HAL
open science

An efficient stochastic approach for building footprint extraction from digital elevation models

Olivier Tournaire, Mathieu Brédif, Didier Boldo, Mélanie Durupt

To cite this version:

Olivier Tournaire, Mathieu Brédif, Didier Boldo, Mélanie Durupt. An efficient stochastic approach for building footprint extraction from digital elevation models. ISPRS Journal of Photogrammetry and Remote Sensing, 2010, 65 (4), pp.317 - 327. 10.1016/j.isprsjprs.2010.02.002 . hal-01881391

HAL Id: hal-01881391

https://hal.science/hal-01881391v1

Submitted on 1 Oct 2018

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

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

An Efficient Stochastic Approach for Building Footprint Extraction from Digital Elevation Models

O. Tournaire [a] M. Brédif [a] D. Boldo [a]

[a] *Laboratoire MATIS, IGN, 73 avenue de Paris, 94165 Saint-Mandé, France*

Abstract

In the last two decades, building detection and reconstruction from remotely sensed data has been an active research topic in the photogrammetric and remote sensing communities. Recently, effective high level approaches have been developed, *i.e.*, the ones involving the minimization of an energetic formulation. Yet, their efficiency has to be balanced by the required amount of processing power required to obtain good results.

In this paper, we introduce an original energetic model for buildings footprint extraction from high resolution digital elevation models ($\leq$1 m) in urban areas. Our goal is to formulate the energy in an efficient way, easy to parametrize and fast to compute, in order to get an effective process still providing good results.

Our work is based on stochastic geometry, and in particular on marked point processes of rectangles. We therefore try to obtain a reliable object configuration described by a collection of rectangular building footprints. To do so, an energy made up of two terms is defined: the first term measures the adequacy of the objects with respect to the data and the second one has the ability to favour or penalize some footprint configurations based on prior knowledge (alignment, overlapping, ...).

To minimize the global energy, we use a Reversible Jump Monte Carlo Markov Chain (RJMCMC) sampler coupled with a simulated annealing algorithm, leading to an optimal configuration of objects. Various results from different areas and resolutions are presented and evaluated. Our work is also compared with an already existing methodology based on the same mathematical framework that uses a much more complex energy. We show how we obtain similarly good results with a high computational efficiency (between 50 and 100 times faster) using a simplified energy that requires a single data-independent parameter, compared to more than 20 inter-related and hard-to-tune parameters.

Key words: Digital Elevation Model, Building footprint, Energetic modeling, Marked point processes, RJMCMC

---

1 Introduction

Building extraction and reconstruction from remotely sensed images has been a motivating topic for many researchers in the past years. Indeed, 3D urban models can be very useful in various applications: virtual tourism, models of wave propagation for telecommunication operators or realistic environments for video games among others. However, before providing a volumetric representation of the buildings, most automatic methods need to focus on their 2D outlines (Haala et al., 1998; Brunn and Weidner, 1997; Lafarge et al., 2008). This is the problem we propose to address in this paper, without any prior

* Corresponding author: Olivier Tournaire.

Tel: (33) 1 43 98 80 00 + 7125 – Fax: (33) 1 43 98 85 81

Email addresses: `olivier.tournaire@ign.fr` (O. Tournaire), `mathieu.bredif@ign.fr` (M. Brédif), `didier.boldo@ign.fr` (D. Boldo).

knowledge on the buildings location and only loose constraints on shape.

The work presented here is centered on footprint extraction. In 3D reconstruction tools, these region of interest (ROI) usually come from cadastral maps, manually delineated outlines, or from some manual or automatic processing on a digital elevation model or an orthoimage. However, such data are costly to create "by hand" and not always available, may be outdated and can suffer of discrepancies with ground truth. Thus it appears necessary to have a rapid, robust and reliable methodology to extract buildings footprint automatically. To do so, we use the framework of Ortner et al. (2007), and propose improvements to make it useful in an operational context, that is to say with reasonable computation times and simple parametrization, without lost, neither in terms of quality, nor robustness, nor generality.

1.1 State of the art

Today, a tremendous variety of approaches exists in the literature. By the way, with the growing amount of data to process due to the increasing resolution of data and extent of areas to cover, automatic methods need to be considered with a particular attention.

Haithcoat et al. (2001) use size, height and shape characteristics to differentiate buildings from other objects on a Lidar digital elevation model (DEM, rasterized heights on a regular grid). Thresholds discriminate small objects like cars and differential geometries quantities eliminate trees. The result is rasterized and footprints are simplified. However, this simplification assumes orthogonality of the building main directions, a too restrictive hypothesis in dense urban areas.

With a similar approach based on simple thresholds, Alharthy and Bethel (2002) evaluate the potential of Lidar data for 3D building extraction. This first analysis provides blobs which need to be converted into intersection free polygons. A coarse representation is then obtained using restrictive hypothesis of orthogonality of buildings parts.

From high-resolution commercial satellite data, Shackelford et al. (2004) involve mathematical morphology operators in a multi-scale framework for the delineation of buildings. This method is however sensitive to errors, as they are propagated from one level of analysis to the next.

In an optimization framework and with lidar data, Wang et al. (2006) first chain boundary points detected on Lidar data and simplify them to obtain an estimate of the building footprint. A Bayesian Maximum a Posteriori (MAP) estimation then allows to improve the footprint. The distance of the boundary points to the polygon and a prior favouring particular directions of the building edges are embedded

In a more recent work, Frédéricque et al. (2008) first detect building footprints before a reconstruction step to populate a database. Their approach focuses on a ROI and extracts its skeleton. A set of rectangle hypotheses is then generated with the principal directions at given points of the skeleton. An iterative algorithm then allows to obtain a simplified graph of rectangles, providing the representation of a building block as a set of rectangles (computation time for the whole process is about 20 seconds for a building block).

In this paper, we use the marked point process framework presented in Ortner et al. (2007). It proposes an energetic formulation and states the problem as a minimization of a functional made up of a data attachment term and a regularizing term. The sampling of this objective function is conducted thanks to a RJMCMC sampler and coupled with a simulated annealing to find its

optimum. This approach is very interesting as it allows us to find an optimal configuration of objects, namely rectangles representing buildings parts, without initialization *prior* on the object localizations or even their number. External knowledge can also be introduced, which allows the modeling of the layout of the buildings.

The main drawback of this approach is its computational efficiency. Ortner's energy is composed of many terms, thus computationally intensive and leading to a very long time to obtain good results. There are also many parameters. Some of them are related to physical values (resolution, lengths, ...), some other are not (weighting factors, number of profiles, ...). As all these parameters have to be tuned simultaneously, this is a major pitfall of the approach. We focus in this paper on a way to simplify this energy, parameters tuning and to improve the computation time without loss of quality. This point is crucial when large areas have to be processed in a reasonable amount of time.

1.2 Global strategy and organisation of the paper

As aforementioned, our work is based on an already existing methodology. Our aim is to propose an efficient way to use it and the rest of the paper is dedicated to the resolution of this problem. First, we remind the foundations of the mathematical framework involved in the process and the optimization outline (section 2). In section 3, we give the essential details of Ortner's energy and propose our new formulation to solve the problem. Section 4 presents the choices we made to implement our approach: they concern geometric implementation details and the data structures involved. Section 5 presents a wide variety of results. They cover different areas, and show the ability of the ap-

proach to deal with various resolutions and data sources. As input data, we will use a DEM, computed either from high resolution images or from a 3D point cloud acquired with an aerial lidar. Finally, section 6 concludes our work and proposes future directions.

2 Mathematical background

In our work, buildings footprint are extracted from digital elevation models with various resolutions (from 10 cm to 1 m). For buildings, a good geometric shape descriptor covering this range of resolutions is the rectangle. Even if buildings are more complicated than a simple rectangle, they can be described by an union of rectangles which is able to generate complex structures and thus handle most of urban built areas (figure 1).

