n) defined by p' and ν' is proportional to (p_n) defined by p and ν. The same kind of result holds for the conditional distributions $j(\ldots)$.

It is thus equivalent to use p and ν or p' and ν'.

3.4.3 Favoring γ_0

We use $h^{\frac{1}{T_t}}$ as p' , and for $\nu()$, we take the product between Lebesgue measure on K (measuring squared meters) and the uniform measure on M, space of marks:

$$\nu(.) = |.|_K \times \frac{|.|_M}{|M|}$$

Let introduce the following β function:

$$\beta(u) = e^{\sum_{i=0}^{3} W_i \mathbf{1}_{\gamma_i(u)}}$$

The W_i are the weights of the γ_i sets. For a high temperature, X is almost distributed according to a Poisson point process distribution that verifies:

$$E_i = \mathbb{E}[N_{\gamma_i}(X)] = W_i \nu(\gamma_i) = W_i \frac{|\gamma_i|_{\mathbb{R}^5}}{|M|_{\mathbb{R}^3}}$$

We can tune the weights such that the E_i are of same orders , making all γ_i be approximatively equivalent. The expected total number of points $\sum W_i |\gamma - i|/|M|$ was tuned to approximatively reflect the density of buildings in a dense urban area.

The main advantage of this parameterization is that it is independent from the area observed K, since $\nu(.)$ goes proportionally with its surface.
Using $\nu(.)$ as the reference measure, the density $h'(.)$ we need to sample from is given by:

$$\begin{aligned} p(\mathbf{x}) = h'(\mathbf{x}) &= \frac{1}{Z'} \Pi_{u \in \mathbf{x}} \beta(u)\ h^{\frac{1}{T_t}}(\mathbf{x}) \\ &= \frac{1}{Z} \exp\left(\sum_{i=0}^{3} W_i \mathbf{1}_{\gamma_i}(u) - \frac{U(\mathbf{x})}{T_t} \right) \end{aligned}$$

using the following notation:

$$\theta_{mes} = \begin{pmatrix} W_0 \\ 0 \\ W_1 \\ 0 \\ W_2 \\ 0 \\ W_3 \\ 0 \end{pmatrix}$$

we obtain that:

$$\boxed{p_t(\mathbf{x}) \propto \exp\left(< \theta_{mes}, t_d(\mathbf{x}) > + \frac{< \theta_d, t_d(\mathbf{x}) > + < \theta_{int}, t_{int}(\mathbf{x}) > + < \theta_{excl}, t_{excl}(\mathbf{x}) >}{T_t} \right)} \qquad (9)$$

3.5 Non jumping transformations

Non jumping transformations are transformations that:

- first, select randomly a point u in the current configuration,
- second, perturb this point to obtain a new version v,
- third, propose to replace u by v: $\mathbf{y} = \mathbf{x} \setminus u \cup v$.

3.5.1 Translation, rotation, dilations

As detailed in [16], we propose to use the transformations described by Figure 20 to perturb some points.

Each of these transformations uses a parameter z that is randomly chosen in some set Σ. For instance, as shown by Figure 21, rotation uses a $z \in \Sigma = [-\Delta\varphi, \Delta\varphi]$ to generate the new angle of the selected object.

Under the following conditions:

- u is chosen uniformly in $\mathbf{x}$,
- the distribution of Z giving z is symmetric over the symmetric space $\Sigma = [-\Delta\varphi, \Delta\varphi]$

the suitable ratio $R(\mathbf{x}, \mathbf{y})$ is given by:

$$R(\mathbf{x}, \mathbf{y}) = \frac{p(\mathbf{x})}{p(\mathbf{y})}$$

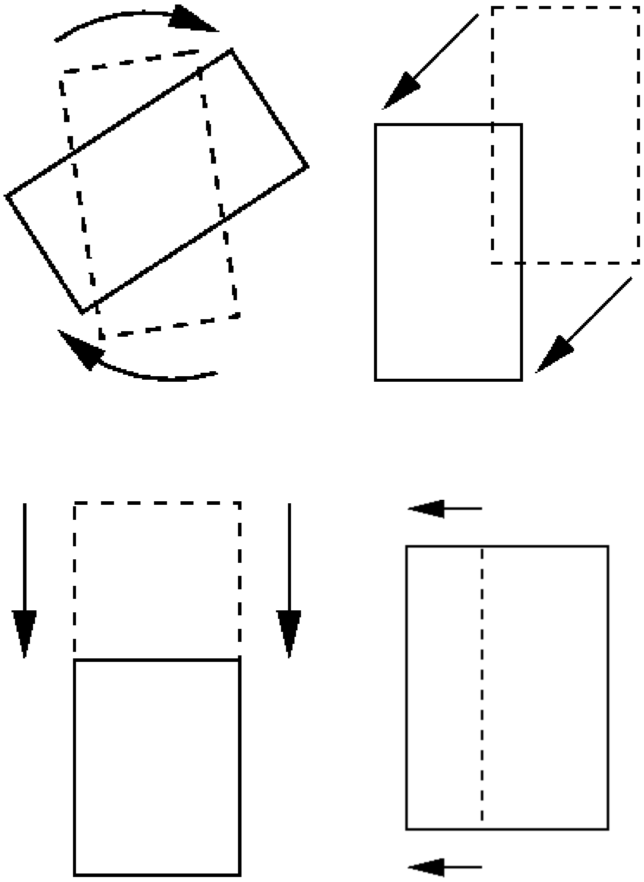


Figure 20: Simple non jumping transformations.

3.5.2 Gibbs like transformation

We also implemented some Gibbs like version of the latter transformations. Details are provided in [16].

The idea is that instead of using a uniformly generated z, it is possible to pre-compute a distribution depending on the target distribution $h(.)$. In [16], we called this kind of transformations 'pre-explorative transformations'.

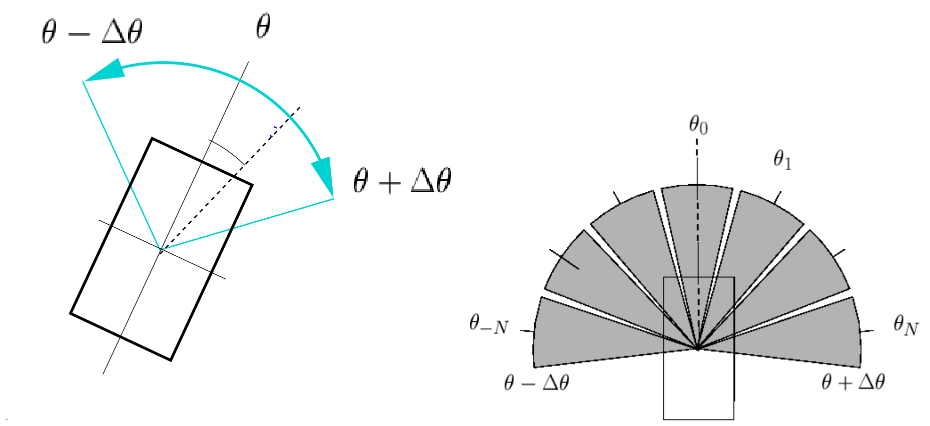


Figure 21: Rotation and pre-explorative rotation scheme.

The advantage of such transformations is that they fit the target distribution by exploring it. As detailed in [16], Green's ratio of these kind of transformation is close to 1:

$$R(\mathbf{x}, \mathbf{y}) \approx 1$$

This comes from the fact that the random variable Z which is used to generate the new object v is not symmetrically distributed. Its distribution follows an estimated approximation of $\pi(\mathbf{x} \setminus u \cup v(z))$.

The **exact** expression of $R(\mathbf{x}, \mathbf{y})$ that insures π-reversibility of the Markov Chain is detailed in [16].

3.6 Birth or death in neighborhoods

The last kind of transformations we use corresponds to the relationships we have defined in our model.

We have defined some positive interactions we favor. It is relevant to favor also the exploration ability of the algorithm in the parts of the space where such alignments occur.

Thus, for a given relevant rectangle (i.e. living in γ_0) we propose to add a new rectangle in its neighborhood.

3.6.1 Birth or death of an aligned point

The first attractive interaction we have defined is the alignment one. The corresponding kernel proposes either to create an alignment or to remove one.

This proposition kernel can choose between two perturbations with probability 0.5 for each:

birth:

1. randomly select a point u of $\mathbf{x} \cap \gamma_0$,
2. generate a new point v aligned with u in the sense defined by $\sim^1$, $\sim^2$, $\sim^4$ or $\sim^4$,
3. propose $\mathbf{y} = \mathbf{x} \cup v$.

death:

1. select a pair of aligned points, provided that at least one of them is in γ_0,
2. choose an object v in this pair with probability 0.5,
3. propose to remove v: $\mathbf{y} = \mathbf{x} \setminus v$.

Reversibility is insured by taking into account two phenomena:

- creating one alignment can generate several ones,
- a building created in the neighborhood of another one may not lie in γ_0.

The expression of the Green's ratio associated to this kind of transformations is detailed in annex B.

3.6.2 Other birth or death in a neighborhood

This type of updating methods has been implemented for each type of relationships considered in this report. Thus we take into account the 12 cases corresponding to all of the attractive relations of alignment, paving interaction and completion.

3.7 Convergence of the algorithm

In [16] we expressed some sufficient conditions under which the algorithm converges. Here we detail this set of conditions and show that the proposed model verifies it.

3.7.1 Stability condition on the density

The first condition is a condition imposed over the density we want to sample from. We need to find a R_g such that:

$$\forall(\mathbf{x}, u) \in \mathcal{C} \times S \qquad p(\mathbf{x} \cup u) \leq R_g p(\mathbf{x}) \tag{10}$$

This condition means that adding a point to a configuration should not decrease to much its energy. One of the consequence of this condition is to insure the integrability of the density $p_t(\mathbf{x})$, since it can be bounded by $p_t(0) R_g^{n(\mathbf{x})}$ which is of the form of an intensity term.

Term involving t_d:

It is important to notice that:

$$t_d(\mathbf{x} \cup u) - t_d(\mathbf{x}) = t_d(u) \quad \text{with} \quad \|t_d(u)\| \leq 1$$

This leads immediately to:

$$| < \theta_{mes} + \frac{\theta_d}{T}, t_d(\mathbf{x} \cup u) - t_d(\mathbf{x}) > | \leq \|\theta_{mes}\| + \frac{\|\theta_d\|}{T} \tag{11}$$

Prior model

The stability condition holds also with respect to the prior model, but this property relies on the fact that the exclusion relation penalizes superpositions of rectangles.

