Some contributions to geometric modeling of urban environments

Florent Lafarge

Some contributions to geometric modeling of urban environments

Habilitation thesis

Florent Lafarge
Inria Sophia Antipolis - Méditerranée

Defended on 29^{th} of september, 2014

Jury

Matial Hebert, Carnegie Mellon University, United States	Reviewer
Konrad Schindler, ETH Zurich, Switzerland	Reviewer
Michael Wand, Utrecht University, The Netherlands	Reviewer
Pierre Alliez, INRIA Sophia Antipolis, France	Examinator
Jean-Daniel Boissonnat, INRIA Sophia Antipolis, France	Examinator
Luc Robert, Autodesk, France	Examinator

Abstract

This habilitation thesis proposes a series of contributions in the field of geometric modeling from physical measurements. These contributions present concepts and algorithms for object extraction, surface reconstruction, and scene modeling for urban environments. The methodology behind these contributions relies on stochastic geometry and graphical models. These probabilistic models are suited to analyzing the diversity and the complexity of urban objects and to exploring large and highly non-convex solution spaces at city-scales. The physical measurements, data structures, and concepts involved in our work lie between computer vision, geometry processing, and photogrammetry. A general summary also provides a vision of the many remaining challenges in the field.

Contents

1 Introduction **4**
1.1 Motivations and challenges 4
1.2 Approaches and trends 6

2 Object extraction **9**
2.1 Point processes 10
2.2 Objects and interactions 14
2.3 Sampling 17
2.4 Some applications 23

3 Surface reconstruction **30**
3.1 Hybrid surfaces 31
3.2 Primitive extraction 32
3.3 Reconstruction by point set structuring 37
3.4 Reconstruction by multiple shape sampling 44

4 Large-scale city modeling **49**
4.1 General strategy 50
4.2 Airborne Lidar 51
4.3 Airborne imagery 58

5 Conclusion **68**
5.1 Summary 68
5.2 Perspectives 69

References **71**

1 Introduction

A computerized shape representation can be visualized (creating a realistic or artistic depiction), simulated (anticipating the real) or realized (manufacturing a conceptual or engineering design). This constitutes the three main goals of geometric modeling and processing. Aside from the mere editing of geometry, central research themes in geometric modeling involve conversions between physical (real), discrete (digital), and mathematical (abstract) representations.

Geometric modeling has become an indispensable component for computational and reverse engineering. Simulations are now routinely performed on complex shapes issued not only from computer-aided design but also from an increasing amount of available measurements. The scale of acquired data is quickly growing: we no longer deal exclusively with individual shapes, but with entire *scenes*, with many objects defined as structured shapes. We are witnessing a rapid evolution of the acquisition paradigms with an increasing variety of sensors.

1.1 Motivations and challenges

In recent years, the evolution of acquisition technologies and methods has translated into an increasing overlap of algorithms and data in computer vision, computer graphics, remote sensing and robotics communities. Beyond the rapid increase of resolution through technological advances in sensors, the line between laser scan data and photos is getting thinner. Combining, *eg* mobile laser scanners with panoramic cameras leads to massive 3D point sets with color attributes (on the order of 200M points per kilometer in a urban street). In addition, it is now possible to generate dense point sets not just from laser scanners but also from photogrammetry techniques when matching an acquisition protocol. Depth cameras are getting increasingly common and, beyond retrieving depth information, we can enrich the main acquisition systems with additional hardware to measure geometric information about the sensor and improve data registration.

These evolutions allow practitioners to measure urban environments at resolutions that were until now possible only at the scale of individual shapes. The related scientific challenge is, however, more than just dealing with massive data sets coming from an increase in resolution, as complex scenes are composed of multiple objects, each object being itself seen as an association of shapes. Understanding the principles that govern the organization of urban environments requires the analysis of structural relationships between objects and shapes.

The geometric modeling of urban scenes has received significant attention over time. This area of research, and especially the 3D reconstruction from physical measurements, is a topic of intellectual and commercial inter-

est in many application domains. Computerized urban models are praised in urban planning for developing new plans in the context of an existing environment, but also in navigation, digital mapping, electro-magnetic wave propagation study for wireless networks, emergency management, disaster control, mission preparation for defense, entertainment industry, etc.

Researchers have concentrated their efforts at three different scales: remotely sensed, terrestrial and indoor. The former has been deeply explored for several decades, mainly driven by the emergence of remote sensing in the eighties. At this scale, a description of the main urban objects is expected, for instance, building roofs and road networks. Terrestrial and indoor scales have been more recently addressed, driven by the advances of sensors in terms of quality and mobility. Embedded in cars or reduced to a simple web-cam, these sensors have provided new data measurements allowing the analysis of streets, facades or building rooms, for example.

Three main challenges can be distinguished in the field: acquisition constraints, quality of models, and full automation.

Acquisition constraints. The acquisition process is usually a difficult task in the urban context. This produces defect-laden data. Noise is one of the typical defects. It can result from approximation in the data registration or directly from the sensor precision. Outliers are also frequent, especially with image stereo matching operations from textureless and reflective surfaces. Outliers also result from the presence of unwanted objects in scenes, for instance the temporary elements and the road signals when the modeling of buildings and facades is considered. Dealing with data which are heterogeneously sampled in the space is also a difficult problem. This arises, in particular, with laser sensors embedded into vehicles, the density of points decreasing according to the distance to objects. The most common and challenging defect remains the missing parts. Data can hardly cover entire complex environments because of the frequent occlusions. Geometric priors are typically exploited to explain such missing parts.

Quality of models. Depending on the application domain, different output properties can be expected. A result of *good* quality is not only a model with a high geometric accuracy, or faithfulness to the physical scene. A *good* computerized representation can be also defined by (i) the model complexity measuring the degree of compaction of the output representation, (ii) the structural guarantees imposing global regularities on the geometry and semantics of the output, and (iii) the visual aspect of the representation. The models must thus be measured and evaluated according to different criteria, usually in conflict between each other. Elaborating flexible metrics and strategies, which enable the combination of all these different criteria,

constitutes one of the main challenges in the field.

Full automation. One of the geometric modeling goals is to be as automatic as possible. Interactive modeling is usually recommended for architectural monuments and historical buildings, but remains ill-adapted to massive data for which considerable human resources would be required. Reaching full automation is an extremely difficult task as urban environments are complex and organized with a high degree of randomness, often resulting from an anarchical creation over time. Urban objects significantly differ in terms of diversity, complexity and density, even within a same scene. A predefined set of urban assumptions is rarely fully respected at the scale of a city. In practice, algorithms exploiting urban assumptions fails to model the entire scenes. To the contrary, algorithms omitting these assumptions are more flexible but the quality of models are usually lower. Faced with this dilemma, some scientists adopt an *automatic-then-interactive* strategy, the second step consisting in the interactive correction of the mistakes produced during the automatic step.

1.2 Approaches and trends

The scientific literature related to the geometric modeling of urban environments is large and lies across several communities such as computer vision, geometry processing, robotics and photogrammetry and remote sensing. The numerous approaches can be classified according to multiple criteria including acquisition specificities, characteristics of outputs, controllability, methodological foundations, and type of observed environments. Without being exhaustive, we present some existing works by listing several dualities. For a deeper review, the reader is invited to consult recent surveys [May08, VAM+09, HK10, MWA+12].

Airborne and terrestrial acquisitions. These constitute the two main types of acquisitions for urban modeling problems. Terrestrial systems are suited to capturing vertical components such as facades. Data usually contain many occlusions as a scene is seen as a set of urban object layers from the sensors. Airborne and satellite systems allow the description of landscapes at bigger scales, and particularly the non-vertical components such as roofs or ground. Methods that exploit such data often assume a 2.5D representation of the scene in the sense that only one layer of objects is present.

Image and Laser. Geometric modeling of urban environments mainly relies on two types of measurements: Multi-View Stereo (MVS) imagery and Laser. As mentioned in [LIP+10], notable differences exist between these two inputs. Imagery has usually a better accessibility and coverage than Laser. Nevertheless, 3D information cannot be straightforwardly obtained

from MVS images because camera calibration and image matching operations are required [Fau93, HZ04]. These active research fields in computer vision have led to numerous surveys and benchmarks such as [BBH03, SS02, SCD+06, SVHVG+08] to cite just a few of them. To the contrary, Laser acquisition directly generates points in the three-dimensional space with high accuracy. The problem of recovering shapes and surfaces is, however, similar in both worlds, 3D-points being the reference element. Indeed, the use of points in MVS imagery as an intermediate step between images and surfaces is now a commonly accepted idea in the vision community, outclassing the direct use of implicit surfaces [VKLP09, FP10, ASS+09, FFGG+10]. In many cases, images and Laser are combined to reinforce the modeling, *eg* [FZ03, MBH12]. In addition to MVS imagery and Laser, other types of acquisition are emerging, in particular, depth cameras such as Kinect enable the reconstruction of objects at short distances in real time [IKH+11], and video [PNF+08].