This section presents the mathematical background on which relies the method-

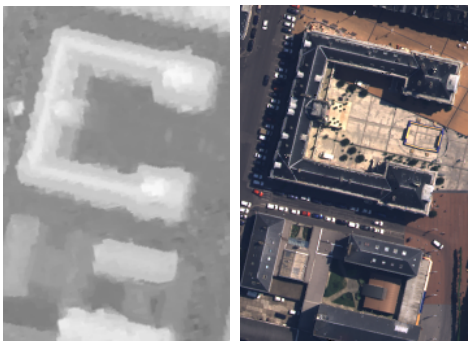

Fig. 1. Examples of buildings on a DEM (*left*) and on an image (*right*), 25 cm, Amiens, France ©IGN 2009. In the bottom of the images, buildings can be modeled by a group of 3 separate rectangles; on the top, the city-hall building can be represented by an union of rectangles (in this case, depending on the generalization, 3 or 5 rectangles are required).

ology developed in this paper. We begin with generalities (sections 2.1 to 2.3)

and then focus on the key points of the framework, namely the sampler and the optimizer (sections 2.4 and 2.5).

2.1 Generalities

The principle of the approach presented in this work is based on marked point processes, an object oriented approach. Let a configuration of objects be a collection of rectangles in image space, each of them representing a building part. Our aim is to find the best configuration. To do so, an energy is built, using 2 terms: the data attachment term measures the consistency of the objects with respect to the image, the regularization term favors specific layouts of rectangles. To optimize the global energy, we use a RJMCMC sampler (Hastings, 1970; Green, 1995) and couple it with a simulated annealing (Salamon et al., 2002). This last point is important because our problem lies in a high dimensional space of unknown dimension and is also non convex. Thus, this global optimization tool is well adapted to our case as it finds the minimum of the functional without being trapped in local minima.

Stochastic models have shown their versatility in various domains of image processing and computer vision (van Lieshout, 2000; Winkler, 2003). A particular family of stochastic approaches, namely marked point processes, originally developed in response to various physical or biological problems, is very useful to detect objects modeled with geometric shapes. First of all, they can handle geometric constraints on the objects of interest and are able to manage very high dimension spaces, as this is the case in our application. They also allow an energetic formulation of the problem which can be seen as a functional minimization. In remote sensing, they have been used by Lacoste et al.

(2005) to detect linear structures like roads and rivers in satellite images. In a forestry context, Perrin et al. (2006) uses ellipses as objects of a marked point process to delineate tree crowns from high resolution aerial images. Lafarge et al. (2008), using building footprints as an input, proposes a stochastic modeling for 3D urban environments reconstruction.

The mathematical background on which relies all those works has already been well presented in the literature. We remind here only the basis of those principles. The reader interested in more details will find theoretical and practical aspects of marked point processes in Cressie (1993) or Stoyan et al. (1996).

2.2 Definitions

Informally, a *point process* is a stochastic model governing the locations of some events $\{x_i\}$ in a bounded set $\mathcal{X}$. More formally, this a measurable application of a probability space into $[\mathbb{N}, \mathcal{P}]$. Here, $\mathbb{N}$ represents the set of sequences of points in an image $\mathcal{P} = [0, X_{max}] \times [0, Y_{max}]$ locally finite (each finite subset of $\mathcal{P}$ must contain only a finite number of points) and simple ($x_i \neq x_j$ if $i \neq j$). Such a sequence will be called a *configuration* and denoted $\mathcal{C}_k \subset \mathbb{N}$ where k is the number of elements in the configuration(figure 2:a). A *marked point process* is a point process where each $\{x_i\}$ has an associated mark $\{m_i\}$ in a set $\mathcal{M}$, thus allowing to define a geometric object. A marked point process is as a point process on $\mathcal{P} \times \mathcal{M}$ (figure 2:b).

Building footprints in a DEM can be seen as the realization of a marked point process of rectangles. For performance purposes (see section 4.3), we choose a parametrization of the rectangle which differs from the classical representation (center, width, height, orientation; see figure 2:c). Formally, it is described by

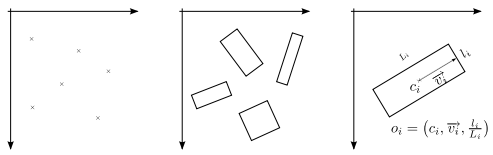

Fig. 2. a) A realization of a point process on $\mathcal{P}$; b) A realization of a marked point process of rectangles on $\mathcal{P} \times \mathcal{M}$; c) An object o_i of our marked point process is described by its center $c_i = (x_i, y_i)$, its semi-major axis $\overrightarrow{v_i}$ and its aspect ratio $r_i = \frac{l_i}{L_i}$.

its center $c_i = (x_i, y_i)$, a vector $\overrightarrow{v_i}$ for the semi-major axis (*i.e.* $\|\overrightarrow{v_i}\| = \frac{L_i}{2}$) and its aspect ratio $r_i = \frac{l_i}{L_i} \leq 1$ where L_i (*resp.* l_i) is the rectangle dimension along (*resp.* across) $\overrightarrow{v_i}$. As we are only interested in rectangles contained in the DEM image support $\mathcal{P}$, the compact set $\mathcal{M}$ of the marks is:

$$\mathcal{M} = \underbrace{\left[\frac{-X_{max}}{2}, \frac{X_{max}}{2}\right] \times \left[\frac{-Y_{max}}{2}, \frac{Y_{max}}{2}\right]}_{\overrightarrow{v_i}} \times \underbrace{]0, 1]}_{r_i} \tag{1}$$

2.3 From a density to an energy

Les us consider a marked point process $\mathcal{X}$ defined through its probability density f with respect to the law $\pi_{\nu(.)}$ of a Poisson process known as the reference process. $f(.)$ can be defined in two ways: in a Bayesian framework, which requires to have a model of heights in the whole DEM or through a Gibbs energy. We choose the second form because of the complexity required to build a height model valid for the whole area of interest. Thus, for a configuration $\mathcal{C}$, the density $f(.)$ is:

$$f(\mathcal{C}) = \frac{1}{Z} \exp - \mathcal{U}(\mathcal{C}) \tag{2}$$

where $Z = \int \exp - \mathcal{U}(\mathcal{C})$ ensures normalization. Moreover, the energy $\mathcal{U}(.)$ can be expressed as a weighted sum of an internal energy measuring the spatial quality of the collection of objects (*prior* or internal field) and an external

energy which qualifies the quality of the objects with respect to the underlying DEM (external field). The unnormalized formulation is then:

$$f(\mathcal{C}) = \exp - [\mathcal{U}_{\text{prior}}(\mathcal{C}) + \mathcal{U}_{\text{data}}(\mathcal{C})] \tag{3}$$

Extracting buildings footprint consists in finding the configuration of objects $\tilde{\mathcal{X}}$ maximizing the posterior probability $\tilde{\mathcal{X}} = argmax\ f(.)$. This will be done thanks to a RJMCMC sampler and a simulated annealing algorithm which offers convergence towards the global optimum of the density.

2.4 RJMCMC sampler

Classical MCMC methods such as Metropolis-Hastings cannot handle dimension jumps, *i.e.*, changes in dimension between samples. RJMCMC, which consists in simulating a Markov Chain on the configuration space, is an extension which allows the modeling of a scene with an unknown number of objects. (Green, 1995) proposed to use reversible jumps, *i.e.*, a transition from one step of the chain to another is guided through a set of proposition kernels, the *jumps*. They enable to build a new configuration from a starting configuration, with the modification of an already existing object, or with adding or removing an object of the current configuration, thus involving a change in the dimension space. Algorithm 1 describes the overall process (a complete description can be found in Robert and Casella (2004)).