Let take the practical example of an attractive relation $\sim^i$. Recalling that $a_{excl} < 0$, we suppose that:

$$a_{excl} < -(a^{\sim^i} + b^{\sim^i}) \tag{12}$$

The induced energy variation when adding a point to a configuration is:

$$U^{al\sim_1}(\mathbf{x} \cup u) - U^{al\sim_1}(\mathbf{x}) = U_{loc}(\mathbf{x} \cup u, u) + \sum_{v \in \mathbf{x}} U_{loc}(\mathbf{x} \cup u, v) - U_{loc}(x, v)$$

where U_{loc} stands for:

$$U_{loc}(\mathbf{x}, v) = -[a^{\sim^i} I^{\sim^i}(\mathbf{x}, v) + b^{\sim^i} L^{\sim^i}(\mathbf{x}, v) - (a^{\sim^i} + b^{\sim^i}) I^{\sim^{excl}}(\mathbf{x}, v))] \tag{13}$$

Thus the negative part of the energy variation corresponds to:

- the part related to $U_{loc}(\mathbf{x} \cup u, u)$ which is bounded by $|a^{\sim^i}| + |b^{\sim^i}|$,
- the variation induced by u over each v belonging to u. Condition (12) gives that the negative part of the energy only involves the objects in x that are in attractive interaction $\sim^i$ but not in exclusive relation $\sim^{excl}$ with u. We have seen that the number of such objects is uniformly bounded by N_{over} and finally:

$$\begin{aligned}
\frac{e^{-U^{al\sim_1}(\mathbf{x} \cup u)}}{e^{-U^{al\sim_1}(\mathbf{x})}} &= \exp\left(-|U^{al\sim_1}(\mathbf{x} \cup u) - U^{al\sim_1}(\mathbf{x})|_+ + |U^{al\sim_1}(\mathbf{x} \cup u) - U^{al\sim_1}(\mathbf{x})|_-\right) \\
&\leq \exp\left(|U^{al\sim_1}(\mathbf{x} \cup u) - U^{al\sim_1}(\mathbf{x})|_-\right) \\
&\leq \exp\left(|a^{\sim^i}| + |b^{\sim^i}| + N_{over}(|a^{\sim^i}| + |b^{\sim^i}|)\right) \\
&\leq \exp\left((N_{over} + 1)(|a^{\sim^i}| + |b^{\sim^i}|)\right)
\end{aligned}$$

Now if we want to be able to sum this result over all attractive relations i we need the following condition:

$$a_{excl} < -\sum_{i=1}^{k} (a^{\sim^i} + b^{\sim^i})$$

and finally the stability condition is proved.

3.7.2 Stability conditions on the sampler

The three next sets of conditions we need concerns the sampler:

1. for each perturbation kernel, the probability of choosing one kind of transformation $p_m(\mathbf{x})$ does not depend on the state $X_t = \mathbf{x}$, and the probability of proposing to do nothing is strictly positive,
2. some bounds are needed on distributions used by birth or death proposition kernels (these distributions are detailed in annex B),
3. and finally, for each relation used to build a birth or death in a neighborhood kernel, the following condition holds:
$$\exists r^{\sim^i} \quad \forall u \in S \quad |\{v \in S \;\; v \sim^i u\}|_S > r^{\sim^i}$$

In this framework, all of these conditions hold.

3.7.3 Convergence property

The fact that these conditions hold give the convergence property for the sampler at a fixed temperature.

proposition *For a fixed temperature $T_t = T$, the Algorithm A (see section 3.1) builds a Markov Chain $(X_t)_{t \geq 0}$ that verify the following properties:*

- *it ergodically converges to the distribution defined by $p_t(\mathbf{x})$ and the uniform Poisson point process with intensity measure $\nu(.)$, equivalent to the distribution defined by $p'_t(\mathbf{x})$ and the non homogeneous Poisson point process with intensity measure $\nu'(.)$,*
- *this convergence holds for any starting point X_0 and is achieved with a geometrical rate.*

The proof is given in [16].

Because we have this convergence of the sampler, we can state on the simulated annealing: making the temperature decrease with a logarithmic rate gives a realization $\mathbf{x}$ that belong to the set of global maxima of density $h(.)$ (see [19] for a proof, using Dobrushin's conditions).

4 Results

We show here some practical results obtained by the proposed method on optical and Laser data. The starting configuration is the empty set.

Both the sampler and the model need a huge number of parameters to be tuned. We first focus on the parameters we used. Then, we present a validation of the results.

4.1 Parameters

4.1.1 Size of the space

The dimension of K, 2D space is given by the data and its resolution and corresponds to the true dimension of the area in squared meters.

For the space of marks, we consider the following limits :

l_{min}	4
l_{max}	40m
L_{min}	6m
L_{max}	40m

Table 1: Dimensions of the space of marks

4.1.2 Definition of the internal field

The relations of alignment, paving and completion require two physical parameters each : the maximum accepted difference of angle d_{tmax}, and the maximum distance between the two considered corners of the two rectangles.

Attractive relations	
d_{tmax}	20°
d_{Cmax}	4m

Intersection	
d_{tmax}	60°

Table 2: Definition of the internal field

4.1.3 Definition of γ_i sets for the data term

The data term relies on some thresholds. Of first importance are the levels defining what is γ_0, th_1 and $th2$.

Minimum length of detected gradient	
$th1$	80 %
$th2$	40%

Thresholds for other levels	
v_{min}	90%
m_{max}	0.02

Table 3: Definition of data levels

4.1.4 Model parameters of the data term

The optimized density uses the following parameters (of course γ_0 is favored, while other levels are penalized).

Level 0	
a_0	0
b_0	1

Other Levels	
a_1	-0.001
b_1	0.001
a_2	-0.01
b_2	0.01
a_3	-0.1
b_3	0.1

Table 4: Definition of data levels

4.1.5 Model parameters of the internal field

In this section we present the model parameters θ_{int} and θ_{excl}. The latter was chosen according to the condition 5. θ_{int} was tuned by hand. A procedure using the ground truth could be implemented to estimate those parameters.

The prior model is quite robust. The relative weights of attractive interactions can be constant as follows. Only the relative weight between the internal field and the external one may be re-tuned.

Alignments	
$a^{\ al}$	0.04
$b^{\ al}$	0.1

Relations of Completion	
$a^{\ comp}$	0.04
$b^{\ comp}$	0.1

Paving Relations	
$a^{\ pav}$	0
$b^{\ pav}$	0.5

Relation of Exclusion	
$a^{\ excl}$	-3

Table 5: Parameters of the prior model

4.1.6 Parameters of the reference measure

Depending on the kind of data used, the parameters involved in θ_{mes} are estimated so that the expected number of points falling in γ_0 is 100 for a square of $100m$ by $100m$ and that :

$$\mathbb{E}_{mes}[N_{\gamma_3}(X)] \approx \mathbb{E}_{mes}[N_{\gamma_2}(X)] \approx \mathbb{E}_{mes}[N_{\gamma_1}(X)] \approx \mathbb{E}_{mes}[N_{\gamma_0}(X)] \approx \frac{|K|}{100}$$

For a given type of data (LASER or optical DEM, for instance), the weights W_i are estimated once for all using a first run of the Markov Chain at very high temperature (so that X is distributed according a distribution close from a Poisson point process one) that allows to estimate $|\gamma_i \cap S|$.

When changing the observed area but keeping the same kind of data, it is not necessary to re-estimate (W_i). However, if the type of data is changed, or the if the thresholds involved in the data term are re-tuned, a re-estimation of (W_i) should be done.

4.1.7 Parameters of the sampler

There are mainly two types of parameters : the mixture parameters (p_m) and the size of the spaces used to generate new components.

For birth or death perturbation kernels, the probability are chosen to be equal and $p_d = p_b = 0.5$.

Here we present the chosen mixture parameters for a basic sampler. These parameters could be estimated in order to be optimal with respect to the sampler efficiency.

Birth or death	
p_{BD}	1/20

Birth or death with respect to $\sim^{al}$	
$p_{BDN_{al}}$	5/20

Birth or death with respect to $\sim^{pav}$	
$p_{BDN_{pav}}$	1/20

Birth or deathwith respect to $\sim^{comp}$	
$p_{BDN_{comp}}$	1/20

Non jumping transformation	
p_T	3/20
p_R	3/20
p_{DL}	3/20
p_{Dl}	3/20

Table 6: Mixture parameters for a basic sampler

4.2 Validation of the results

4.2.1 Area of interest

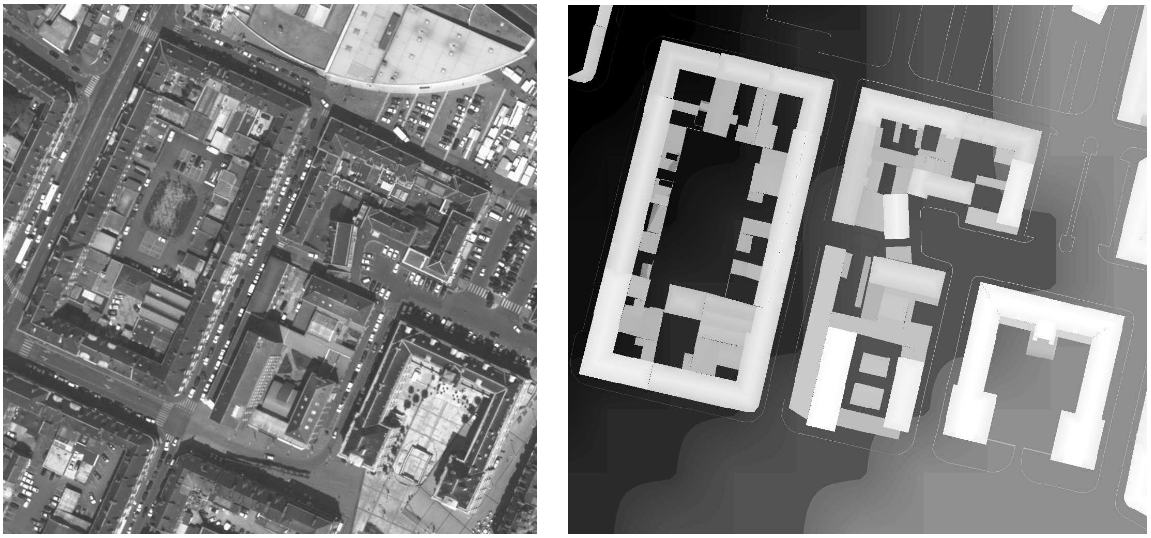
We have faced hard problems when trying to validate the results we obtained. We have focused on an area described on the aerial photography in show Figure 22. This figure also presents the ground truth used to validate our results.