Offline and online modeling. The geometric modeling of scenes can be tackled either as a static problem, for which input data are not supposed to evolve in time (offline), or as a dynamical problem where a continuous flow of information makes the system update the output model (online). If offline modeling remains largely explored in Vision and Geometry Processing, online modeling constitutes one of the main challenges in the Robotics community, in particular, for navigation-based applications to solve SLAM (simultaneous localization and mapping). For such problems, geometric modeling is more difficult as the acquisition system needs to be localized at any time of the track so that new information can validate, modify or complete the current output model [DWB06].

Geometry and semantics. Geometry refers to questions of shape, size and relative position of objects in space. Semantics refers to the meaning and the nature of the objects composing a scene. Geometry and semantics are closely correlated in urban modeling. Semantics impacts on geometry in the sense that knowing the nature of an object allows us to adapt the modeling of the objects with specific geometric priors. For instance, piecewise-planar models are more suitable for modeling buildings that free-form surfaces which are more adapted for modeling trees. Geometry also impacts on semantics as the geometric relationships between objects can help to discover their nature. More and more works propose approaches combining geometry and semantics, such as [GP12, LGZ+13, HZC+13].

Detection and classification. Object detection and scene classification constitute two closely-related problems in urban analysis. The former consists in searching for one or several specific objects from input data, whereas

the latter aims to explain the entire scene by labeling the input data by classes of interest. Numerous works have been proposed for both problems. Detecting urban objects is usually addressed by locating and fitting predefined models, *eg* [May08], and by learning discriminative geometric attributes, *eg* [GKF09]. Scene classification has been deeply explored by graphical models [Bis06], in particular through random field theory. This problem is still an active field of research in computer vision with advances in hierarchical modeling [PWZ08] or multi-source procedures [MBH12] for instance.

Reconstruction and generation. Reconstruction and generation of cities represents two distinct problems. Reconstruction is the process of creating a model as close as possible to data measurements in terms of accuracy [MWA+12]. Generation consists in artificially creating realistic models given some predetermined rules and procedural mechanisms [VAM+09]. In recent years, these two distinct problems have tended to merge, in particular with recent works on inverse procedural modeling. This field of research constitutes one of the main challenges in city modeling.

Free-form and structure. Objects can be modeled by free-form representations, or by more specific representations exploiting geometric primitives, and beyond them, structural relationships. Free-form representations have been deeply explored in literature, in particular for smooth shapes from nature and designers. Urban scenes are mainly composed of man-made objects for which the notion of structure is important [MWZ+13]. Structure is a generic term, not necessarily well-defined, that refers i) to the way the individual shapes are grouped to form objects, object classes or hierarchies, ii) to geometry when dealing with similarity, regularity, parallelism or symmetry [MPWC12], and iii) to domain-specific semantic considerations. Discovering structural relationships is of interest for i) consolidating and reinforcing the data, in particular in presence of occlusions and corrupted measurements [LZS+11], ii) increasing the geometric regularity of output models [LWC+11], and iii) simplifying the modeling with a solution space reduction [CY00].

Local and global strategies. Contrary to local strategies, global strategies assume that entities composing a scene interact, even if they are spatially far away from each other. The choice between local and global is usually a trade-off between output quality and performance. Local strategies are computationally less complex (and thus faster), whereas global approaches lead to output solutions with more regularities. The Markovian assumption constitutes an interesting alternative between purely-local and global strategies as it restricts the dependency of entities in a certain neighborhood. This assumption has been deeply exploited in vision and image, in particular to

address labeling problems. In geometric modeling, interactions between entities are usually spatial, and correspond to geometric constraints that can be either hard (binary condition) or soft (continuous score). Global strategies are commonly used for primitive-based surface reconstruction, constrained meshing and surface approximation, *eg* [ZLAK14].

Solvers. Many problems can be formulated as the minimization of a function. Depending on the configuration space, the form of the function, and potential initial configurations, different types of optimization techniques can be considered to find or approximate the optimal solution. For instance, variational solvers are usually relevant choices when both function gradients can be computed (or estimated easily) and good initializations are available. Combinatorial solvers deal with discrete configuration spaces that can be embedded into graphs. Probabilistic tools as Monte Carlo samplers are of interest for optimizing highly non-convex functions. Globally speaking, the choice of the solver must be associated with the strategy.

2 Object extraction

We refer by object extraction, the search of parametric entities representing a specific class of interest from data measurements. It differs from data segmentation and classification in the sense that (i) objects describe a portion of the observed scene only, and (ii) objects are expected to have particular geometric shapes. Many approaches have been proposed in the literature to address this problem, as level sets, active contours, geodesic paths or elastic curves to cite just a few of them. We focus here on one particular approach based on stochastic geometry and called *point processes*.

Point processes are probabilistic models introduced by [BL93] to extend the traditional Markov Random Fields (MRF) with an object-based formalism. Indeed, Markov point processes can address object recognition problems by directly manipulating parametric entities in dynamic graphs, whereas MRFs are restricted to labeling problems in static graphs. These mathematical tools exploit random variables whose realizations are configurations of parametric objects, each object being assigned to a point positioned in the scene. The number of objects is itself a random variable, and thus must not be estimated or specified by a user. Another strength of Markov point processes is their ability to take into account complex spatial interactions between the objects and to impose global regularization constraints within a scene.

After briefly introducing these mathematical tools, we present a series of contributions on the modeling and the sampling of point processes as well as some applications to urban scene analysis.

2.1 Point processes

Definitions and notations. A point process describes random configurations of points in a continuous bounded set K. Mathematically speaking, a point process Z is a measurable mapping from a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ to the set of configurations of points in K such that

$$\forall \omega \in \Omega, p_i \in K, Z(\omega) = \{p_1, ..., p_{n(\omega)}\} \tag{1}$$

where $n(\omega)$ is the number of points associated with the event ω. We denote by $\mathcal{P}$, the space of configurations of points in K. Fig. 1 shows a realization of a point process for $K \subset \mathbb{R}^2$.

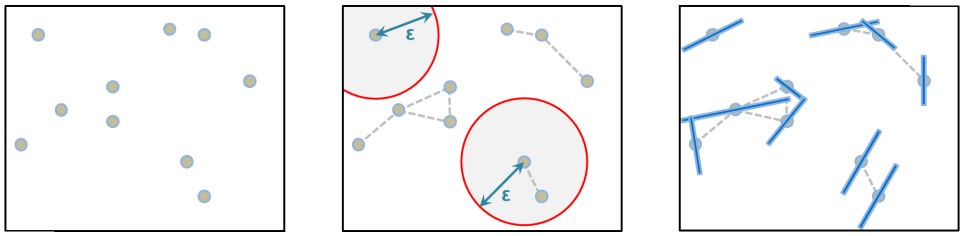


Figure 1: Point processes. From left to right: realizations of a point process in 2D, of a Markov point process, and of a Markov point process of line-segments. The dashed lines represent the pairs of points interacting with respect to the neighboring relationship which is specified here by a limit distance ϵ between two points (Eq. 6).

The most natural point process is the homogeneous Poisson process for which the number of points follows a discrete Poisson distribution whereas the position of the points is uniformly and independently distributed in K. Point processes can also provide more complex realizations of points by being specified by a density $h(.)$ defined in $\mathcal{P}$ and a reference measure $\mu(.)$ under the condition that the normalization constant of $h(.)$ is finite:

$$\int_{\boldsymbol{p} \in \mathcal{P}} h(\boldsymbol{p}) d\mu(\boldsymbol{p}) < \infty \tag{2}$$

The measure $\mu(.)$ having the density $h(.)$ is usually defined via the intensity measure $\nu(.)$ of an homogeneous Poisson process such that

$$\forall B \in \mathcal{B}(\mathcal{P}), \;\; \mu(B) = \int_B h(\boldsymbol{p}) \nu(d\boldsymbol{p}) \tag{3}$$

Specifying a density $h(.)$ allows the insertion of data consistency, and also the creation of spatial interactions between the points. Note also that $h(.)$ can be expressed by a Gibbs energy $U(.)$ such that

$$h(.) \propto \exp -U(.) \tag{4}$$

Markovian property. Similarly to random fields, the Markovian property can be used in a point process to create a spatial dependency of the points in a neighborhood.
A point process Z of density h is *Markovian under the neighborhood relationship* $\sim$ if and only if $\forall \mathbf{p} \in \mathcal{P}$ such that $h(\mathbf{p}) > \mathbf{0}$,

(i) $\forall \tilde{\boldsymbol{p}} \subseteq \boldsymbol{p}$, $h(\tilde{\boldsymbol{p}}) > 0$,

(ii) $\forall u \in K$, $h(\boldsymbol{p} \cup \{u\})/h(\boldsymbol{p})$ only depends on u and its neighbors $\{p \in \boldsymbol{p} : u \sim p\}$.

The expression $h(\boldsymbol{p} \cup \{u\})/h(\boldsymbol{p})$ can be interpreted as a conditional intensity. The Markovian property for random fields can thus be naturally extended in case of point processes by defining a symmetric relationship between two points of K. As shown later, the Markovian property is essential to facilitate the sampling of point processes.