2.5 Simulated annealing

The RJMCMC sampler is coupled with a simulated annealing in order to find the optimum of the density $f(.)$. Instead of $f(.)$, we use in the optimization

Algorithm 1: RJMCMC sampler

Input: An initial configuration X_0 (empty or random configuration)

- Randomly choose a proposition kernel Q_i according to a defined law
- Build a new configuration y from X_t w.r.t. the selected kernel
- Compute the acceptance ratio $R\left(X_t, y\right)$
- Compute the acceptance rate $\alpha = \min\left(1, R\left(X_t, y\right)\right)$
- With probability $\begin{cases} \alpha & \text{set } X_{t+1} = y \\ (1-\alpha) & \text{set } X_{t+1} = X_t \end{cases}$

process $f\left(.\right)^{\frac{1}{T_t}}$, where T_t is a sequence of decreasing temperatures which tends to zero as t tends to infinity. Theoretically, convergence is guaranteed whatever the initial configuration X_0 is if the decrease of the temperature follows a logarithmic scheme. In practice this is impossible to use such a scheme since it is too slow. Thus, it is generally replaced with a geometric decrease which gives a good solution close to the optimal one. To estimate initial and final temperatures, we sample the configuration space and consider the variance of the global energy (Salamon et al., 2002).

3 Energetic modeling

Formulation of the energies is the core of our work. This is one of the major aspect of the simplicity and computational efficiency of our implementation. Before presenting our modeling (section 3.2), we describe how Ortner builds the terms of its energy (section 3.1). Classically, it is made-up of a regularizing term (section 3.1.1) and a data attachment term (section 3.1.2).

3.1 Ortner's energy

3.1.1 Regularizing term

Ortner's energy for the internal field is decomposed into 4 different interactions, computed thanks to the relative positions of interacting rectangle corners. The first interaction regards *alignment* of objects. Based on 2 thresholds, it aims at favouring small angular differences between neighbouring objects. The associated energy is defined through a weighted sum of distances between the corners and the angular difference. The second term is related to *completion*. As the data term detects discontinuities along two opposite sides of the rectangle, its goal is to detect discontinuities in the orthogonal directions. It is also based on the previous thresholds and favours orthogonal orientations between the objects. Another term favours *paving* interactions, *i.e.* arrangements of objects along the side were the discontinuity lies. Finally, the last term aims at avoiding redundant objects, *i.e.*, objects with a high intersection surface.

Once all these terms are defined, the final external energy is made up of a weighted sum of the interactions. This involves 7 parameters.

3.1.2 Data attachment term

The global data term is defined as a sum of 4 energies, each one associated with a set of objects. To do so, a candidate rectangle is sliced orthogonally to the main direction to study profiles. Three measures are then used to compute the data term. First, a *hit length* counts the number of high gradients along profiles. A *volume rate* taking into account the number of profiles, their width

and length is also computed. Finally, a *moment rate* measures the average distance between a rectangle hypothesis and hit points.

These measures are based on physical parameters and thresholds. However, a lot of parameters generally imply a fine tuning. When the input data change, the parameters need to be tuned again individually and in interaction. It is also quite uneasy to weigh the relative importance of the 3 terms in the final data term. As for the internal field, 7 parameters are involved in the computation of the external field.

3.2 Proposed energy

We now present our choices to define in a simple and efficient fashion our energetic terms. To avoid parameters tuning and sensitive thresholds setting, we have drastically simplified the energetic terms. The remaining parameters are also related to physical quantities, which eases their tuning.

3.2.1 Regularizing term

The regularizing energy $\mathcal{U}_{\text{prior}}$, is used to favor some configurations of rectangles and to penalize some others. It requires the definition of a neighborhood system. Let us remind the general definition of a neighborhood. If $\sim$ is a symmetric relation on $\mathcal{P} \times \mathcal{M}$, the neighborhood $\mathcal{N}(u, \mathcal{C})$ of an object u in $\mathcal{C}$ is defined as the set of objects in relation with u:

$$\forall u \in \mathcal{C},\ \mathcal{N}(u, \mathcal{C}) = \{v \in \mathcal{C} \ \mid \ u \sim v,\ u \neq v\} \tag{4}$$

We choose to define the neighborhood system thanks to a geometric criterion. For an object $u \in \mathcal{C}$, it is simply the set of objects intersecting u:

$$u \sim v \Leftrightarrow u \cap v \neq \emptyset \tag{5}$$

This choice is really natural, because a single building can be represented by a single rectangle, that is to say, without any interacting object and thus an empty neighborhood. In the case of buildings blocks, the arrangement of rectangles required to represent such a structure is composed of contiguous or slightly overlapped objects (figure 1).

To obtain the desired configurations, our energy is very simple and has only a repulsive term. For a pair of neighboring objects u_i and u_j, it is simply the weighted intersection area $\mathcal{S}_{\text{inter}}$ of the associated rectangles o_i and o_j:

$$\mathcal{U}_{\text{prior}}\left(u_i, u_j\right) = \beta.\mathcal{S}_{\text{inter}}\left(o_i, o_j\right) \tag{6}$$

So, this regularizing term is only repulsive. β is a weighting parameter which tunes the importance of prior energy versus data energy. In our work, the attractiveness is only given through the data attachment term. Furthermore, in order to avoid accumulation of objects where the internal energy is favourable, the external energy defined through equation 6 is perfectly designed. Indeed, the higher the overlap area between objects is, the higher the energy is, and therefore, this kind of configuration tends to be rejected as the process evolves. Furthermore, intersection area is not normalized as we want to have a physical interpretation of the energy (and not a proportion); this is also interesting as small objects are less relevant than bigger ones (small ones should take into account the position of big ones).

We think that it is not necessary to design a complex system of interactions and, in fact, experience has proven that if the external term is well designed,

a simple internal term is sufficient and models the problem quite well. For instance, it is not necessary to introduce an alignment interaction, because the modeling of the data term already includes this property (section 3.2.2)
Finally, for a given configuration $\mathcal{C}$, the global internal energy is given by the sum of individual terms for each pair of neighboring objects:

$$\mathcal{U}_{\text{prior}}(\mathcal{C}) = \sum_{i \neq j} \mathcal{U}_{\text{prior}}(u_i, u_j) \tag{7}$$

3.2.2 Data attachment term

We now focus on the external energy term which aims at measuring the adequacy of objects with respect to the DEM. This section explains the choices we made in order to detect building footprints, that is to say how we define an attractive object.
As we aim at finding building footprints from DEMs, which are rasterized height data, we are naturally looking for altimetric discontinuities. The higher the discontinuity is, the more likely the probability to find the edge of a building at this place is. From this, we obviously use a measure based on the DEM gradient and define it in order to align rectangles edges with the highest height variations in the DEM. Large height variations are represented by high values of the gradient magnitude. To test if a rectangle matches DEM discontinuities, we use the dot product of the gradient vector and the normal vectors attached with the edges of the rectangle.
Considering a rectangle edge segment $[a, b]$, its outgoing normal $\overrightarrow{n} = \frac{\overrightarrow{ab}^{\perp}}{\|ab\|}$ from a to b (where $^{\perp}$ denotes a $\frac{\pi}{2}$ vector rotation), its external energy is:

$$\mathcal{U}_{\text{data}}(a, b) = \int_0^{\|ab\|} \left(\overrightarrow{n} \cdot \overrightarrow{\nabla} \left(a + s \frac{\overrightarrow{ab}}{\|ab\|} \right) \right) ds \tag{8}$$

where $\overrightarrow{\nabla}(p)$ is the gradient magnitude at pixel p. This can be reformulated, using orthogonal vectors: $\overrightarrow{\nabla} \cdot \overrightarrow{n} = \overrightarrow{\nabla}^{\perp} \cdot \overrightarrow{n}^{\perp} = \overrightarrow{\nabla}^{\perp} \cdot \frac{\overrightarrow{ab}}{\|ab\|}$.