This ground truth consists in a very precise DEM built semi-automatically by IGN using precise aerial optical data. The problems we face using this ground truth are :

- some buildings are missing,
- in some places the shape on the ground truth is too precise with respect to the data used,
- it is hard to compare two vectorial representations of the same urban area.

4.2.2 Hausdorff Distance

For a given result (i.e. an estimated configuration of rectangle $\mathbf{x}$), it is possible to compute the average surface of missed buildings and the average surface of over-detected buildings with respect to the ground truth.



a) b)

Figure 22: First zone of interest : a) aerial photography of Amiens (25cm) and b) ground truth (20cm) provided by IGN .

4.3 First result : LASER data

We first present an example on LASER data. This LASER data are at a resolution of 0.5m per pixel and of 0.01m for the gray levels.

The main problem encountered with LASER data is a problem of smoothness. Such data are built using an aircraft and several flights over the area. Measured points are not equidistant. Thus some areas are precisely covered while some others are not, and the reconstruction is obtained using some interpolation between measured points.

It makes the profile be a lot smoother as shown in Annex A, and thus low level filter parameters are tuned in order to deal with smooth discontinuities.

4.3.1 Data parameters

The results we present in this section have been obtained using the following low level parameters.

Grid of points	
r	0.5m
e	1m
l_{ext}	10m

Filter	
l_{regul}	2m
σ_l	0.7*r
σ_h	1.5m

Table 7: Low level parameters used for LASER data

4.3.2 Estimated configuration

Figure 23 shows the estimated configuration, the ground truth and the associated land register map which has been estimated.

For each detected rectangle, the mean elevation on the LASER data has been computed.

The estimated land register is obtained by associating, to each rectangle, the mean elevation on an image at 20cm of resolution.

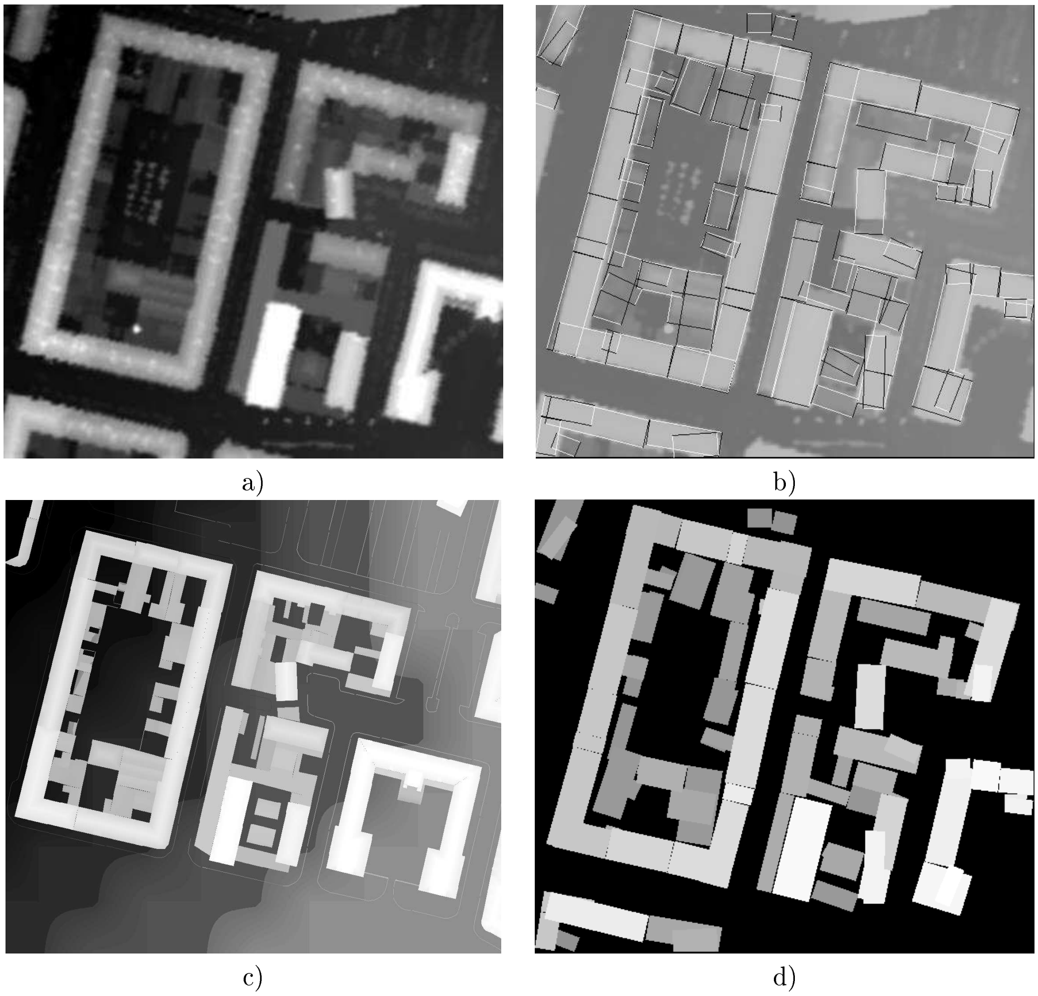


Figure 23: First zone of interest : a) Laser data, b) estimated rectangles, c) ground truth and d) estimated land register plan.

4.3.3 Error computation

The ground truth is at $20cm$ of resolution. We compare it to the land register we have automatically obtained.

However, the ground truth is too precise, especially with respect to the low buildings. We thus present here the Haussdorf distance between the obtained land register and two kinds of classification : one with only high buildings and the other with all buildings.

The vectorial representation extracted from the LASER DEM is projected on an image at $20cm$ of resolution. Once this image has been correctly shifted, it is possible to compute the surface that has been misclassified. This process actually gives a partition into three classes presented in Figure 24:

gray: part of the area which is well classified,

white: part of the area classified as buildings while it is not,

black: buildings that are missed.

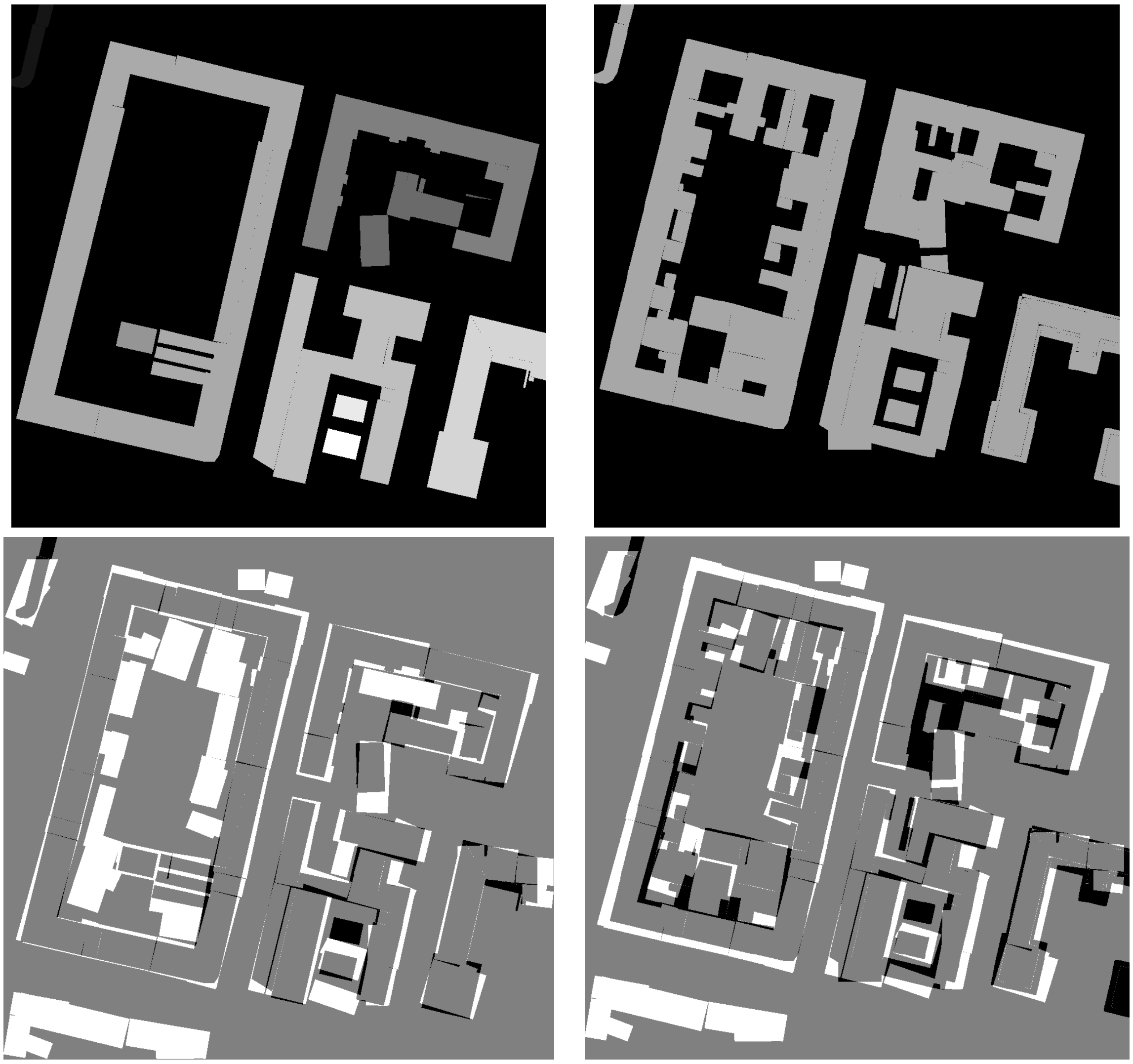


Figure 24: Left: comparison of the obtained result and the ground truth on high buildings only. Right: same comparisons using all the buildings. Top: reference used. Bottom: difference between the obtained land register and the reference: in white, overdetected parts of buildings; in black, missed areas.

4.3.4 Misclassification rates and comments

Of course it is hard to quantify such a result. In order to compare two results, we propose to use the rate of misclassified surface. Some buildings have been missed. This is due to the smooth shape they have on the LASER Data.