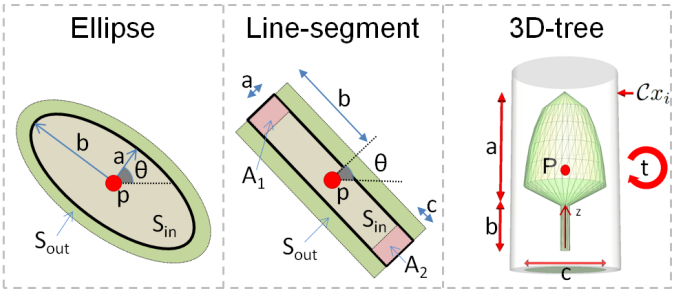


Figure 2: Examples of geometric objects used in point processes. (left) Ellipses and (middle) line-segments are defined by a 2D-point $p \in K$ (center of mass of the object) and some marks, *eg* the semi-major axis b, the semi-minor axis a, and the angle θ for an ellipse. (right) 3D-trees are defined by a 3D-point $p \in K$ (center of mass of the object), a type $t \in \{$conoidal, ellipsoidal, semi-ellipsoidal$\}$, and 3 additional parameters which are the canopy height a, the trunk height b and the canopy diameter c.

From points to parametric objects. Each point p_i can be marked by additional parameters m_i such that the point becomes associated with an object $x_i = (p_i, m_i)$. This property is particularly attractive to address vision problems requiring the handle of complex parametric objects (see Fig. 2). We denote by $\mathcal{C}$, the corresponding space of object configurations where each configuration is given by $\boldsymbol{x} = \{x_1, ..., x_{n(\boldsymbol{x})}\}$. For example, a point process on $K \times M$ with $K \subset \mathbb{R}^2$ and the additional parameter space $M =]-\frac{\pi}{2}, \frac{\pi}{2}] \times [l_{min}, l_{max}]$ can be seen as random configurations of 2D line-segments since an orientation and a length are added to each point (see

Fig. 1). Such point processes are also called marked point processes in the literature.

The most popular family of point processes corresponds to the Markov point processes of objects specified by Gibbs energies on $\mathcal{C}$ of the form

$$\forall \boldsymbol{x} \in \mathcal{C}, \quad U(\boldsymbol{x}) = \sum_{x_i \in \boldsymbol{x}} D(x_i) + \sum_{x_i \sim x_j} V(x_i, x_j) \tag{5}$$

where $\sim$ denotes a symmetric neighborhood relationship, $D(x_i)$ is a unitary data term measuring the quality of object x_i with respect to data, and $V(x_i, x_j)$, a pairwise interaction term between two neighboring objects x_i and x_j. The $\sim$relationship is usually defined via a limit distance ϵ between points such that

$$x_i \sim x_j = \{(x_i, x_j) \in \boldsymbol{x}^2 : i > j, ||p_i - p_j||_2 < \epsilon\} \tag{6}$$

In the sequel, we consider Markov point processes of this form. Note that this energy form has similarities with the standard multi-label energies for MRFs [SZS+08]. Our problem can indeed be seen as a generalization of these MRF models where **(i)** the dimension of the configuration space $\mathcal{C}$ is variable (graph structure is not static but dynamic), **(ii)** labels are defined in discrete or/and continuous domains so that complex parametric objects can be handled, **(iii)** there are no constraints on the form of the pairwise interaction term V imposed by the minimization techniques.

The unitary data term D is commonly formulated by a Bhattacharyya distance when inputs are images. It consists in comparing intensity inside and outside the object using statistical operators, *ie* mean and standard deviation. Details can be found in [Des11, VL14]. More complex measures can also be used, in particular with tree-dimensional spaces requiring photo-consistency considerations [UB11] or point density analysis [SMS07]. Pairwise interaction V is often a basic repulsion term which avoids spatial overlaps of objects. No restriction is imposed on the form of V. As discussed latter in Section 2.2, this advantage is also a drawback as complex interactions lead to slower convergences and also the presence of numerous model parameters.

Simulation. Point processes are usually simulated using a RJMCMC sampler [Gre95] to search for the configuration which minimizes the energy U. This sampler consists of simulating a discrete Markov Chain $(X_t)_{t \in \mathbb{N}}$ on the configuration space $\mathcal{C}$, converging towards a target density specified by U. At each iteration, the current configuration $\boldsymbol{x}$ of the chain is locally perturbed to a configuration $\boldsymbol{y}$ according to a density function $Q(\boldsymbol{x} \to .)$, called a proposition kernel. The perturbations are local, which means that $\boldsymbol{x}$ and $\boldsymbol{y}$ are very close, and differ by no more than one object in practice. The

configuration $\boldsymbol{y}$ is then accepted as the new state of the chain with a certain probability depending on the energy variation between $\boldsymbol{x}$ and $\boldsymbol{y}$, and a relaxation parameter T_t. The kernel Q can be formulated as a mixture of sub-kernels Q_m chosen with a probability q_m such that

$$Q(\boldsymbol{x} \to .) = \sum_m q_m Q_m(\boldsymbol{x} \to .) \tag{7}$$

Each sub-kernel is usually dedicated to specific types of moves, as the creation/removal of an object (Birth and Death kernel) or the modification of parameters of an object (*eg* translation, dilatation or rotation kernels). The kernel mixture must allow any configuration in $\mathcal{C}$ to be reached from any other configuration in a finite number of perturbations (irreducibility condition of the Markov chain), and each sub-kernel has to be reversible, *ie* able to propose the inverse perturbation. Note that the Birth and Death kernel, whose formulation is based on the updating mechanism proposed in [GM94a], is necessary to simulate point processes. Details on kernel formulation can be found in [Des11].

Algorithm 1 RJMCMC sampler for point processes

1- Initialize $X_0 = \boldsymbol{x}_0$ and T_0 at $t = 0$;
2- At iteration t, with $X_t = \boldsymbol{x}$,

- Choose a sub-kernel $\mathcal{Q}_m$ according to probability q_m
- Perturb $\boldsymbol{x}$ to $\boldsymbol{y}$ according to $Q_m(\boldsymbol{x} \to .)$
- Compute the Green ratio

$$R = \frac{\mathcal{Q}_m(\boldsymbol{y} \to \boldsymbol{x})}{\mathcal{Q}_m(\boldsymbol{x} \to \boldsymbol{y})} \exp\left(\frac{U(\boldsymbol{x}) - U(\boldsymbol{y})}{T_t}\right) \tag{8}$$

- Choose $X_{t+1} = \boldsymbol{y}$ with probability $\min(1, R)$, and $X_{t+1} = \boldsymbol{x}$ otherwise

The RJMCMC sampler is controlled by the relaxation parameter T_t, called the temperature, depending on time t and approaching zero as t tends to infinity. Although a logarithmic decrease of T_t is necessary to ensure the convergence to the global minimum from any initial configuration, one uses a faster geometric decrease of the form

$$T_t = T_o.\alpha^t \tag{9}$$

where α and T_0 are, respectively, the decrease coefficient and the initial temperature. Such a geometric decrease gives an approximate solution close to the optimum. The decrease coefficient α is typically chosen as constant value both inferior and close to 1. T_0 is estimated through the variation of

the energy U on random configurations. More precisely, T_0 is usually chosen as twice the standard deviation of U at infinite temperature:

$$T_0 = 2.\sigma(U_{T=\infty}) = 2.\sqrt{\langle U^2_{T=\infty}\rangle - \langle U_{T=\infty}\rangle^2} \tag{10}$$

where $\langle U\rangle$ is the means of the energy of the samples. More details on temperature decrease schemes can be found in [SSF02].

2.2 Objects and interactions

Point processes proposed in the image and vision literature have been designed with some technical limitations. We present here several solutions for improving point processes in terms of flexibility and applicability.

Library of geometric shapes. The conventional point processes are restricted to the use of a single type of objects, the dimension of the mark space M being fixed. This drawback is particularly penalizing in urban context where many various shapes exit. For instance, rectangles are considered in [BDZ12, TPL07], line-segments in[LDZ05, SMS07], ellipses in [DMZ09], or cylinders in [UB11]. Ortner *et al.* [ODZ08] proposed to overcome this drawback by considering two marked point processes each using a different type of objects (rectangles and segments). The two processes are sampled jointly by a Markov Chain Monte Carlo algorithm. However, in this approach, both energy formulations and simulated annealing tunings become too complex to manage because cooperative interactions between both processes must be taken into account. This model cannot be adapted in practice to deal with a large number of object types.

We propose in [LGD10] to generalize the conventional marked point process framework in order to jointly sample various types of geometric objects. To do so, we consider a finite library of marks allowing the definition of multiple type of geometric objects. The mark space M associated with this library is then specified as a finite union of mark bounded subsets M_q:

$$M = \bigcup_{q=1}^{N_s} M_q \tag{11}$$

where each subset M_q corresponds to one of the N_s specific shape types. In other words, the associated marked point process is able to deal with objects having different numbers of control parameters, as illustrated on Fig. 3. Such a process, called by extension a *multi-marked point process*, implies two important changes with respect to conventional point processes.