$$\mathcal{U}_{\text{data}}(a,b) = \overrightarrow{ab} \cdot \left(\int_0^1 \overrightarrow{\nabla}\left(a + s\ \overrightarrow{ab}\right) ds \right)^{\perp} \tag{9}$$

For efficiency, the integral is discretized using a nearest neighbor scheme, where the (s_i) correspond to the segment $[a,b]$ crossing a pixel boundary at a point $p_i = a + s_i\overrightarrow{ab}$, such that s_i is increasing, $s_0 = 0$ and $s_{n+1} = 1$. $\overrightarrow{\nabla}_i$ denotes the constant value approximation of $\overrightarrow{\nabla}$ along the segment $[p_i, p_{i+1}]$ (see Figure 3):

$$\mathcal{U}_{\text{data}}(a,b) = \overrightarrow{ab} \cdot \left(\sum_{i=0}^{n} (s_{i+1} - s_i)\, \overrightarrow{\nabla}_i \right)^{\perp} \tag{10}$$

$\mathcal{U}_{\text{data}}(a,b)$ is the dot product of the edge vector $\overrightarrow{ab}$ and the average gradient along the edge $\int \overrightarrow{\nabla}$. It is thus linear with the edge length, and with the mean gradient magnitude (corresponding to the average façade height), yielding a measure of a façade surface. Note that the integral formulation and the discretized $(s_{i+1} - s_i)$ weighting ensures that the measure is isotropic. It further takes into account that the orientations of the vectors to only measure the gradient outgoing the rectangle edges. It thus favours rectangle edges alignment with large height variations in the DEM. For an object u_i, its external energy is the sum of the external energies of the four individual segments $[p_i^j, p_i^{j+1}]$ subtracted from a constant w_{data}:

$$\mathcal{U}_{\text{data}}(u_i) = w_{\text{data}} - \sum_{j=0}^{3} \max\left(0, \mathcal{U}_{\text{data}}\left(p_i^j, p_i^{j+1}\right)\right) \tag{11}$$

This constant w_{data} is used in order to give to each rectangle an initial weight. This weight can be considered as the minimal façade surface required for a building to be detected and it depends on the desired level of generalization. The data term thus becomes negative when there is enough gradient across the

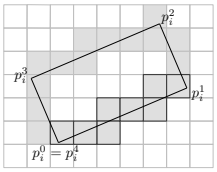

Fig. 3. Explanation of the external energy for a segment $[p_i^j, p_i^{j+1}]$ of a given rectangle. The grid stands for the DEM pixels; the segment $[p_i^j, p_i^{j+1}]$ begins at p_i^j, ends at p_i^{j+1} and crosses 8 pixels of the DEM (represented with a dark grey outline).

rectangle edges. However, setting this parameter, as it has a straightforward physical interpretation, is easier than setting an obscure weighting parameter. In this sense too, our energy is simple and easy to compute.

Another notable aspect of this energy is that, using the $\max(0, \cdot)$ expression, only segments of the rectangles that have a positive discontinuity support $\mathcal{U}_{\text{data}}(a, b)$ are taken into account, *i.e.*, the orientation of the discontinuity must match the segment outgoing normal. Finally the global external term of the energy for a configuration $\mathcal{C}$ is:

$$\mathcal{U}_{data}(\mathcal{C}) = \sum_{\{u_i\} \in \mathcal{C}} \mathcal{U}_{data}(u_i) \tag{12}$$

3.3 Contributions of our energy definition

Before ending this section, we would like to sum up the contribution of our work with regard to that of Ortner. This simply consists in comparing our energies, the overall number of parameters to adjust and their meaning.

Our regularizing term is only based on the overlapping area of the neighboring objects. This is a real simplification since this formulation has a physical meaning. Thus, our *prior* energy is only repulsive and aims at avoiding superposition of objects. In fact, it is not necessary to develop a complex system of interactions, because if the data term is well designed, it fulfills the attractive

part of the global energy. However, as buildings are not composed of parts perfectly aligned or perpendicular, slightly overlapping objects must not be too much penalized. This is the purpose of the β parameter defined in equation 6. It allows to tune, with respect to the surface of the intersection area, the level of penalization for an overlapping configuration. More precisely, it weights the overlapping surface versus the façade surface. So a value of 10 means that a rectangle needs $10\,m^2$ of façade to compensate $1\,m^2$ of overlapping.

The other part of the global energy, the data attachment term is only based on a simple observation: building edges are orthogonal to the high gradient of the DEM. Thus, it can be simply defined as a quantity depending on the amount of gradient orthogonal to each edge of the rectangles. Moreover, we also have simplified the definition and the tuning of the parameters since there is only one physical parameter, w_{data}, which allows to tune the level of generalization. The necessary highlighted points in Ortner et al. (2007) for its data term are:

- an object is attractive if it has enough gradient along its edges,
- a non attractive object is slightly repulsive,
- the better the object is aligned with discontinuities, the better its energy is.

In our implementation, they are all respected with an efficient to compute and simple definition of the energy. Moreover, both terms express areas related to building: its footprint and its façade surface. We thus do not have heterogeneous quantities which are uneasy to compare and to weigh each other.

4 Implementation

Previous sections were dedicated to an energy-based definition of our marked point process for buildings outline detection. We now express our choices for its effective and generic implementation which relies on two crucial points: data-processing structure and geometric optimization.

Although we are looking for an effective approach, we want it to be totally generic as we use marked point process in other contexts. Based on Boost libraries, the code is portable and fully C++ compliant. No assumptions have been made on the geometric properties of objects (this code would work for segments or circles) nor on the dimensions of the exploration space (this code would work for studying a point cloud). Changing an energy definition only requires to code the corresponding method. The total code size is less than 1 500 lines, which makes it rather easy to understand and use.

4.1 Proposition kernels

We first detail the simple transition kernels we use. The first ones, birth and death kernels are necessary to change to higher or lower dimension and guarantee that the Markov Chain visits the whole configuration space (Geyer and Møller, 1994). Birth (*resp.* death) simply consists in adding (*resp.* removing) a random object from (*resp.* to) the current configuration.

The perturbation kernel, allows to modify the mark of an object. Although it is not necessary to ensure convergence, its use is strongly recommended in the optimization process to avoid consecutive birth and death steps. It not only

increases computational efficiency but also enables to finely position objects at the end of the process when all of them are detected but just need to be adjusted in shape or position.

Our implementation uses only 2 perturbations. The left perturbation of fig-

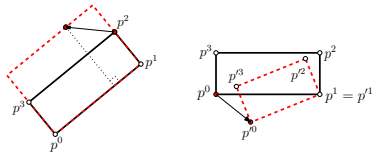

Fig. 4. Examples of some perturbations of a rectangle (the original rectangle is in black, the transformed one is in dashed red). (left) Translation of an edge. (right) Combination of rotation and similitude around one of the corner. The translation vector (see section 4.3.1) is applied to the red corner.

ure 4 translates an edge of the rectangle: 2 corners remain fixed, while the center and the aspect ratio are modified. The right perturbation combines a rotation and a similarity around a rectangle corner: the chosen corner and the aspect ratio remain fixed whereas the center and the 3 other corners are modified.

4.2 Data-processing structure

Remember that an object configuration is made up of interacting objects, *i.e.*, each object has a set of neighbours (which could be empty). The Markov property naturally leads to use a graph as data-processing structure to store a configuration and maintain the $\sim$ relation between the objects (see figure 5). Indeed, an object u_i of a configuration can be seen as a node, and a neighborhood relation between 2 objects is easily represented by an undirected edge

(the $\sim$ relation is symmetric).