Missed area (black)	5%
False alarm (white)	11%

Table 8: Percentage of the misclassified surface using the LASER data

Two comments can be made:

- some artifacts on the DEM are considered as buildings,
- some buildings missing on the ground truth have been found in the DEM, thus our algorithm has completed the provided data.
- quite frequently, the side of the rectangles are subject to overdetection. This occurs because of the accumulation step in the low level filter, and because of the smoothness of the data.

4.3.5 Computing time

The image corresponds to an area which is approximatively 200m by 200m big. Since the resolution of the data is $0.5m$, the size of the processed image is about 400×400.

The automatic extraction of the land register requires 3 hours which is a quite long time, since the algorithm has been implemented in C++ on a Linux engine working at `1Ghz` with `1 Go Ram`.

However, it is important to notice that the next result has taken approximatively the same time to be obtained, even if size of the optical DEM is around 1000×1000. Size of the data is important for the quality of results, but what influences the most the computational time is the complexity of the observed area.

4.4 Second result: Optical DEM

The second result has been obtained on a DEM built using optical data and a correlation algorithm. The resolution is $0.2m$ per pixel, and $0.1m$ per height unit.
The main problem of this DEM is the noise. A lot of small artifacts appear on the DEM. However, the parameters we use are approximatively the same as for the LASER Result.

4.4.1 Data parameters

The results we present in this section have been obtained using the following low level parameters. The first threshold σ_l is higher than for LASER data because on this optical DEM discontinuities are less smooth. Regulation length l_{regul} is higher because this DEM is noisier.

Grid of points	
r	0.5m
e	1m
l_{ext}	10m

Filter	
l_{regul}	4m
σ_l	1*r
σ_h	2m

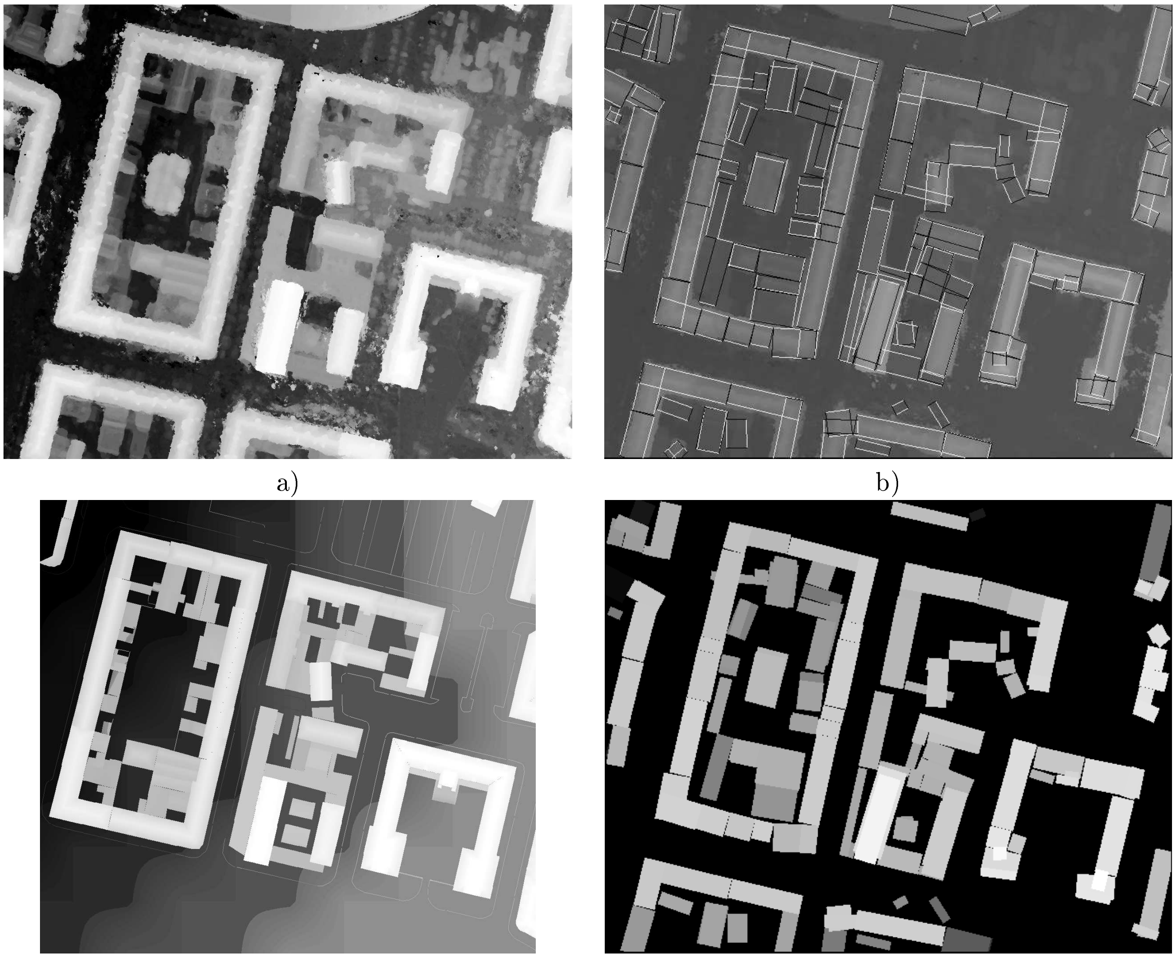
Table 9: Low level parameters used for the correlation data

4.4.2 Estimated configuration

Figure 25 shows the configuration estimated, the ground truth and the associated cover map which has been estimated.

4.4.3 Error computation

The ground truth is at a resolution of $20cm$. We compare it to the land register we automatically obtained.



a) b) c) d)

Figure 25: First zone of interest: a) optical DEM, b) estimated rectangles, c) ground truth and d) land register plan estimated.

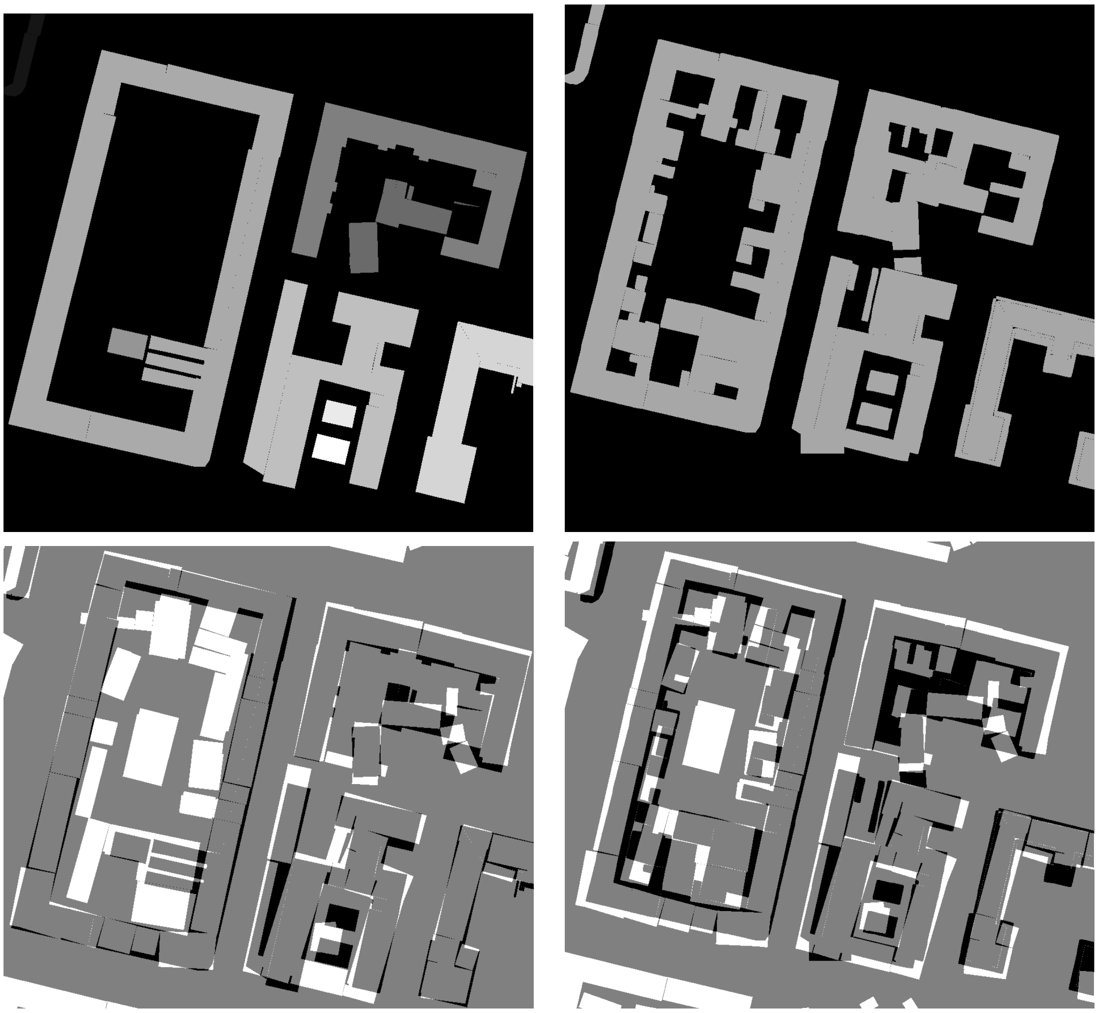


Figure 26: Left: comparison of the obtained result and the ground truth on high buildings only. Right same comparison using all the buildings. Top: reference used, down: difference between the obtained land register and the reference. In white, overdetected parts of buildings; in black, missed areas.

4.4.4 Misclassification rates and comments

The following table presents errors made.

Since the data are noisier than the LASER data, results are a bit worse. Moreover it is worth to notice

Missed area (black)	8.9%
False alarm (white)	10.6%

Table 10: Percentage of the misclassified surface using the optical DEM

that some trees were classified as buildings. It is nevertheless possible to do some post-processing to optimize this result.

Of course it is of important to notice that, except the low level filter parameters, all the other parameters have been kept constant.

4.5 Third Result: complex artifacts

4.5.1 Result

Here we aim at showing the limitations of our algorithm with respect to the building extraction problem.

Thus, we present a result of lower quality on Figure 27, obtained on the optical DEM. The considered area is made of a parking lot, a river, some trees and some complex buildings.