- *Energy restriction.* Energy must deal with multiple types of geometric objects. In particular, the data term measuring the object fitting

quality to the data must give consistent and homogeneous values in spite of a diversity of object types, *eg* lineic/surface/volume or rectilinear/curved. The spatial interactions between objects must also be formulated so that interactions between specific classes of objects are not favored with respect to other ones. The energy proposed in [LGD10] provides a general formulation for multiple types of geometric objects.

- *Switching kernel.* For efficiently simulating such point processes, an additional kernel must be introduced in the RJMCMC algorithm so that moves modifying the type of an object can be proposed. Contrary to the Birth and Death kernel, these moves do not change the number of objects in the configuration, but they change the number of parameters. For instance, moving from an ellipse to a rectangle generates two additional parameters. This kernel creates bijections between the different types of objects based on the idea proposed by [Gre95]. Details on the switching kernel computation are given in [LGD10].

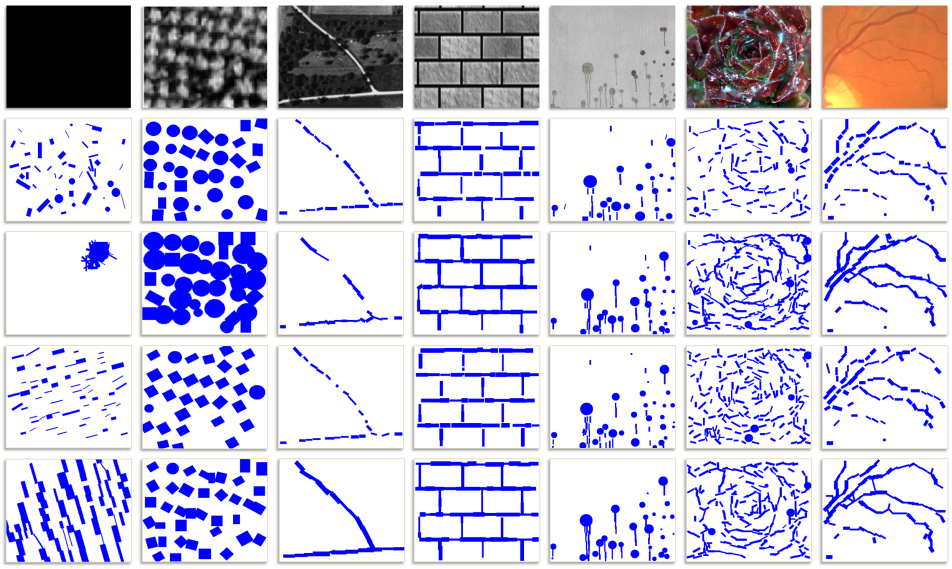


Figure 3: Point processes of multiple geometric objects using various spatial interactions. *(from top to down)* Input images, results by using a non-overlapping interaction, a connection interaction, an alignment interaction, and a combined connection/alignment interaction. Generally speaking, complex interactions improve the result quality, but generate extra running times and extra model parameters.

Junction-points. Point processes allow the manipulation of geometric objects, each point being marked by a set of parameters to create an object. Such a correspondence between point and object might not be adapted to some problems in which complex structures must be recovered. This is the case in particular for line-network extraction problems for which planar graphs constitute much better outputs than line-segments. In the literature, most of point processes [BDZ12, TPL07, LDZ05, SMS07, ODZ08] relies on the formulation of complex spatial interactions in order to favor object configurations with particular geometric structures. This choice is arguable as it leads to (i) minimize highly non-convex energies, and (ii) define many model parameters to tune or to estimate. In practice, the simulation of these point processes is slower, and the amount of model parameters makes the processes unstable and non flexible.

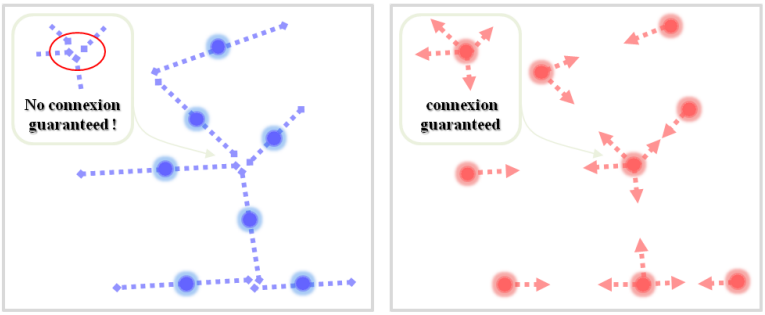


Figure 4: Junction-point process. A marked point process of line-segments (left) cannot describe a line-network with accuracy as the line-segments are not ideally connected. To the contrary, a junction-point process (right) brings structural guarantees as a unique planar graph is associated to each junction-point configuration.

We propose in [CFL13] a new family of point processes able to manipulate planar graphs, called *junction-point processes*. The main idea relies on the fact that each point of a realization informs the directions where its adjacent points are located. A junction-point process on F is a point process on K for which every point $p_i \in K$ is completed by a set of directions $(\gamma_i^{(1)}, ..., \gamma_i^{(k)})$, and optionally a set of additional parameters $(w_i^{(1)}, ..., w_i^{(k)})$, so that a point configuration is associated to a unique planar graph in F. We define a k-junction-point x_i by

$$x_i = (p_i, \gamma_i^{(1)}, .., \gamma_i^{(k)}, w_i^{(1)}, .., w_i^{(k)}) \tag{12}$$

where k represents the number of directions. Each junction-point configuration $x = \{x_1, ..., x_n\}$ is associated to a planar graph G_x. The additional

parameters $(w_i^{(1)}, ..., w_i^{(k)})$ of the junction-point x_i correspond to user-defined attributes on adjacent edges of the node i in the graph G_x. Contrary to the conventional marked point processes used in the literature, junction-point processes do not require complex geometric priors as a graph structure is directly guaranteed by construction. In addition, the sampling procedure is thus more stable and the model parameters are highly reduced. Fig. 4 illustrates the advantages of junction-point processes with respect to a conventional point process of line-segments for addressing line-network extraction problems.

2.3 Sampling

The results obtained by conventional point processes are convincing and competitive with respect to other families of methods, but the performances are particularly limited in terms of running time and convergence stability, especially on large scenes. Point processes usually rely on standard sampling procedures, mainly on the Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm [Gre95]. The running time generated by such a sampler is reasonable only from data of small size. For example, the building extraction algorithm proposed by [ODZ08] requires around six hours from an image portion of size 1000×1000 pixels only (0.25 km^2 area) using a 2GHz computer. Such a solution is obviously not reasonable when dealing with large-scale aerial and satellite images.

In the literature, few works have addressed the optimization issues from large-scale data. The proposed solutions are mainly based on some improvements of the traditional RJMCMC sampler. In particular, data considerations can be used to drive the MCMC sampling, renamed Data-Driven MCMC, with more efficiency [TZ02]. The idea consists in modeling the proposition kernels of the sampler in function of discriminative tests from data so that the ratio of relevant perturbations is strongly increased. This strategy can be dangerous if the proposition kernels are not correctly estimated from data. Some works have also proposed parallelization procedures by using multiple chains simultaneously [HG00] or decomposition schemes in configuration spaces of fixed dimension [BJB10, GLGG11]. However they are not designed to perform on large scenes, and cannot be used for configuration spaces of variable dimension. Parallel tempering [ED05] runs multiple chains in parallel at different temperatures while frequently exchanging configurations during the sampling. This technique brings robustness to the cooling schedule, but remains slow in practice as each chain explores the whole configuration space. A mechanism based on multiple creation and destruction of objects has also been developed to address population counting problems [DMZ09]. Nevertheless this algorithm is semi-deterministic and can only address problems in which object interactions are simple. In addition, object creations require the discretization of the point coordinates which induces a

significant loss of accuracy.

These alternative versions of the conventional MCMC sampler globally allow the improvement of optimization performances in specific contexts. That said, the gain in terms of time remains weak and is usually realized at the expense of convergence stability, especially in large scenes. We present here two solutions for improving sampling performances, one based on the integration of diffusion dynamics within the MCMC mechanism, and the other relying on a parallelization procedure exploiting the Markovian property of point processes.

Jump-Diffusion. One of the main drawbacks of the conventional RJMCMC samplers is the difficulty to make the point processes converge at low temperature. The local adjustment of objects is usually a long and fastidious step as many irrelevant propositions are rejected. We present a technique for accelerating the local adjustment of objects by exploiting the energy gradient. Relying on the original idea of Grenander and Miller [GM94b], we propose in [LGD10] a Jump-Diffusion mechanism consisting in inserting a diffusion dynamics within the conventional RJMCMC sampler. Such a mechanism has been successfully adapted to various computer vision problems such as pose estimation [SGJM02] and image segmentation [HTZ04].

Jump-Diffusion combines the conventional Markov Chain Monte Carlo algorithms [Gre95] and the Langevin equations [GH86]. Both dynamics play different roles in the Jump-Diffusion process: the former performs reversible jumps between the different subspaces of $\mathcal{C}$, whereas the latter conducts stochastic diffusion within each continuous subspace. The conventional RJMCMC algorithms use perturbation kernels that allows the exploration of each subspace by only modifying parameters of the objects. Here, this kernel is substituted by a diffusion dynamic which provides a faster exploration of the subspace using energy gradient considerations. The global process is controlled by a unique relaxation temperature T. The diffusions are interrupted by jumps after a constant time interval. Note that some Jump-Diffusion adaptations in the literature prefers interrupt diffusions according to a Poisson distribution [HTZ04].