In a more detailed way, we use a valued graph, *i.e.*, each edge carries the reg-

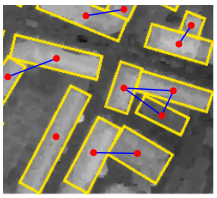

Fig. 5. The neighborhood system. Each rectangle is associated to a node (red). Each rectangle has a set of neighbours connected by the graph's edges (blue). ©IGN 2009

ularizing energy. From the nodes point of view, each has a set of attributes: the associated object, *i.e.*, its position in image coordinates, marks and data energy. Given such a structured graph, we exactly have all the required information about the associated configuration.

On this matter, one of the main advantage of the graph structure is that we do not need to recompute the whole energy at each modification of the configuration: only local changes are sufficient. The data term is only impacted by the external energy of the node which undergoes the modification (one addition (*resp.* subtraction) for birth (*resp.* death), an addition and a subtraction for perturbation); the regularizing term is updated thanks to the new energies carried by the edges adjacent to the impacted node and the possible new or removed edges (new interacting objects or objects not anymore interacting).

4.3 Geometry

The computational bottleneck of the algorithm is its geometric routines. We thus have to find an optimized parametrization of the rectangle to be efficient.

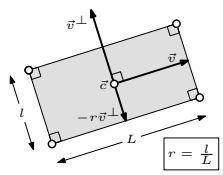

Fig. 6. The rectangle $(c, \overrightarrow{v}, r)$.

4.3.1 Rectangle representation

A rectangle is represented (figure 6) by its center point c, a vector $\overrightarrow{v}$ from the center to the middle of one of its smaller sides and its aspect ratio r (*i.e.* the ratio of the length l of the side orthogonal to $\overrightarrow{v}$ to the length L of the side parallel to $\overrightarrow{v}$). Contrary to the usual representation with the center, an orientation angle and the two side lengths L and l, which is also minimal, our representation avoids the time consuming square roots and trigonometric functions, limits the use of divisions, and otherwise only use additions, subtractions and multiplications, which is the key of an efficient handling of 2D rectangles. Using the notations $v^2 = \overrightarrow{v} \cdot \overrightarrow{v}$ and $\overrightarrow{v}^{\perp} = (-v_y, v_x)$ for a $\frac{\pi}{2}$ rotation:

$$\text{corners} = c \pm \overrightarrow{v} \pm r\,\overrightarrow{v}^{\perp} \quad \text{and} \quad \text{area} = rv^2$$

4.3.2 Rectangle Perturbations

The proposed representation enables low cost perturbations that do not require any square root or trigonometric operation. Pure translations affect only c, while combinations of rotations and isotropic scaling only involve $\overrightarrow{v}$. The proposed perturbations involve translating a rectangle edge while keeping its opposite edge fixed, and a composition of a isotropic scaling and a rotation of the rectangle around one of its corners as illustrated on figure 4. They are conveniently performed on a rectangle $(c, \overrightarrow{v}, r)$ using the following transfor-

mations, where $(\varepsilon, \varepsilon') \in \{-1, 1\}^2$ selects the fixed corner or edge :

$$\text{Rotation/scaling} \rightarrow (c + \varepsilon \overrightarrow{w} + \varepsilon' r \overrightarrow{w}^{\perp}, \overrightarrow{v} + \overrightarrow{w}, r) \tag{13}$$

$$\text{Edge translation} \underset{\varepsilon'=-1}{\rightarrow} (c + \varepsilon(t-1)\overrightarrow{v}, t\overrightarrow{v}, \frac{r}{t}) \tag{14}$$

$$\underset{\varepsilon'=1}{\rightarrow} (c + \varepsilon(t-1) r \overrightarrow{v}^{\perp}, \overrightarrow{v}, rt) \tag{15}$$

Note that translating an edge using one of the equations above may yield an aspect ratio $r > 1$. This is simply fixed by using the alternative equivalent representation $(c, r\overrightarrow{v}^{\perp}, \frac{1}{r})$. The translation perturbation moves 2 corners by $2\varepsilon(t-1)\overrightarrow{v}$ or $2\varepsilon(1-t)r\overrightarrow{v}^{\perp}$ and the rotation/scaling one moves the 3 unfixed rectangle corners by $2\overrightarrow{w}$ and/or $2r\overrightarrow{w}^{\perp}$. It is then straightforward to limit the magnitude of the corner translations. This upper bound on the corner translation magnitude is expressed in meters, and is therefore not sensible to changes in resolution. The experiments we carried out have shown that these perturbations are sufficient, so that, for instance, pure rotations and translations are not needed.

4.3.3 Intersection test

The proposed representation allows to efficiently test whether a point $\overrightarrow{p}$ is inside a rectangle $(c, \overrightarrow{v}, r)$. First, the point is expressed in a frame centered at c and oriented along $\overrightarrow{v}$: $x = (\overrightarrow{p} - c) \cdot \overrightarrow{v}$ and $y = (\overrightarrow{p} - c) \cdot \overrightarrow{v}^{\perp}$. Then $\overrightarrow{p}$ is inside the rectangle if and only if $|x| \leq v^2$ and $|y| \leq rv^2$. The energy functional needs to check efficiently whether 2 rectangles R_1 and R_2 intersect and, if so, compute their intersection area. Since rectangles are convex polygons, the *Separating Axis Theorem* states that they do not intersect if and only if a line supporting one of the 8 sides of the 2 rectangles separates the 2 rectangles. A naive implementation tests, for each side s of a rectangle, whether the 4 points

of the other rectangle are on the other side of the supporting line of s. The symmetries of the problem allow a more efficient implementation: computing only once the vector $\overrightarrow{c_{12}} = c_2 - c_1$ between rectangle centers, the absolute values: $\lambda = |\overrightarrow{v_1} \cdot \overrightarrow{v_2}|$, $\mu = |\overrightarrow{v_1} \cdot \overrightarrow{v_2}^{\perp}|$, and the squared norms v_1^2 and v_2^2 (the latter 2 quantities may be cached in each rectangle). For instance, the corners of R_2 are on the side of the line segment $[c_1 + \overrightarrow{v_1} - r_1\overrightarrow{v_1}^{\perp}, c_1 + \overrightarrow{v_1} + r_1\overrightarrow{v_1}^{\perp}]$ opposite to R_1 if and only if $\forall(\varepsilon, \varepsilon') \in \{-1, 1\}^2$:

$$\left(c_2 + \varepsilon\overrightarrow{v_2} + \varepsilon' r_2 \overrightarrow{v_2}^{\perp}\right) \cdot \overrightarrow{v_1} > (c_1 + \overrightarrow{v_1}) \cdot \overrightarrow{v_1}$$
$$\Leftrightarrow \overrightarrow{c_{12}} \cdot \overrightarrow{v_1} > v_1^2 + \lambda + r_2\mu \quad (16)$$

The opposite edge of R_1 yields the similar expression $(-\overrightarrow{c_{12}}) \cdot \overrightarrow{v_1} > v_1^2 + \lambda + r_2\mu$, so that they can be tested together using the absolute value $|\overrightarrow{c_{12}} \cdot \overrightarrow{v_1}|$. Finally, the rectangle intersection is tested using only 4 quantities, for separating axes along $\pm\overrightarrow{v_1}$, $\pm\overrightarrow{v_2}$, $\pm\overrightarrow{v_1}^{\perp}$ and $\pm\overrightarrow{v_2}^{\perp}$:

$$\begin{array}{rcrcl} \text{Intersects}(R_1, R_2) & = & |\ \overrightarrow{c_{12}} \cdot \overrightarrow{v_1}\ | & \leq & v_1^2 + \lambda + r_2\mu \\ & & \& \ |\ \overrightarrow{c_{12}} \cdot \overrightarrow{v_2}\ | & \leq & v_2^2 + \lambda + r_1\mu \\ & & \& \ \left|\overrightarrow{c_{12}} \cdot \overrightarrow{v_1}^{\perp}\right| & \leq & r_1 v_1^2 + \mu + r_2\lambda \\ & & \& \ \left|\overrightarrow{c_{12}} \cdot \overrightarrow{v_2}^{\perp}\right| & \leq & r_2 v_2^2 + \mu + r_1\lambda \end{array} \quad (17)$$

The efficiency of this rectangle intersection test is measured by the low number of operations involved and more importantly the absence of square roots or trigonometric operations (table 1).