4.5.2 Comments

This area appears to be difficult to deal with, because of:

- the river, along the buildings. This river is responsible for the buildings being detected "on the road": because of the river, there is a discontinuity on the data DEM and thus buildings are added.
- the trees in the parking lot: these trees were classified as buildings, as in the previous result.
- the trees along the buildings of the top of the image: they also have been classified as buildings, because of the geometric correlation.
- the roof texture: the periodic discontinuities in the large roofs give an oversegmentation.
- the river and the trees in the parking lot: the discontinuities provided by the river and the trees led our algorithm to consider some buildings are present.

A post-processing may improve the obtained result. Such a post processing should involve:

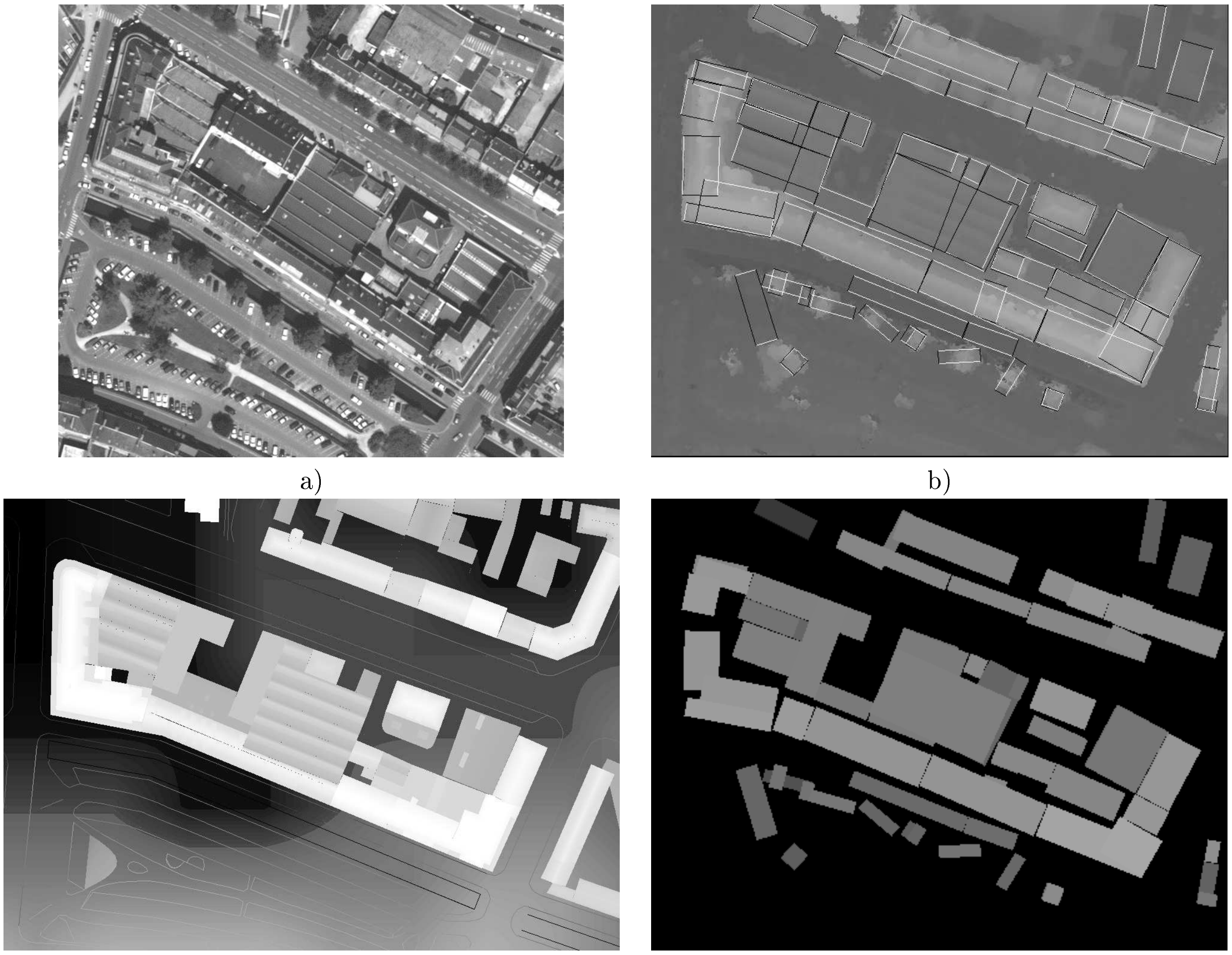
- the elimination of detected buildings corresponding to trees (using, for instance, a criterion on the variance of the normal to the surface),
- the elimination of other detected structures that do not correspond to a building,
- the fusion of rectangles into more complex shapes to be able to deal with complex roofs.

4.6 Fourth and fifth results: other areas

We present the two last results obtained on the optical DEM. The dimensions of these two areas are in the same range than former used areas : 200 meters by 200 meters corresponding to 1000 pixels by 1000 pixels images.

4.6.1 Old city

The first result is presented in Figure 28. This area corresponds to an old part of the city of Amiens. This area is difficult to deal with, since there are a lot of trees and a lot of little houses. We have compared the ground truth and a the result obtained on a subset of the images, since the complete ground truth is not available. The following table gives misclassification rates.



a) b)

c) d)

Figure 27: Second zone of interest: a) aerial photography, b) optical DEM and rectangle extracted, c) ground truth and d) land register plan estimated.

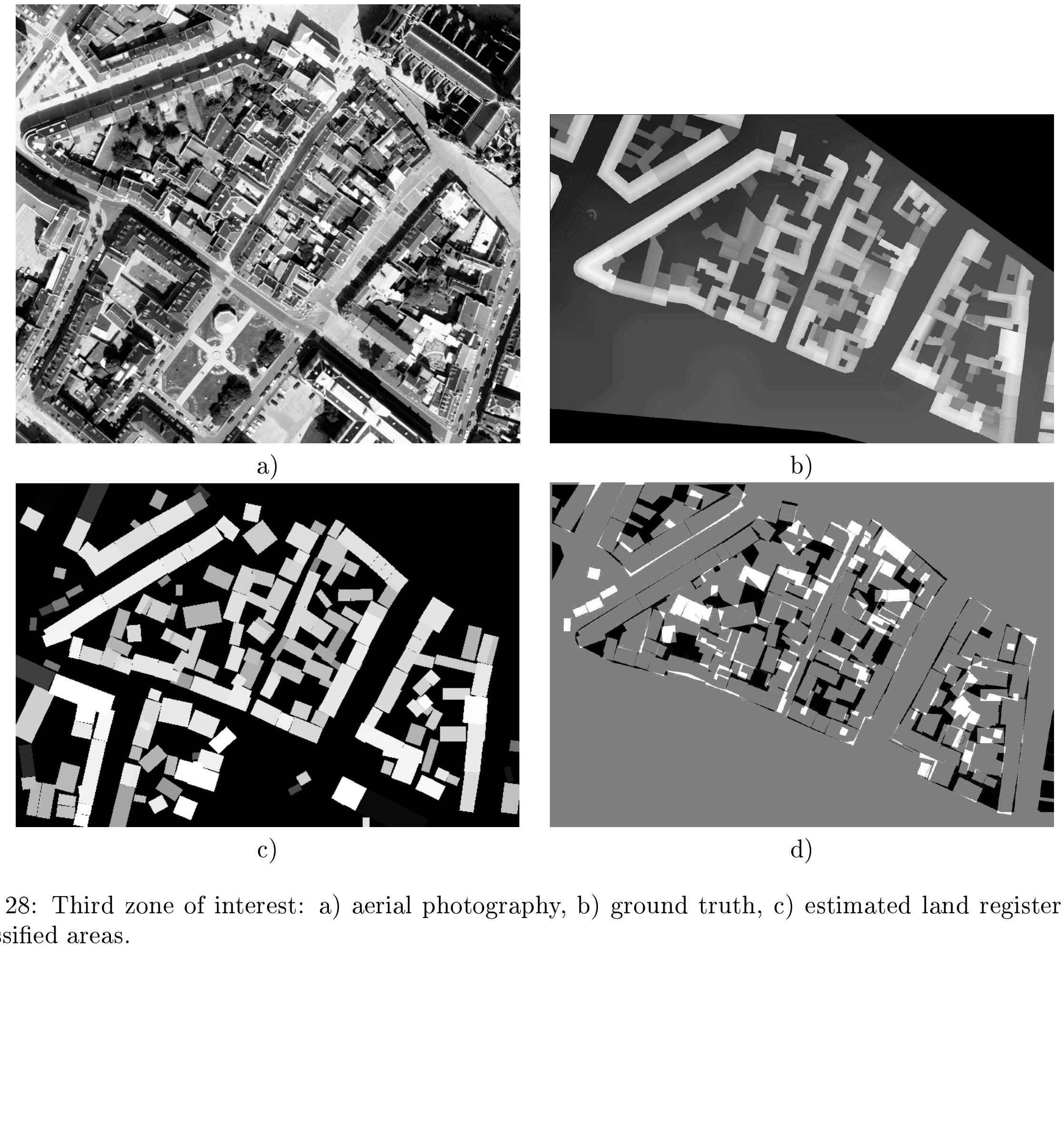


Figure 28: Third zone of interest: a) aerial photography, b) ground truth, c) estimated land register and d) misclassified areas.

Missed area (black)	8.01%
False alarm (white)	5.3%

Table 11: Percentage of the surface that was misclassified using the optical DEM (fourth result)

4.6.2 Fourth zone

The last result is presented in Figure 29. The optical DEM is smooth in the backyards of buildings. Discontinuities are difficult to detect in these cases.