The diffusion process controls the evolution of the object configuration in their respective subspaces. Stochastic diffusion equations are driven by Brownian motions depending on the relaxation temperature T. The diffusions are used to explore each subspace of $\mathcal{C}$. If $x(t)$ denotes the variables at time t, then

$$dx(t) = -\frac{dU(x)}{dx}dt + \sqrt{2T(t)}dw_t \tag{13}$$

where $dw_t \sim N(0, dt^2)$. At high temperature ($T >> 0$), the Brownian motion is necessary to guarantee the convergence of the general process. In particular, it avoids the process to get stuck in local minima. At low

temperature ($T << 1$), the role of the Brownian motion becomes negligible and the diffusion dynamics acts as a gradient descent. This mechanism is however restricted to specific energy forms as the energy gradient must be computable, or at least estimable. Details concerning the valid forms of energy can be found in [LGD10].

Parallelization. The conventional RJMCMC sampler performs successive perturbations on objects. Such a procedure is obviously long and fastidious, especially for large scale problems. A natural idea consists in sampling objects in parallel by exploiting their conditional independence outside the Markovian spatial neighborhood. Such a strategy implies partitioning the space K so that simultaneous perturbations are performed at locations far enough apart to not interfere and break the convergence properties.

Let $(X_t)_{t\in\mathbb{N}}$, be a Markov chain simulating a Markov point process with a MCMC dynamics, and $\{c_s\}$ be a partition of the space K, where each component c_s is called a cell. Two cells c_1 and c_2 are said to be *independent on* X if the transition probability for any random perturbation falling in c_1 at any time t does not depend on the objects and perturbations falling in c_2, and vice versa. One can demonstrate that the transition probability of two successive perturbations falling in independent cells under the temperature T_t is equal to the product of the transition probabilities of each perturbation under the same temperature [VL14]. In other words, realizing two successive perturbations on independent cells at the same temperature is equivalent to performing them in parallel.

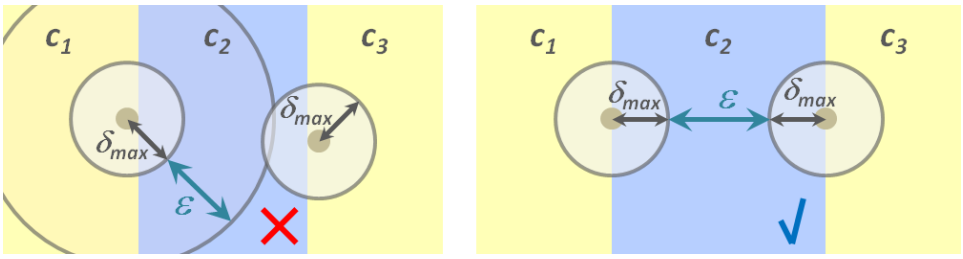


Figure 5: Independence of cells. On the left case, the width of the cell c_2 is not large enough to ensure the independence of the cells c_1 and c_3: the two grey points in c_1 and c_3 cannot be perturbed at the same time. On the right case, the cells c_1 and c_3 are independent as Eq. 14 is satisfied.

In practice, this implies that the two cells must be located at a minimum distance from each other. As illustrated in Fig. 5, this distance must take into account the width ϵ of the neighboring relationship induced by the

Markovian property so that every possible object falling in the first cell cannot be a neighbor of the objects falling in the second cell. As an object can be displaced to another cell during a perturbation, the minimum distance must also consider the length of the biggest move allowed, denoted by $\delta_{\max}$. Considering these two constraints, the independence between two cells c_1 and c_2 is then valid when

$$\min_{p_1 \in c_1,\, p_2 \in c_2} ||p_1 - p_2||_2 \geq \epsilon + 2\delta_{\max} \tag{14}$$

Given this cell independence condition (Eq. 14), the objective is now to find a good partition of the space K. The natural idea consists of partitioning K into a regular mosaic of cells with size greater than or equal to the minimum distance between independent cells, *ie* to $\epsilon + 2\delta_{\max}$. The cells can then be regrouped into $2^{\dim K}$ sets such that each cell is adjacent to cells belonging to different sets. In the sequel, such a set is called a *mic-set* (set of Mutually Independent Cells). Each cell of a mic-set can thus be perturbed simultaneously using a MCMC dynamics. Sampling objects in parallel via a regular partitioning is however not optimal because the spatial point distribution is uniform and does not take into account the characteristics of observed scenes. To overcome this problem, we create a non-regular partitioning of the scene using a Data-driven mechanism.

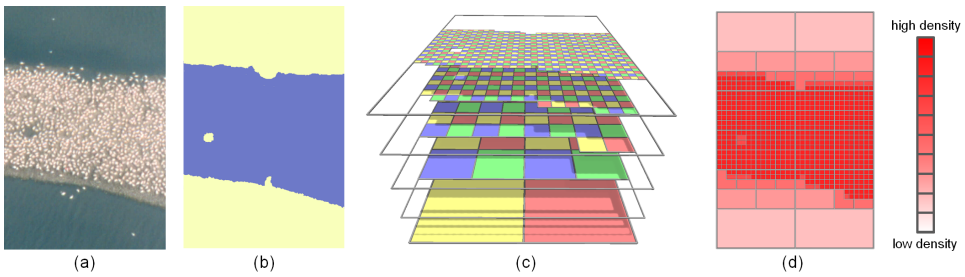


Figure 6: Space-partitioning tree in dimension two. (b) A class of interest (blue area) is estimated from (a) an input image. (c) A quadtree is created so that the levels are recursively partitioned according to the class of interest. Each level is composed of four mic-sets (yellow, blue, red and green sets of cells) to guarantee the sampling parallelization. (d) The accumulation of the probabilities $q_{c,t}$ over the different levels of the quadtree generates a density map allowing the points to be non-uniformly distributed in the scene. Note how the density map focuses on the class of interest while progressively decreasing its intensity when moving away.

Our idea consists in creating a proposition kernel as an accumulation of uniform sub-kernels spatially restricted on the domain of the cells. The

sub-kernel accumulation mechanism is driven by a space-partitioning tree $\mathcal{K}$ which is defined as a set of L sub-partitions of K, denoted by $\{c_s\}^{(1)}, .., \{c_s\}^{(L)}$, and organized so that, for $i = 2..L$, $\{c_s\}^{(i)}$ is a subdivided partition of $\{c_s\}^{(i-1)}$. Each level of the space-partitioning tree corresponds to a set of cells having an identical size. A 1-to-$2^{\dim K}$ hierarchical subdivision scheme is considered to build the space-partitioning tree, typically a quadtree in dimension two and an octree in dimension three. Given a space-partitioning tree, a density map specifying how the points must be spatially distributed in the space K can then be constructed. The creation of this density map relies on the accumulation of the uniform sub-kernels spatially restricted to the subspace supporting every cell of the space-partitioning tree $\mathcal{K}$, as defined in Eq. 7 and illustrated in Fig. 6. In order to create a relevant space-partitioning tree, the data are used to guide the cell subdivision. We assume that a class of interest in K, in which the objects have a high probability to belong to, can be roughly distinguished from the data. A cell at a given level of the tree is divided into $2^{\dim K}$ cells at the next level if it overlaps with the given class of interest. The hierarchical decomposition is stopped before that the size of the cell becomes inferior to $\epsilon + 2\delta_{\max}$, *ie* before that the cell independence condition (Eq. 14) is not longer valid.

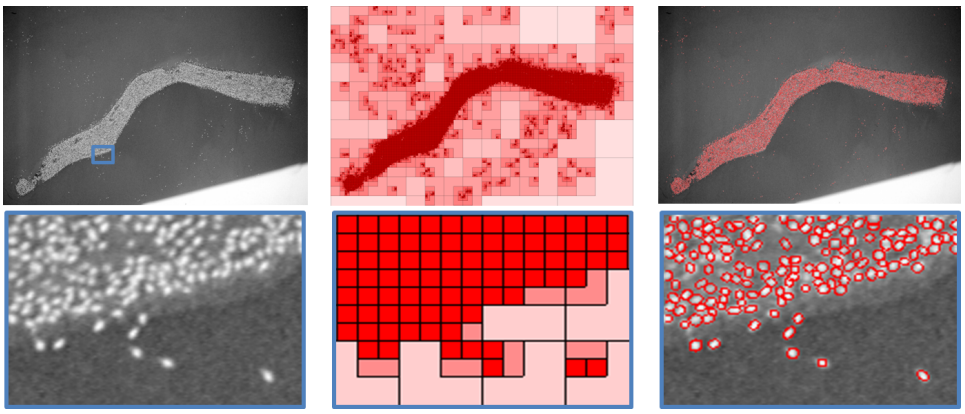


Figure 7: Bird counting by a point process of ellipses [VL14]. (right) More than ten thousand birds are extracted in a few minutes from (left) a large scale aerial image. (middle) A quadtree partitioning the scene is used to create a density map so that the objects are more frequently proposed in the locations of interest. Note, on the cropped parts, how the birds are accurately captured by ellipses in spite of the low quality of the image and the partial overlapping of birds.