Table 1

Cumulative operation counts, assuming (v_i^2) and $(r_i v_i^2)$ values are cached, which saves 2 additions and 6 multiplications. The intersection test returns as soon as a separating axis is found, so that the exact operation count among the last 4 columns depends on the which axis, if any, is found to be a separating axis.

Operation counts	$\lambda, \mu, \overrightarrow{c_{12}}$	$\overrightarrow{v_1}$?	$\overrightarrow{v_2}$?	$\overrightarrow{v_1}^{\perp}$?	$\overrightarrow{v_2}^{\perp}$?
$\lvert \cdot \rvert$ absolute values	2	3	4	5	6
$+$ additions	4	7	10	13	16
$\times$ multiplications	4	7	10	13	16

4.3.4 Intersection area

To compute the intersection area of 2 rectangles R_1 and R_2, the bounding box R_3 of R_2 aligned with R_1 is considered. Figure 7 shows how R_3 may be partitioned into 4 right triangles T_i and the rectangle R_2. The area of intersection between 2 aligned rectangles R_1 and R_3 is trivial. Likewise, the ones between the rectangle R_1 and a right triangle T_i, which edges adjacent to the right angle are aligned with R_1 may be computed efficiently, using techniques similar to the rectangle intersection test of the previous section. $(R_2, T_1, T_2, T_3, T_4)$ being a partition of R_3, the intersection area of R_1 and R_2 is computed as:

$$\text{area}(R_1 \cap R_2) = \text{area}(R_1 \cap R_3) - \sum_{i=1}^{4} \text{area}(R_1 \cap T_i) \tag{18}$$

If $\text{area}(R_1 \cap R_3) = 0$ (R_1 and R_3 do not intersect) or if $\text{area}(R_1 \cap R_3) - \sum_{i=1}^{k} \text{area}(R_1 \cap T_i)$ vanishes before $k = 4$, then the $\text{area}(R_1 \cap T_i)$ terms for $i > k$ have to be zero and their computation is thus avoided, yielding a faster answer to simpler (more separated) problems.

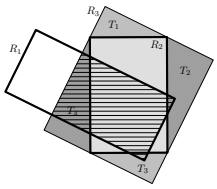

Fig. 7. Intersection area of R_1 and R_2 is eased by considering the bounding box R_3 of R_2 aligned with R_1.

4.3.5 Segment iteration

Figure 8 illustrates that the segment length between 2 vertical (*resp.* horizontal) line crossings is a constant dt_x (*resp.* dt_y). The iteration over the pixels crossed by a segment can be computed efficiently by computing various quantities as a preprocess. For instance, after crossing a vertical line, the next vertical line crossing is computed by adding dt_x. The next line crossed is along the direction that has the closest line crossing length t.

Such a scheme is particularly interesting to compute efficiently the data term

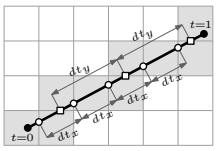

Fig. 8. Iterating over the pixels crossed by a segment.

of the energy.

5 Experimental results

In our experiments, we used aerial (50 cm GSD [1] ; figure 14) satellite (50 cm GSD; figure 10) and lidar (50 cm GSD; figure 11) DEMs. Satellite and aerial DEMs were obtained with a graph-cut optimization approach in a multiview framework

[1] Ground Sampling Distance

(Roy and Cox, 1998; Pierrot-Deseilligny and Paparoditis, 2004). Images were acquired by the 16 mega-pixels *Institut Géographique National* (IGN) digital camera. Inter and intra stripes overlaps are about 60%. The Toposys® lidar scanner was used for the acquisition of 3D lidar point clouds. DEMs have then been computed from the 3D lidar points using a simple rank filter. In this section, all given computation times were obtained on a linux Ubuntu 8.10 PC, with an Intel® Core™ 2 Duo CPU T7700 @ 2.50GHz.

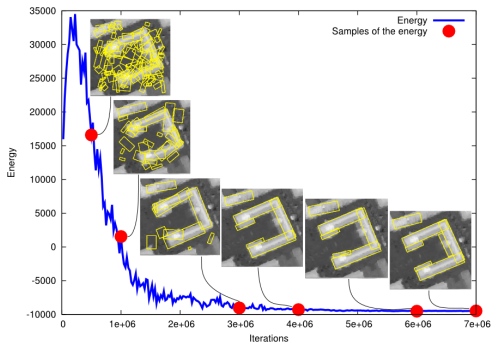

Fig. 9. The blue curve shows the evolution of the energy. The samples denoted by red points correspond to the superimposed configurations.

5.1 Qualitative results

We first illustrate the evolution of the detection at different steps of the algorithm on a sample building (figure 9). At the beginning of the process, there are many objects, not well located. 500 000 iterations later (a few seconds), some objects begin to find their correct location. Then, as long as the temperature decreases, more and more objects are accurately positioned, and finally, all objects are well detected and located, until convergence. The associated energy graph is presented on the same figure with highlighting representative

samples.

Figure 10 compares our results with Ortner's ones on a bigger area. In both

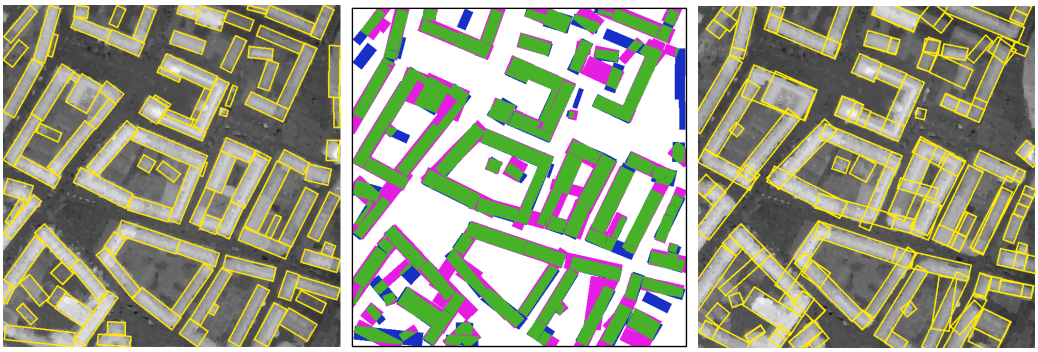

Fig. 10. Comparison of the results obtain with our strategy (left) and with Ortner's strategy (right). The middle image is a direct comparison: in blue, our results alone, in magenta, Ortner's results alone, in green the common part of both results. 41% of the pixels are classified as buildings with our approach, while Ortner's one gives a result of 46%.

cases, all building parts are well detected and rectangle edges correspond most of the time to true buildings limits. False alarms exist in both results and are mainly due to tree alignments or trees in courtyards, but, could be avoided for instance by computing a vegetation mask based on NDVI index (Iovan et al., to appear). Both approaches also exhibit that some rectangles straddle the streets or embrace 2 buildings. However, this is not prejudicial, because a further step can deal with these cases to obtain a more relevant description of the buildings by splitting existing footprints. More generally our results tends to be less broken up, that is to say that for a given building part, we have only one rectangle, whereas Ortner's results tend to superimpose some rectangles or divide a coherent building part in several rectangles. This is due to the attractive paving interaction that we removed from our regularization term formulation. The last point to notice concerns the undetected buildings. Most of the time, they are the small ones in inner courtyards: the height disconti-

nuity between the roof and the ground is too weak to be detected. They also exist in our approach when a building part is too small ("9"-shaped building at the center of the image) and can be explained by a too high value of w_{data}.