Missed area (black)	11.4%
False alarm (white)	5.9%

Table 12: Percentage of the surface that was misclassified using the optical DEM (fifth result)

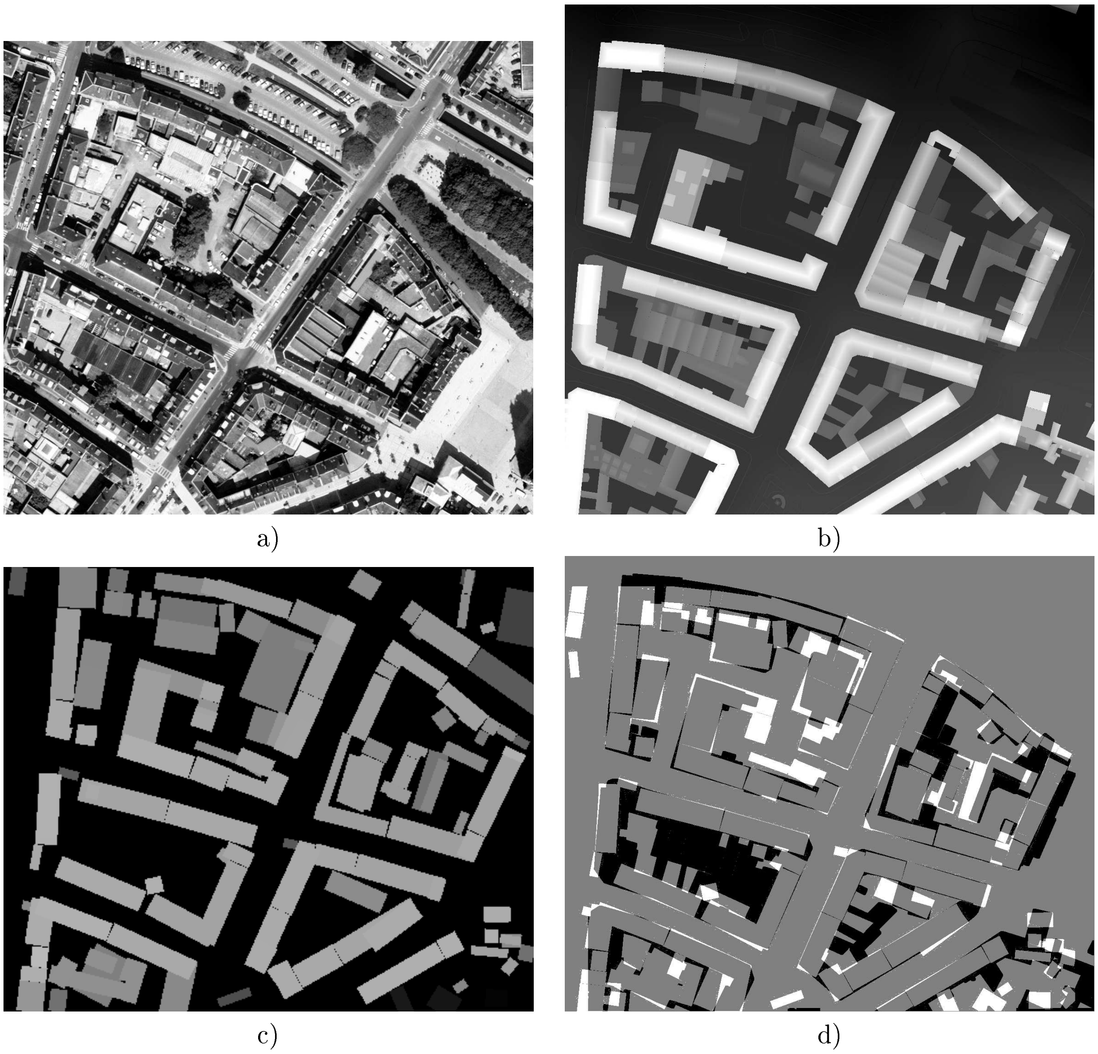


Figure 29: Fourth zone of interest : a) aerial photography, b) ground truth, c) estimated land register and d) misclassified areas.

Conclusion

In this report we have presented a powerful algorithm that proved to be efficient on a difficult problem.

The practical advantages of the algorithm we proposed are the following :

- the algorithm extracts simple rectangular shapes from altimetric data,
- it is an automatic approach, there is no need of any interaction with an operator,
- the algorithm proved to be efficient even on dense urban areas,
- the parameterization proved to be robust, since the same model is used for both LASER and optical DEMs,
- the data term might be easily adapted to other types of altimetric data,
- the obtained results might be very useful for more precise 3D reconstruction algorithms such as those proposed in [10].

Nevertheless, there are some drawbacks :

- the algorithm is quite slow, it still requires a few hours to process a 1000 by 1000 pixel image,
- it is a stochastic algorithm : the result may be different between two runs, even if the error made would be of same order,
- some parameters must be tuned by hand,
- some mistakes are done due to the type of data used. For instance, it is sometimes difficult to make a difference between a building and a tree, and sometimes, the rectangular shape used appears to be too simple,
- the knowledge provided by altimetric data seems not to be sufficient to obtain a result of very high quality.

However, one should point out that :

- we have presented an elegant way of mixing low level information and high level knowledge by using point processes,
- we have experimentally proved that a non Bayesian model can be powerful,
- we have introduced a relevant way of using non homogeneous Poisson point processes.

In the future, we should :

- implement a MCMC-ML procedure to see if it is relevant to estimate some parameters,
- try to do data fusion and use more data,
- use more complex models of building to improve the quality of the obtained result,
- improve the computational time : maybe a hierarchical approach can improve it by first detecting higher buildings, and then improving the obtained description; another possibility is to use some primitives and introduce a non-homogeneous birth or death that allows to make a building appear only in relevant positions.

Annex A : profile simplification

We present here some results obtained with the low level filter. These results are useful to understand the influence of the corresponding parameters.

We present here 5 results on a profile of interest. The 5 experiments are described by the following table.

	Type of Data	Parameters			
		r	σ_h	l_{regul}	σ_l
Exp 1	Optical	0.5 m	2 m	1 m	1*r
Exp 2	Optical	0.5 m	2 m	4 m	1*r
Exp 3	Optical	0.5 m	2 m	1 m	2*r
Exp 4	Optical	1 m	2 m	1 m	1*r
Exp 5	Laser	0.5 m	2 m	1 m	1*r

Table 13: Parameters used for the different experiments

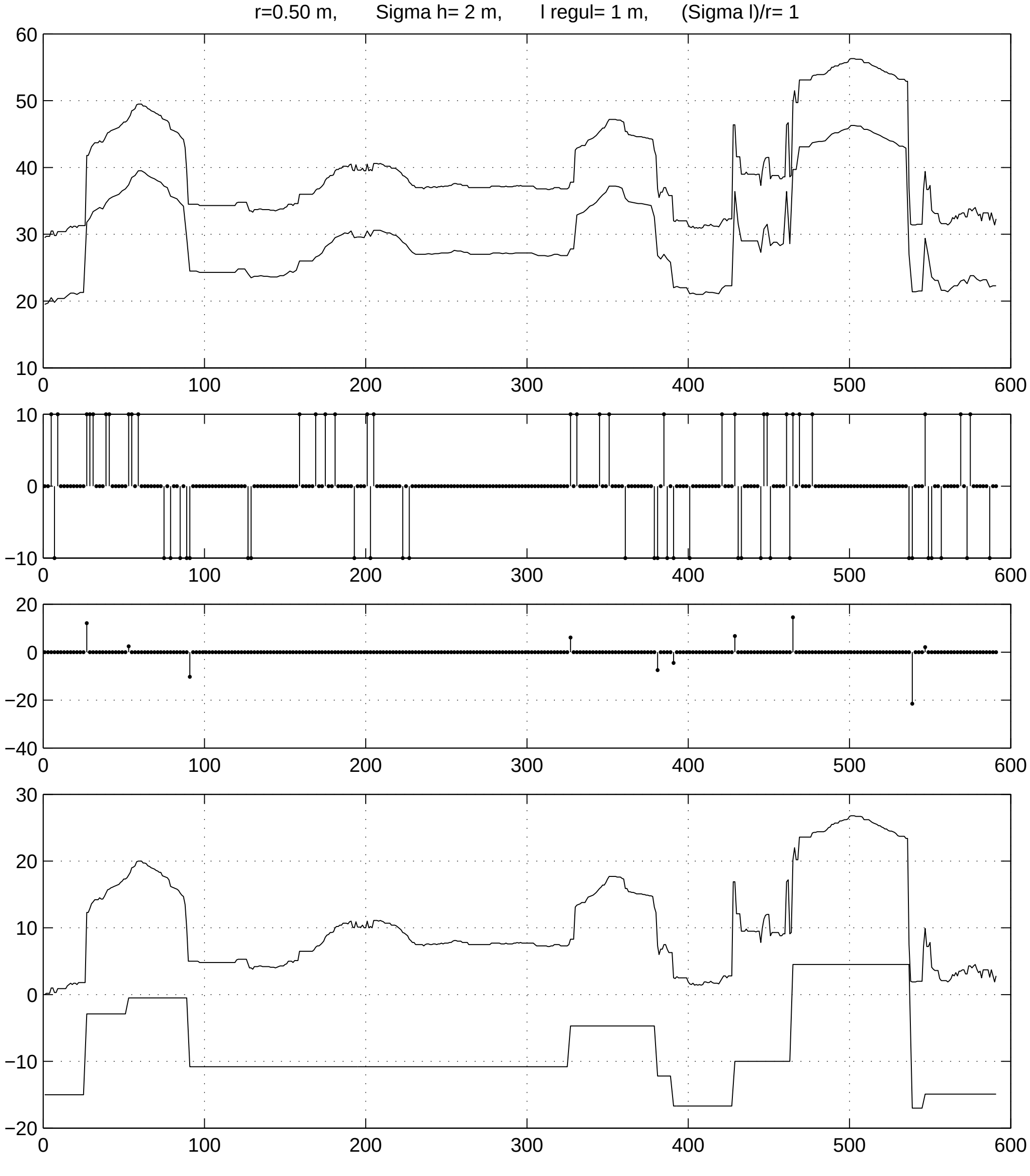


Figure 30: Profile simplification : experiment 1. Top to down : true profile, sub-sampled profile (depending on r), hit gradients ($\geq \sigma_l$), detected discontinuities after accumulation, opening and thresholding by σ_h, and finally true profile versus simplified profile obtained.