Given a space-partitioning tree $\mathcal{K}$ composed of L levels, and $2^{\dim K}$ mic-sets for each level, a general proposition kernel Q can then be formulated as

a mixture of uniform sub-kernels $Q_{c,t}$, each sub-kernel being defined on the cell c of $\mathcal{K}$ by the perturbation type $t \in \mathcal{T}$, such that

$$\forall \boldsymbol{x} \in \mathcal{C},\ Q(\boldsymbol{x} \rightarrow .) = \sum_{c \in \mathcal{K}} \sum_{t \in \mathcal{T}} q_{c,t}\, Q_{c,t}(\boldsymbol{x} \rightarrow .) \tag{15}$$

where $q_{c,t} > 0$ is the probability of choosing the sub-kernel $Q_{c,t}(\boldsymbol{x} \rightarrow .)$. The probability $q_{c,t}$ allows us to specify the intensity of the density map, given the space-partitioning tree. In practice, this probability is chosen as

$$q_{c,t} = \frac{\Pr(t)}{\#\text{cells in } \mathcal{K}} \tag{16}$$

where $\Pr(t)$ denotes the probability of choosing the perturbation type $t \in \mathcal{T}$. The expression of $q_{c,t}$ (Eq. 16) allows the finest levels in the space-partitioning tree to be favored so that the perturbations mainly focus on the domain supporting the class of interest and its surrounding.

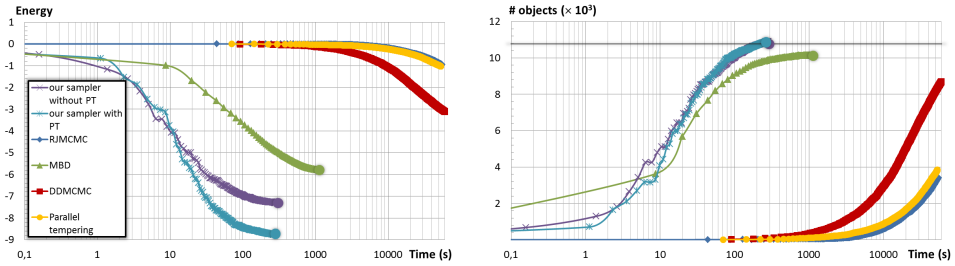


Figure 8: Performances of various samplers. The left graph describes the energy decrease over time from the bird image presented in Fig. 7 (the colored dots correspond to algorithm convergence). Note that time is represented using a logarithmic scale, and that the slow convergence of RJMCMC [Gre95], DDMCMC [TZ02], and parallel tempering [ED05] algorithms is not displayed on the graph. The right graph shows the evolution of the number of objects during the sampling. Contrary to the other samplers, the number of objects found by our sampler with and without space-partitioning tree (PT) is very close to the ground truth (black line).

This kernel formulation is embedded into a MCMC dynamics so that the proposed sampler allows a high number of simultaneous perturbations generated by a data-driven proposition kernel. The proposed sampler, detailed in Algorithm 2, can be seen as a parallelized extension of the traditional RJMCMC with data-driven proposition kernel.
Note that the temperature parameter is updated after each series of simultaneous perturbations such that the temperature decrease is equivalent to a

Algorithm 2 Data-driven parallel sampler

1-Initialize $X_0 = \boldsymbol{x}_0$ and T_0 at $t = 0$;
2-Compute a space-partitioning tree $\mathcal{K}$;
3-At iteration t, with $X_t = \boldsymbol{x}$,

- Choose a mic-set $S_{mic} \in \mathcal{K}$ and a kernel type $t \in \mathcal{T}$ according to probability $\sum_{c \in S_{mic}} q_{c,t}$
- For each cell $c \in S_{mic}$,
  - Perturb $\boldsymbol{x}$ in the cell c to a configuration $\boldsymbol{y}$ according to $Q_{c,t}(\boldsymbol{x} \to .)$
  - Calculate the Green ratio

$$R = \frac{\mathcal{Q}_{c,t}(\boldsymbol{y} \to \boldsymbol{x})}{\mathcal{Q}_{c,t}(\boldsymbol{x} \to \boldsymbol{y})} \exp\left(\frac{U(\boldsymbol{x}) - U(\boldsymbol{y})}{T_t}\right) \tag{17}$$

  - Choose $X_{t+1} = \boldsymbol{y}$ with probability $\min(1, R)$, and $X_{t+1} = \boldsymbol{x}$ otherwise

cooling schedule by plateau in a standard sequential MCMC sampling. In practice, the sampling is stopped when no perturbation has been accepted during a certain number of iterations. The mechanism of this sampler is shown on Fig. 7 from a point process of ellipses, as well as performances with respect to other samplers on Fig. 8.

2.4 Some applications

Point processes are flexible tools that can be used in different application domains in vision. We present here some results from concrete problems. Note that more qualitative and quantitative results, as well as comparative evaluations can be found in [LDZPD08, LGD10, VL14, CFL13] and from an online benchmark (http://www-sop.inria.fr/members/Florent.Lafarge/benchmark/evaluation.html).

Population counting. Population counting is conventionally addressed with probabilistic models by estimating the population density from data, *eg* [LZ10]. Point processes can be used for such a problem by detecting a specific class of objects. In addition to the number of objects, point processes allow us to discover statistical information related to the position, the size and the spatial organization of objects.

In the experiments shown on Fig. 9 and 10, a conventional point process marked by ellipses is considered. Ellipses are simple geometric objects

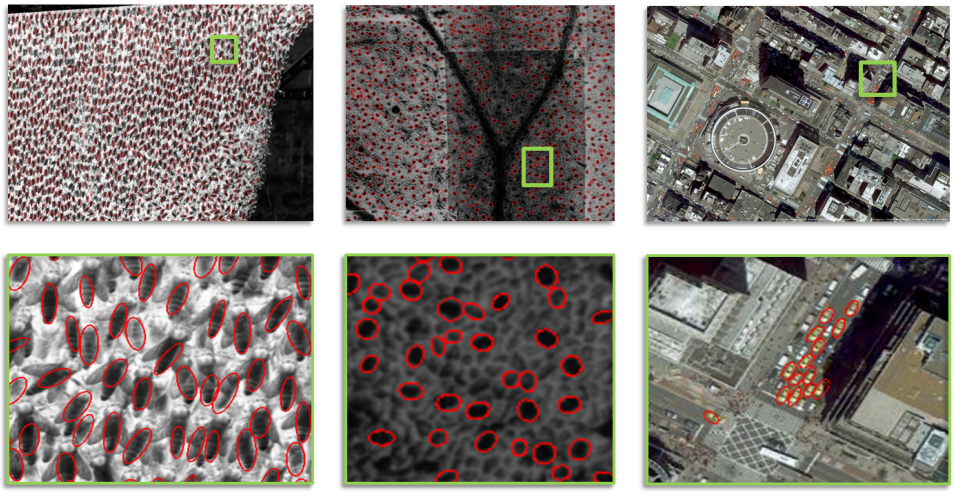


Figure 9: Various population counting problems [VL14]. The point process captures different objects of interest by ellipses in large scenes, as (left) bees from beehive pictures, (middle) opened stomata from microscope images of leaf, and (right) yellow cabs from aerial images. 1167 bees (respectively 757 stomata and 87 taxis) are detected in 12 minutes (respectively 168 seconds and 165 seconds). Note that the running time is higher for bee detection because the partitioning scheme of the data-driven parallel sampler contains few cells, *ie* 75. As shown on the close-ups, the objects are globally well detected in spite of the high concentration and overlap of objects.

defined by a point (center of mass of an ellipse) and three additional parameters. This object shape is general enough to capture numerous entities as cells from microscope images, cars from aerial images, or bees from photos. The energy is specified by a unitary data term based on the Bhattacharyya distance between the radiometry inside and outside the object, and a pairwise interaction penalizing the strong overlapping of objects. Details on the energy formulation are given in[VL14]. The data-driven parallel sampler presented in Alg. 2 is used to simulate the point process. As illustrated on Fig. 10, this sampler is more efficient than the other existing solutions in terms of both time and energy reached at convergence. Such a model also compete well with specialized counting methods as [LZ10].

Structure extraction. Point processes can be used for recovering specific structures, as for instance line-networks from images (Fig. 11 and 13) or building footprints from Digital Elevation Models (Fig. 12). As discussed previously, two strategies can be exploited for introducing structural consid-

erations in point processes.