Figure 11 shows a result on some common buildings as the ones of figure 10.

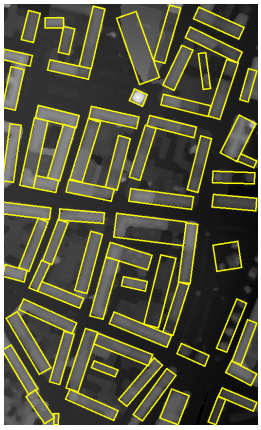

Fig. 11. Result obtained with a lidar DEM ©IGN 2009. The result is completely comparable with the result of the satellite DEM (figure 10) on the common buildings (around the "9"-shaped building).

However, the DEM comes from an aerial based lidar point cloud (discontinuities are sharper and there is less noise on the ground). The results are not significantly different, which demonstrates the ability of the algorithm to successfully operate on various data sources. The aforementioned problems are also encountered: missing small buildings, straddling of the streets, single rectangle for multiple buildings, trees, ... In a more detailed way, figure 12 compares both results on a particular building. They are slightly different, but can be considered as equivalent since nearly the same discontinuities are detected.

The last test is on a building with a complex shape (figure 13). Indeed, it is not regular and only a few parts are rectangular, while the higher part is curved.

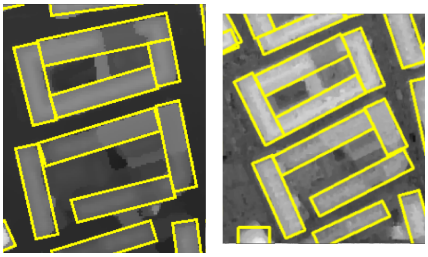

Fig. 12. Comparison of the results obtained with a lidar (left) and an satellite DEM (right) ©IGN 2009

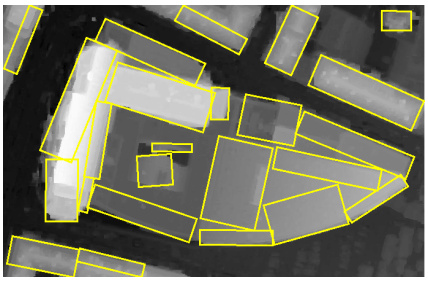

Fig. 13. Comparison of both approaches on a complex building ©IGN 2009

5.2 Quantitative results

Figures 14 and 15 are detailed areas on an aerial DEM with a 20 cm GSD (St-Mandé, France). We used the result presented on figure 14 as a basis for a comparison with a ground truth (coming from a digitized cadastral map) and with Ortner's results. Both approaches are compared in table 2 using classical indicators [2]. They show that both compared approaches are equivalent. The high false alarm rate can mostly be explained by the detection of the trees, which can be avoided with a vegetation mask. Indeed, we do not have such a mask and the ground truth is more accurate than the DEM itself. Thus, these results are relevant for comparing both methods, but not to evaluate

[2] detection rate $= \frac{TP}{total}$, false alarm rate $= \frac{FP}{total}$ and quality rate $= \frac{TP}{FP+total}$, where TP denotes pixel-wise *true positive*, TP *false positive* and *total* is the number of building pixels in the reference.

the general quality of the approach. To test the reliability of the method, we have manually removed the rectangles corresponding to trees in our result (last line of table 2). False alarm and quality rates are clearly improved. The small decrease of the detection rate corresponds to a rectangle which embraced trees and small buildings in the original result.

On the building block presented on figure 15, our approach successfully detects

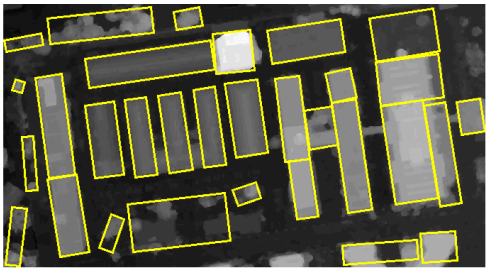

Fig. 14. Result obtained with a 20 cm aerial DEM ©IGN 2009.

its main parts. However, some small parts and some other with a low height above the ground are undetected. This can be explained by the definition of our data energy which measures the total amount of gradient outgoing the rectangle edges. The small parts that our approach does not detect do not have enough gradient. If w_{data} is decreased, buildings will be detected but as an unexpected aggregate of multiple small rectangles. However, if we accept a small generalization, the results are still reliable.

5.3 Timing and parameters

Table 3 compares computation times of both approaches for the presented results. Our approach clearly provides faster results. The ratio can achieve a factor of 100 in our favour. We have also run a test on a 1 km^2 area to have an idea of the computation time needed for processing a whole city. The computation complexity is more dependant on the number of objects than on the

Table 2

Quantitative comparison between our work and Ortner's approach on the result presented on figure 14. Results are clearly equivalent.

	Rates (%)		
	Detection	False alarm	Quality
Our methodology	86.1	34.9	63.8
Ortner's methodology	88.3	45.8	60.5
Our methodology (without trees)	83.8	14.3	73.3

Fig. 15. Result obtained with a 20 cm aerial DEM ©IGN 2009. See also table 2.

image size. However, such a large area contains enough buildings to provide a relevant idea of what we could expect while processing a whole city. In our test, we used a 50 cm GSD aerial DEM (2000×2000 pixels) on a dense urban area with almost 1100 objects. The result was obtained in about 50 minutes.

In our implementation, only a few parameters have to be set. Regarding the energies, the weighting β of the intersection area between 2 objects (equation 6) has always been set to 10 in our tests. A higher value avoids intersection but does not change the final result: objects overlap less, but the general layout of the results is similar (see figure 16).

Table 3

Computation times of the compared approaches. Our methodology provides a gain factor varying from 80 to 100.

Image size	♯ objects	Computation times	
		Our methodology	Ortner's methodology
654×665 (figure 10)	≃100	≃180 s	≃4 h
435×263 (figure 11)	≃70	≃60 s	N/A
758×1359 (figure 14)	30	≃100 s	≃3 h
978×1149 (figure 15)	11	≃100 s	≃3 h

A more important parameter is w_{data} (equation 11) which gives the minimal

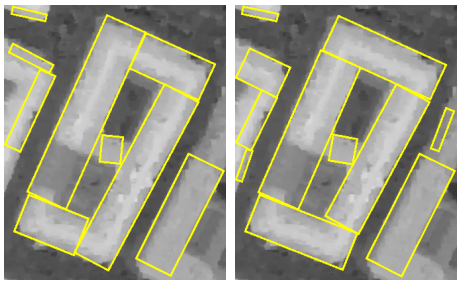

Fig. 16. Influence of β parameter (left, $\beta = 10$; right, $\beta = 100$). Results are strictly equivalent in terms of energy, even if the layout is not the same.

façade surface that a building must have to be detected by our algorithm. Indeed, it must be large enough to avoid the detection of very small structures mainly due to noise in the DEM, or isolated trees. However, if it is too large, it is more robust to the previous problems, but, small buildings parts will not be detected. The effects of different values of w_{data} on the results are presented on figure 17.