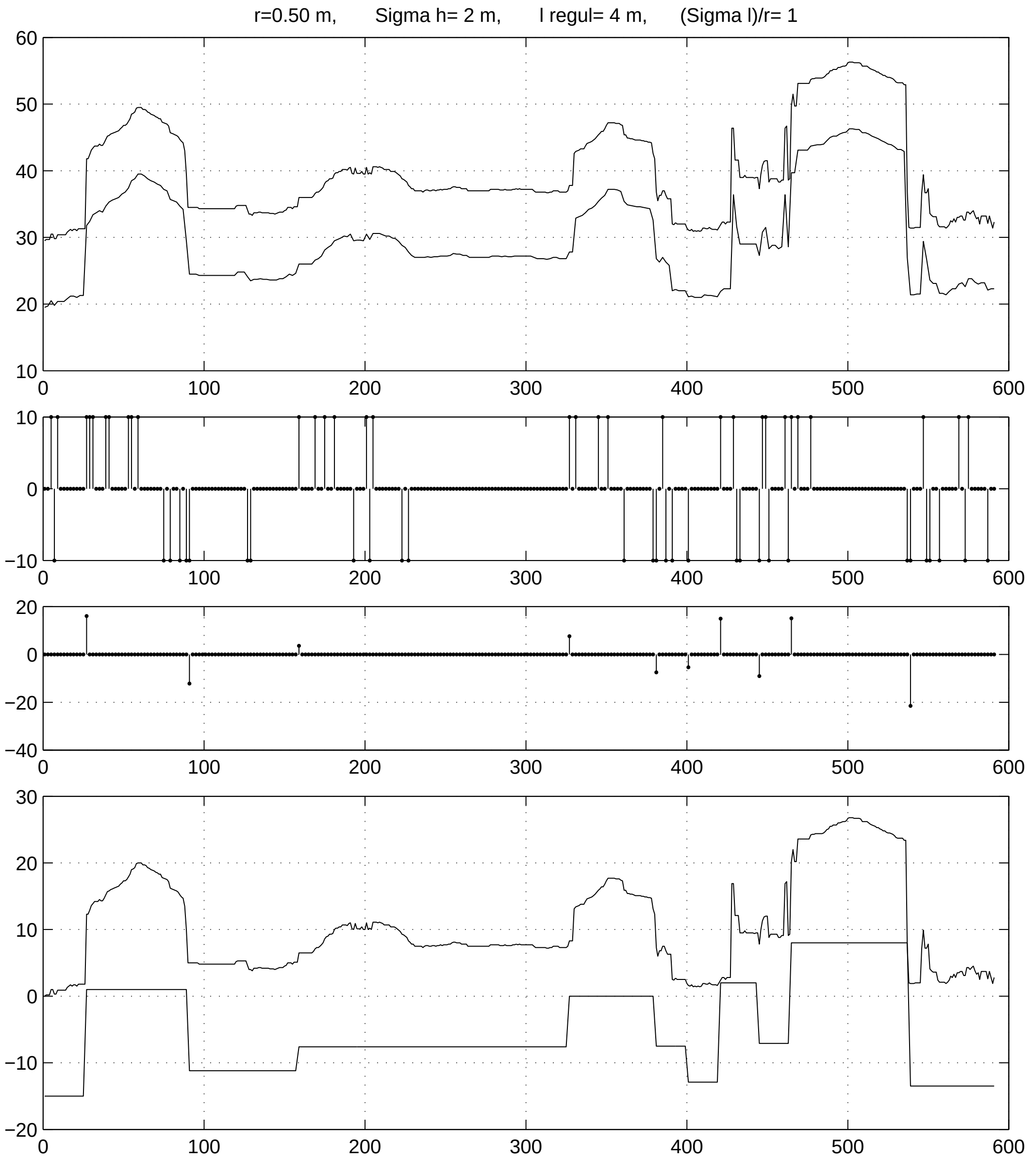


Figure 31: Profile simplification : experiment 2. The regularization parameter is set to $l_{regul} = 4m$.

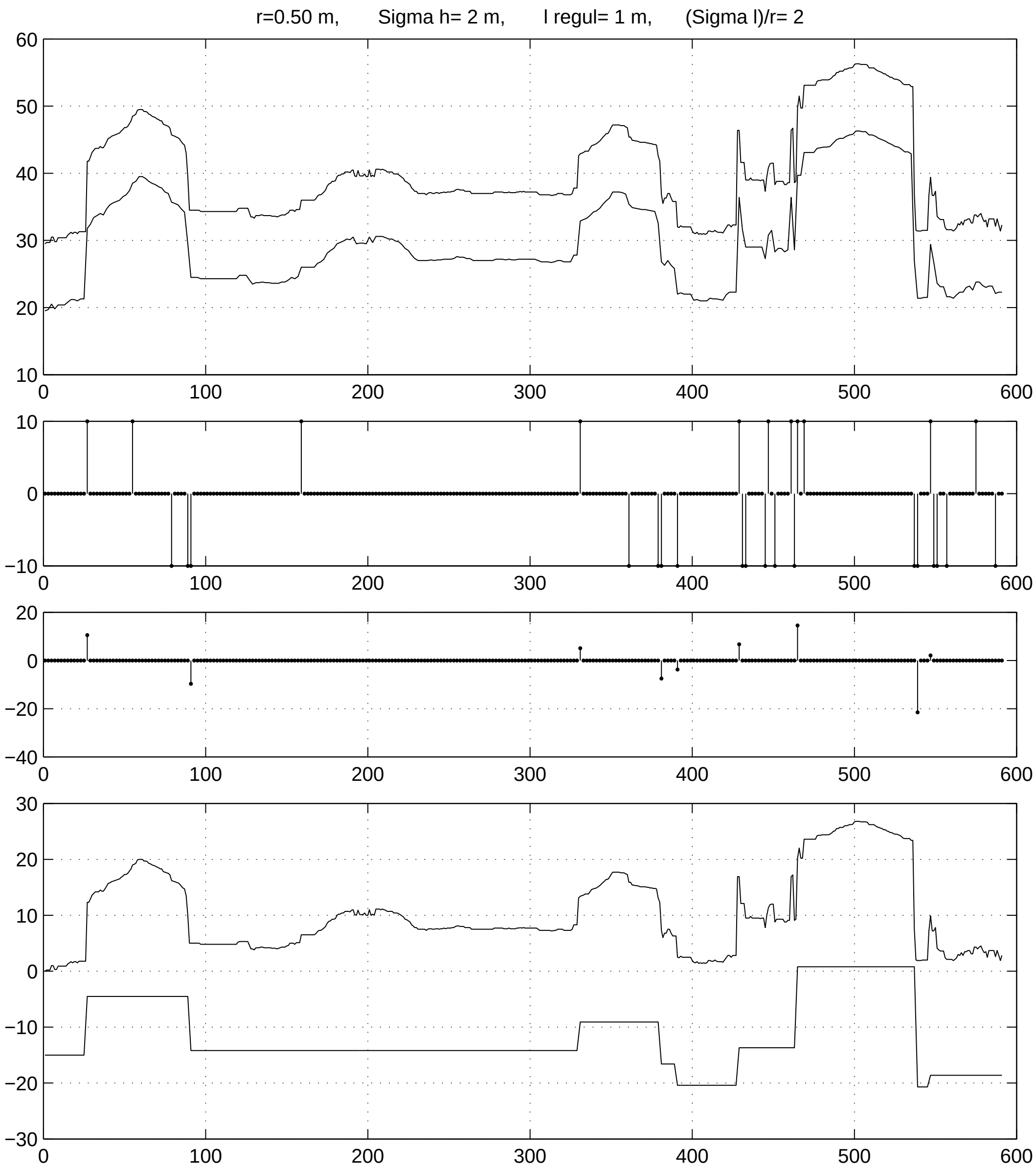


Figure 32: Profile simplification : experiment 3. The low threshold is twice as big than for exp. 1.

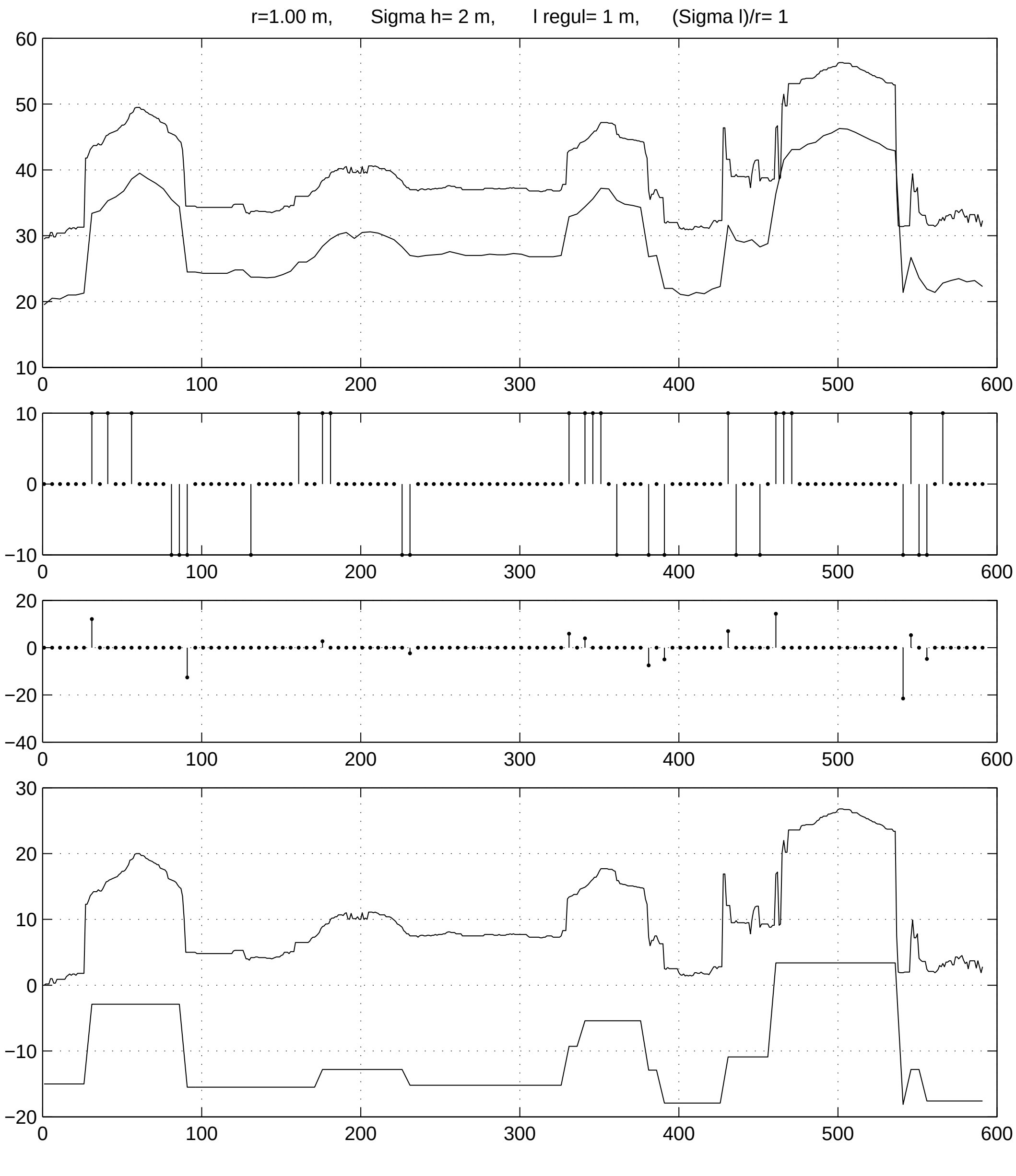


Figure 33: Profile simplification : experiment 4. The resolution parameter r is set to 1 m.