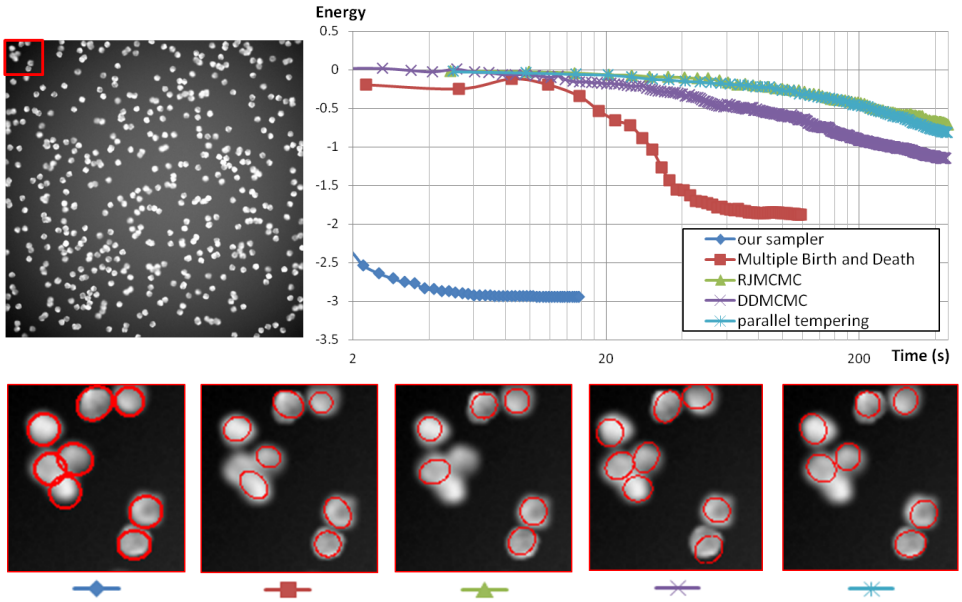


Figure 10: Performances of various samplers on cell counting [VL14]. The top right graph presents the performances of the existing algorithms in terms of time and energy from (top left) a microscope image, whereas the bottom close-ups show the quality of the reached configurations. The data-driven parallel sampler presented in Alg. 2 allows both low running time and good configuration quality.

The first option consists in associating simple geometric objects using complex spatial interactions as connection, alignment or paving. In [VL14, LDZPD06, LDZPD08], typical objects are line-segments for capturing road networks and rectangles for building footprints.The pairwise potential is more complex than for population counting as it includes geometric interactions for favoring predefined structures. For road networks for instance, interactions for connecting extremities of line-segments are considered. This option brings flexibility but usually requires additional optimization efforts. Indeed, the design of complex geometric interactions make the energy highly non-convex.

The second option consists in encoding the structure directly through the geometric objects. As shown in Fig. 13, junction-points forming a planar graph can be used to extract line-networks. The resulting configurations are guaranteed to be structurally valid by construction.

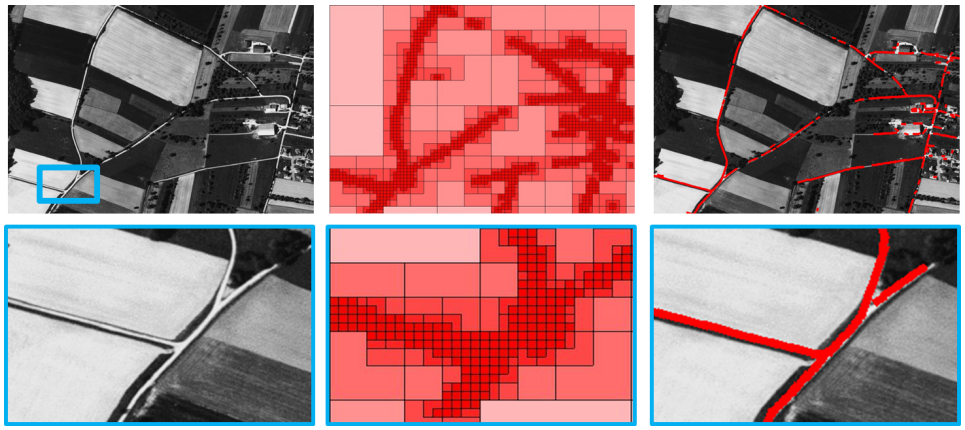


Figure 11: Road-network extraction by point process of line-segments [VL14]. (middle) Using a rough density map, (right) the road network is recovered (red segments) by point process of line-segments in 16 seconds from (left) a satellite image. Similarly to existing object-based methods, some parts of the network can be omitted when roads are hidden by trees at some locations, as shown on the close-up.

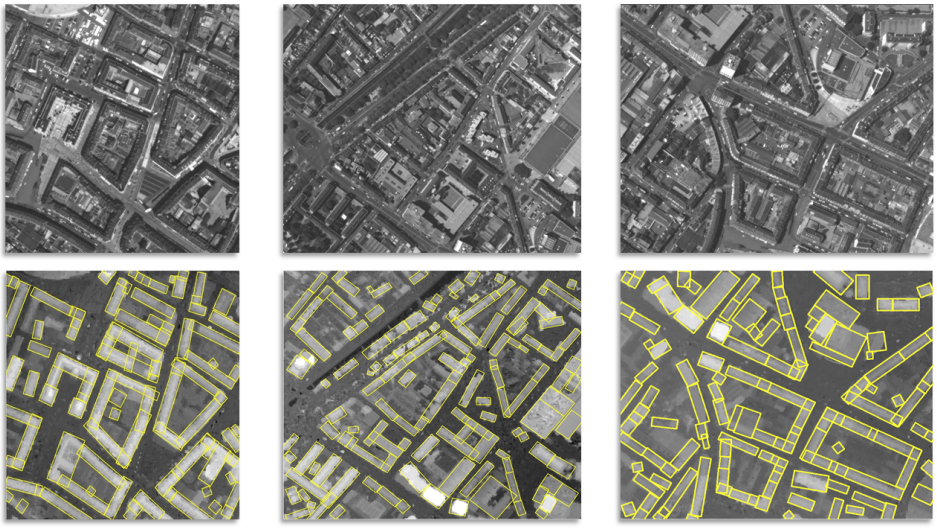


Figure 12: Building footprint extraction by point process of rectangles [LDZPD08]. (top) Buildings contained in urban scenes have a high pixel intensity from (bottom) Digital Elevation Models. They are captured as a set of connected rectangles located in the bright areas from Digital Elevation Models.

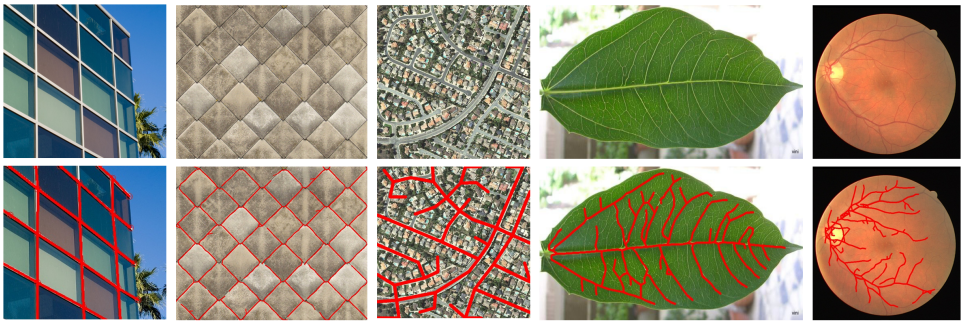


Figure 13: Line-network detection by Junction-point process [CFL13]. The algorithm is able to extract both regular (first and second columns, facade and tiles) and free-form (three left columns, roads in a residential area, leaf and blood vessels in a retinal image) line-networks. Note in particular that line-networks with different widths are recovered with few omissions, *eg* blood vessels or leaf.

Texture analysis. Many methods analyze textures as regular structures that are endlessly repeated. In this context, point processes can be of interest, in particular to characterize a texture with a structural signature resulting from layouts of simple geometric objects. Fig. 14 illustrates the potential of point processes for sketching natural textures as configurations of simple geometric shapes. The obtained results describe well the main structural patterns and reveal interesting fine details on a large range of textures, both spatially homogeneous and heterogeneous. A more advanced data term would allow to improve the results on reflective textures, *eg* the metal grid and tile roof. For a deeper use of point processes, one would propose texture classification using first and second order statistics on object configurations.