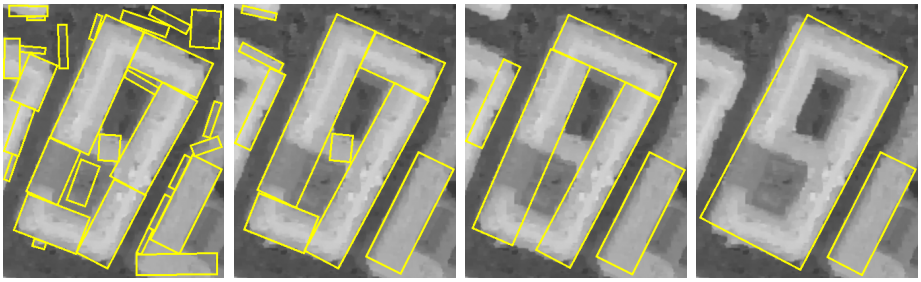

Fig. 17. Influence of w_{data} parameter. From left to right and top to bottom, w_{data} value is 25, 100, 250 and 1000. For a too small value, too many rectangles are detected (noise in the DEM), and, as the value increase, the level of generalization becomes higher.

6 Conclusion and future works

We have presented in this paper a reliable method to extract building footprints from DEM. Our aim is to provide an efficient implementation with regards to the quality of the results, the computation time and the simplicity of parametrization. Results and their associated computation times clearly show that the goals are reached.

This efficiency relies on 2 aspects. The first one is the definition of the energy to minimize, the second one are the choices and the efforts made in the implementation. Regarding the energy, we choose a very simple definition while protecting the essential aspects allowing to describe what a building footprint in a DEM is. The simplicity of our definitions leads to very efficient computation times and simple parametrization, what is our primary goal. The efficiency of our approach also comes from our implementation of the algorithm. On the one hand, we have an original parametrization of the base object of the process, the rectangle, allowing to transform each object with a minimum of operations and avoiding the high costs of square roots and trigonometric functions. On the other hand, the underlying graph structure of a configura-

tion allows to effectively apply the changes involved in the algorithm.

It is also important to note that the both terms of our energy are homogeneous in the sense that they both measure a surfacic quantity. This gives our formulation an easy interpretation and avoids mixing several quantities such as angles, distance or areas.

Extracting footprints is the first step towards building 3D reconstruction. However, there is between these two stages an intermediate step in order to obtain a fine representation of the roof ridge tiles. An approach has already been presented in Lafarge et al. (2006) which consists in detecting discontinuities in the rectangles and merging rectangles into connected polygons. However, the algorithm is only based on the geometric properties of the objects. This could be improved by using the DEM to find polygon edges corresponding to DEM discontinuities.It could also be interesting to improve the simulated annealing parametrization. In fact, at the beginning of the process, when temperature is high, most of the moves are accepted. As long as the process evolves, birth and death are less and less accepted, and perturbations become dominant. To further improve computation times and propose only relevant modifications of the configuration at the end of the process, we could detect when the optimization only consists in finely positioning the objects (for example when the acceptation ratio are small enough for both birth and death on the past iterations). Hence, once this step detected, we will be able to propose only data-driven modifications, *i.e.*, objects close to already detected objects, without intersection. We think this approach could be efficient for our purpose. Stochastic annealing or jump diffusion could also be interesting orientations.

Acknowledgments

The authors would like to thank J. Zerubia and M. Ortner (ARIANA / INRIA) for providing the binary used to compare with our results. They also express their gratitude to M. Durupt (MATIS / IGN) for the tim she spent running tests and comparisons.

References

Alharthy, A., Bethel, J., 2002. Heuristic filtering and 3D feature extraction from LiDAR data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Science 34 (Part 3), 29–34.

Brunn, A., Weidner, U., 1997. Extracting buildings from digital surface models. International Archives of Photogrammetry and Remote Sensing 32 (Part 3/4W2).

Cressie, N. A. C., 1993. Statistics for spatial data. Probability and Mathematical Statistics. Wiley-Interscience, New-York, USA.

Frédéricque, B., Daniel, S., Bédard, Y., Paparoditis, N., 2008. Populating a building multi-representation data base with photogrammetric tools: recent progress. ISPRS Journal of Photogrammetry and Remote Sensing 64 (4), 441–460.

Geyer, C. J., Møller, J., 1994. Simulation and likelihood inference for spatial point processes. Scandinavian Journal of Statistics 21 (4), 359–373.

Green, P. J., 1995. Reversible Jump Markov Chain Monte Carlo computation and Bayesian model determination. Biometrika 82 (4), 711–732.

Haala, N., Brenner, C., Anders, K. H., 1998. 3D urban GIS from laser altimeter

and 2D map data. International Archives of Photogrammetry and Remote Sensing 32 (3), 339–346.

Haithcoat, T. L., Song, W., Hipple, J. D., 2001. Building footprint extraction and 3D reconstruction from LiDAR data. In: Proc. of the Remote Sensing and Data Fusion over Urban Areas. IEEE/ISPRS, Roma, Italy. 8-9 November 2001, pp. 74–78.

Hastings, W. K., 1970. Monte Carlo sampling using Markov chains and their applications. Biometrica 57 (1), 97–109.

Iovan, C., Boldo, D., Cord, M., to appear. Detection, characterization and modeling vegetation in urban areas from high resolution aerial imagery. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

Lacoste, C., Descombes, X., Zerubia, J., 2005. Point processes for unsupervised line network extraction in remote sensing. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (10), 1568–1579.

Lafarge, F., Descombes, X., Zerubia, J., Pierrot-Deseilligny, M., 2006. A structural approach for 3D building reconstruction. Research Report 6048, INRIA.

Lafarge, F., Descombes, X., Zerubia, J., Pierrot-Deseilligny, M., 2008. Building reconstruction from a single DEM. In: Proc. of the Conference on Computer Vision and Pattern Recognition. IEEE, Anchorage, AK, USA. 24-26 June 2008.

Ortner, M., Descombes, X., Zerubia, J., 2007. Building outline extraction from digital elevation models using marked point processes. International Journal of Computer Vision 72 (2), 107–132.

Perrin, G., Descombes, X., Zerubia, J., 2006. 2D and 3D vegetation resource parameters assessment using marked point processes. In: Proc. of the In-

ternational Conference on Pattern Recognition. IAPR, Hong-Kong, China. 20-24 August 2006.

Pierrot-Deseilligny, M., Paparoditis, N., 2004. A multiresolution and optimization-based image matching approach: An application to surface reconstruction from SPOT5-HRS stereo imagery. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 1/W41) (on CD-ROM), http://www.micmac.ign.fr (accessed 15.09.2009)).

Robert, C. P., Casella, G., 2004. Monte Carlo statistical methods, 2nd Edition. Springer texts in statistics. Springer.

Roy, S., Cox, I. J., 1998. A maximum-flow formulation of the N-camera stereo correspondence problem. In: Proc. of the International Conference on Computer Vision. IEEE, Bombay, India. 4-7 January 1998, pp. 492–499.

Salamon, P., Sibani, P., Frost, R., 2002. Facts, conjectures and improvements for simulated annealing. Society for Industrial and Applied Mathematics, Philadelphia, USA.

Shackelford, A. K., Davis, C. H., Xiangyun Wang, X., 2004. Automated 2D building footprint extraction from high-resolution satellite multispectral imagery. In: Proc. of the Geoscience and Remote Sensing Symposium, 2004. Vol. 3. IEEE, Anchorage, AK, USA. 20-24 September 2004, pp. 1996–1999.

Stoyan, D., Kendall, W. S., Mecke, J., 1996. Stochastic Geometry and its Applications, 2nd Edition. Wiley-Interscience.

van Lieshout, M. N. M., 2000. Markov point processes and their applications. Imperial College Press, London, England.

Wang, O., Lodha, S. K., Helmbold, D. P., 2006. A Bayesian approach to building footprint extraction from aerial LiDAR data. In: Proc of the International Symposium on 3D Data Processing, Visualization, and Transmission.

IEEE, Chapel Hill, USA. 14-16 June 2006, pp. 192–199.

Winkler, G., 2003. Image analysis, random fields and Markov Chain Monte Carlo methods: a mathematical introduction, 2nd Edition. Springer-Verlag.