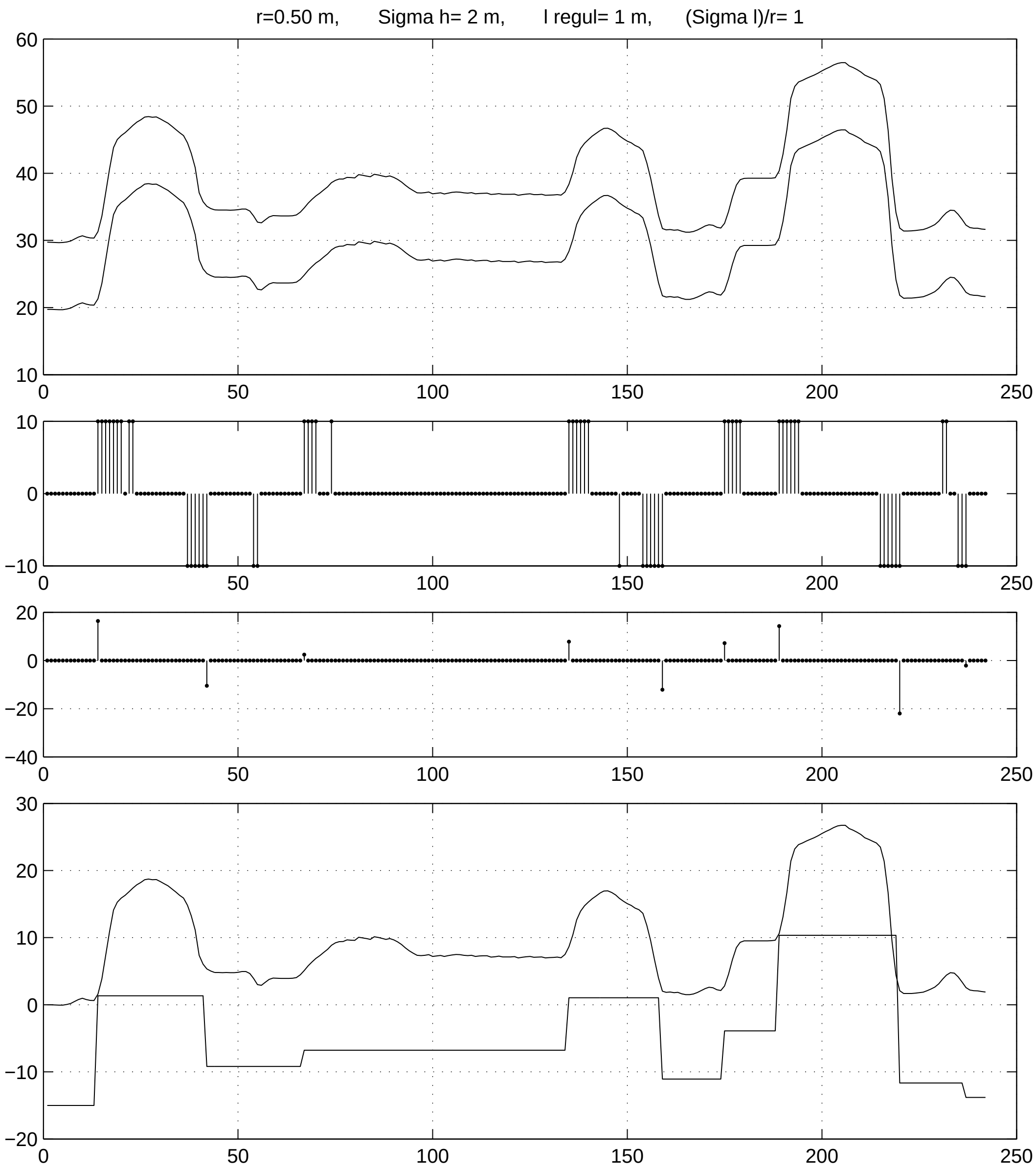


Figure 34: Profile simplification : experiment 5. Result on the corresponding Laser profile.

Annex B : birth or death in a neighborhood

We present here briefly how we have implemented this kind of transformation and the associated Green's ratio.

For two given rectangles, all attractive relations are described by:

- a distance constraint over the two corners of interest $C(u)$ and $C(v)$,
- a distance constraint over the angle of the rectangles.

Birth

If birth has been chosen (with probability p_b), the procedure is the following :

1. select randomly an object u in $\gamma_0 \cup \mathbf{x}$. If the set is empty do nothing.
2. once u has been chosen, generate v, new rectangle such that $v \sim u$. A good solution is to simulate $Z \in \mathbb{R}^5$ as :
$$z = \begin{pmatrix} x \\ y \\ \delta\theta \\ L \\ l \end{pmatrix}$$
where :
    - (x, y) are the coordinates of the corner of interest $C(v)$ of the generated object. Thus, the point made by this couple should be in a disk of center $C(u)$ and radius d_{max}. We propose to uniformly generate this couple in the disk. A good solution is to uniformly generate points in a square of side $2 * d_{max}$ until one of them falls in the disk. A polar parametrisation should be avoided, as pointed out in [16],
    - $\delta\theta$ is the difference of angle between u and v : it should be generated in $[-d_t, d_t]$.
    - L, l are the length and the width of v, generated in $[L_{min}, L_{max}]$ and $[l_{min}, l_{max}]$.
3. once z is obtained, construct the unique associated v.
4. compute $\mathcal{R}^+(\mathbf{x} \cup v)$, set of pairs of related points in $\mathbf{x} \cup v$ such that one of the point is in γ_0 :
$$\mathcal{R}^+(\mathbf{x}) = \{\{w, w'\} \;\; w \in \mathbf{x} \;\; w' \in \mathbf{x} \quad w \sim w' \quad \{w, w'\} \cap \gamma_0 \neq \emptyset\}$$
compute $s(\mathbf{x} \cup v) = \text{card}\mathcal{R}^+(\mathbf{x})$ and j_d :
$$j_d^{\mathbf{x}\cup v}(v) = \frac{1}{s(\mathbf{x} \cup v)} \sum_{u \in \mathbf{x}\cup v \,:\, u \sim v} \frac{1}{2}\mathbf{1}_{\gamma_0}(u)\mathbf{1}_{\gamma_0}(v) + \mathbf{1}_{\gamma_0}(u)(1 - \mathbf{1}_{\gamma_0}(v))$$
and the j_b term :
$$\sum_{u \in \mathbf{x} \,:\, u \sim v} j_b^{\mathbf{x}}(u) = \sum_{u \in \mathbf{x} \,:\, u \sim v} \frac{\mathbf{1}_{\gamma_0}(u)}{\text{card}\gamma_0(\mathbf{x})}$$
5. compute the associated Green's ratio given by :
$$R(\mathbf{x}, \mathbf{x} \cup v) = \frac{h(\mathbf{x} \cup v)}{h(\mathbf{x})} \frac{pd}{p_b} \frac{j_d^{\mathbf{x}\cup v}}{\sum_{u \in \mathbf{x} \,:\, u \sim v} j_b^{\mathbf{x}}(u)} \frac{\pi * d_{max}^2 * d_t * 2}{\pi}$$

Death

If death has been chosen, the procedure is the following :

1. select randomly a pair $\{w, w'\}$ of related objects in $\mathcal{R}^+(\mathbf{x})$,
2. choose a point v in the chosen pair :
   - if both w and w' are in $\gamma_0(\mathbf{x})$ choose one of the two with probability 0.5,
   - otherwise v is taken as the unique object in the pair belonging to γ_0,
3. compute Green's ratio :

$$R(\mathbf{x}, \mathbf{x} \setminus v) = \frac{h(\mathbf{x} \setminus v)}{h(\mathbf{x})} \frac{p_b}{p_d} \frac{\sum_{u \in \mathbf{x} \,:\, v \sim u} j_b^{\mathbf{x} \setminus v}(u)}{j_d^{\mathbf{x}}(v)} \frac{\pi * d_{max}^2 * d_t * 2}{\pi}$$

Acknowledgments

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

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] 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.

[3] 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.

[4] 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.

[5] F. Fuchs. *Contribution à la reconstruction du bâti en milieu urbain, à l'aide d'images aériennes stéréoscopiques à grande échelle. Etude d'une approche structurelle.* PhD thesis, Université René Descartes, Paris V, France, 2001.

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

[7] 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.

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

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

[10] H. Jibrini. *Reconstruction automatique des bâtiments en modèles polyédriques 3D à partir de données cadastrales vectorisées 2D et d'un couple d'images aériennes à hautes résolution.* PhD thesis, ENST, Paris, France, 2002.

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

[12] R. Nevatia and K. Price. Automatic and interactive modeling of buildings in urban environments from aerial images. In *ICIP02*, volume III, pages 525–528, September 2002.

[13] 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/index-eng.html`.

[14] M. Ortner, X. Descombes, and J. Zerubia. Building detection from digital elevation models. *INRIA Research Report 4517*, July 2002.

[15] M. Ortner, X. Descombes, and J. Zerubia. Building detection from digital elevation models. In *ICASSP*, volume III, Hong Kong, April 2003.

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

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

[18] P. Soille. *Morphological Image Analysis.* Springer-Verlag, Heidelberg, 2nd edition, 2003.

[19] M. N. M. van Lieshout. Stochastic annealing for nearest-neighbour point processes with application to object recognition. *CWI Research Report, BS-R9306, ISSN 0924-0659*, 1993.

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

[21] S. Vinson and L. D. Cohen. Multiple rectangle model for buildings segmentation and 3D scene reconstruction. In *16th IEEE Int. Conf. on Pattern Recognition*, Québec, Canada, August 2002.

[22] S. Vinson, L. D. Cohen, and F. Perlant. Extraction of rectangular buildings using DEM and orthoimage. In *Scandinavian Conference on Image Analysis*, Bergen, Norway, June 2001.

[23] O. Viveros-Cancino. *Analyse du milieu urbain par une approche de fusion de données satellitaires optiques et radar.* PhD thesis, Université de Nice-Sophia Antipolis, June 2003.

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