3D shape recognition. The previous experiments have been led with two dimensional point processes from images, *ie* with $\dim K = 2$. This can also be used to address 3D problem, in particular the recognition of 3D shapes from unstructured point clouds. Contrary to supervised methods, *eg* [GKF09], we assume the shapes of interest can be modeled by predefined parametric 3D-templates. In [VL14], we formulated a 3D point process for detecting trees from Laser scans of urban environments composed of many other different objects such as buildings, ground, cars, fences, wires, *etc.* This model also allows the recognition of the shapes and types of trees. The objects associated with the point process correspond to a library of different 3D-templates of trees. The unitary data term of the energy measures the distance from points to the surface of the 3D-object, whereas the pairwise interaction takes into account constraints on object overlapping as well as on tree type competition. Compared to the former applications, the config-

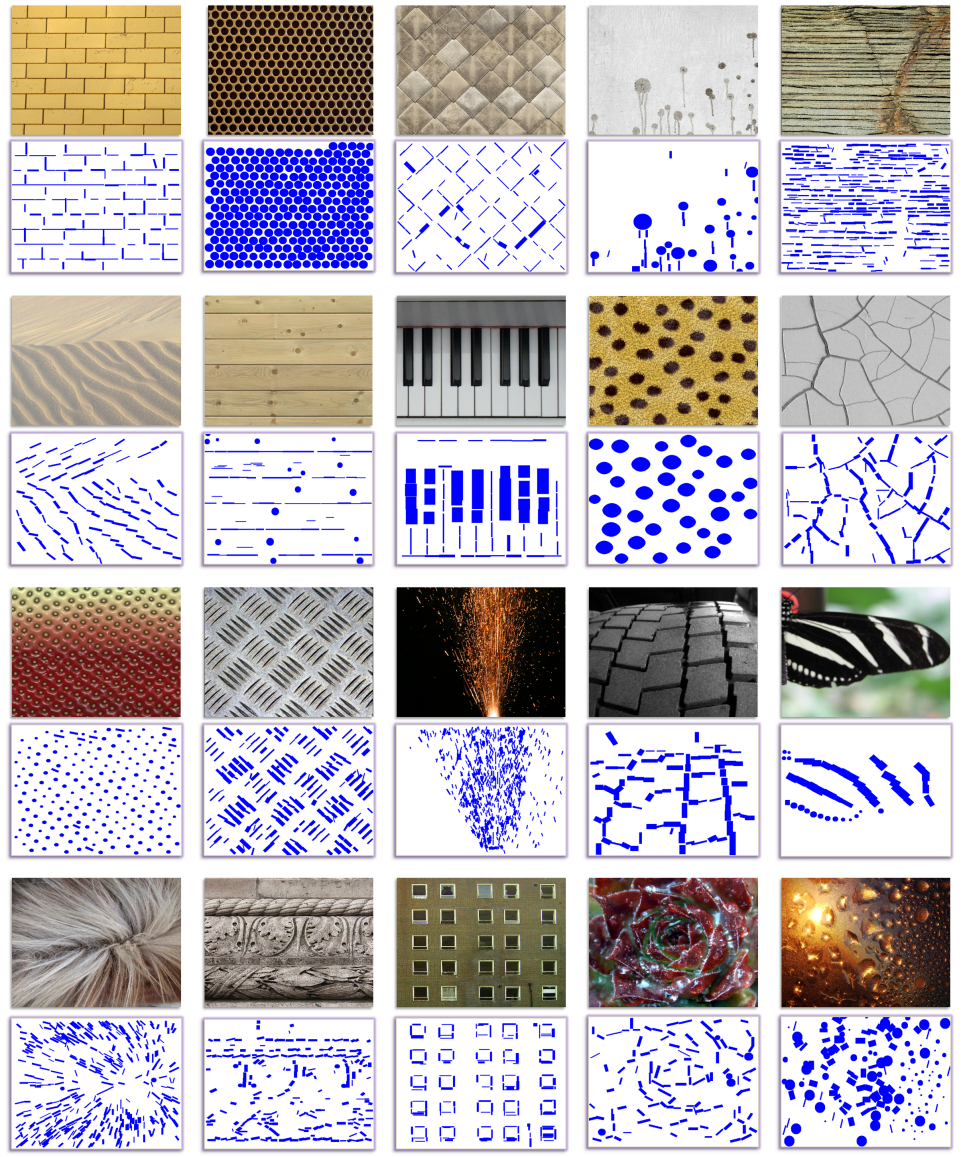


Figure 14: Texture analysis. Point processes can be used to sketch textures as configurations of simple geometric shapes. In particular, these configurations can help to analyze and synthesize textures as detailed in [GZW03, LGD10].

uration space $\mathcal{C}$ is of higher dimension since the objects are parametrically more complex.

Fig. 15 shows results obtained from laser scans of large urban and natural environments. 5.4 thousand trees are extracted on a 1km^2 urban area from 2.3 million input points. The performances could be improved by reducing the space $\mathcal{C}$ with a 3D-point process on manifolds, *ie* where the z-coordinate of points is determined by an estimated ground surface.

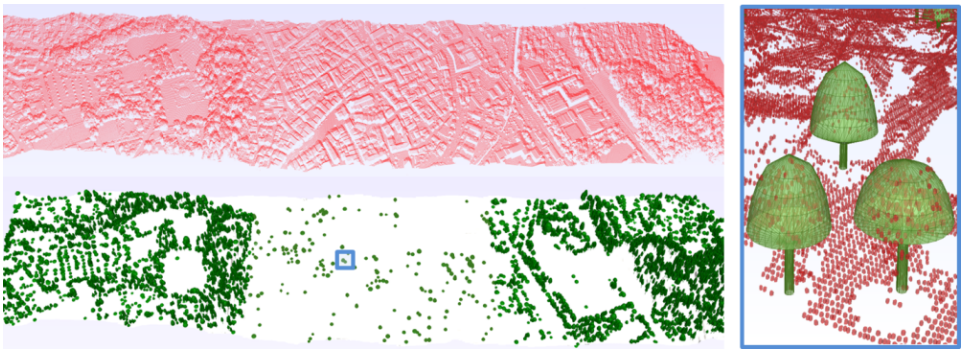


Figure 15: Tree recognition from point clouds by a 3D-point process specified by 3D-parametric models of trees. The algorithm proposed in [VL14] detects trees and recognizes their shapes from a large-scale Lidar scan (2.3M points), in spite of other types of urban entities, *eg* buildings, car and fences, contained in input point clouds. Note, on the cropped part, how the parametric models fit well to the input points corresponding to trees, and how the interaction of tree competition allows the regularization of the tree type in a local neighborhood.

Discussion. Several conclusions on point processes can be drawn from the research directions we explored. First geometric objects must remains simple as running time increase exponentially when object dimension increases. Second, geometric interactions must be manipulated with caution. In particular, complex interactions can easily generate an undesirable amount of model parameters, and also make the sampling difficult to achieve with highly non-convex energy forms. Experiments led in [CFL13] suggest geometric interactions must rather be integrated in the object definition itself when possible. Third, sampling point processes in vision is still at its early stage, and can benefit from important advances in the near future, in particular in terms of parallelization.

3 Surface reconstruction

Surface reconstruction is a traditional research topic which consists in recovering the surface of an object given some data measurements. The geometry processing community has deeply explored this topic by considering laser scans as inputs, whereas computer vision researchers have mainly addressed the problem from Multi-View Stereo images. Many different approaches have been proposed in the literature for reconstructing surfaces. Two main categories of approaches can be considered: smooth and primitive-based.

Smooth approaches recover C_1-surfaces using either implicit or explicit representations. Implicit methods indirectly describe surfaces using level-sets [KF98, KBH06, HK06, LB07]. They commonly rely on two key elements: (i) a 3D function computed from the input points allowing to both approximate and smooth the surface, as signed distances or radial basis functions, and (ii) a solver for extracting the surface, *eg* linear least squares or graph-cuts. Implicit methods are effective solutions but most of them require specific additional attributes associated to point locations such as oriented normals, lines of sight or measurement confidences. Explicit methods reconstruct surfaces using mesh-based structures such as Delaunay triangulations. Relying on the idea that points close to the surface are also close in space, these methods provide convincing results when the sampling is dense enough and hampered with only little amount of noise [ABK98, LPK09]. Several methods have been proposed to preserve sharp features by either preliminarily detecting smooth regions [FCOS05, JWS08] and sharp crease [SYM10] or inserting local shape priors into the reconstruction process [GSH+07].

Primitives-based methods have become popular in recent years. Efficient algorithms are now available for detecting geometric primitives [SWK07] (also called proxies) and for readjusting them according to global relationships such as as coplanarity, coaxiality or parallelism [LWC+11]. Primitive-based methods are a relevant alternative to smooth reconstructions when the inferred surfaces contain many canonical parts such as planar components, and for large data sets as dealing with primitives may improve computational efficiency. Chen et al. [CC08] and Chauve et al. [CLP10] use an arrangement of planes for approximating surfaces, which provides a rich solution space but only when all planes are perfectly detected. Missing planes may be completed by ghost components [CLP10], but this comes at the price of a lower surface accuracy. Vanegas et al. [VAB12] propose a method for reconstructing urban structures from laser range scans under the restrictive Manhattan-World assumption. Schnabel et al. [SDK09] reconstruct surfaces while filling holes from incomplete point sets through graph-cut based primitive extension by assuming that each hole can be entirely described by primitive arrangements. Another solution is to interactively complete or correct primitive-based methods, such as the user-assisted snapping approach propose in [ASF+12]. Note also that some methods recover structures from

surfaces [CSAD04]. The latter methods however require to preliminarily extract an accurate mesh from the input point set. Primitive-based methods are particularly attractive when dealing with large scenes containing canonical parts, but in general they remain less robust and flexible than smooth solutions. The first concern lies into the restrictive representation, as a complex scene can rarely be entirely described by a set of canonical primitives. The second concern lies into the reliability of the primitive detection step: an ideal primitive and primitive adjacency extraction with no under- nor over-detected parts cannot be guaranteed.

For a deeper literature review, surveys and benchmarks can be found in [BLN+13] from Laser scans, and in [SCD+06, SVHVG+08] from MVS images.

3.1 Hybrid surfaces

There is no one conventional way to measure the quality of reconstructed surfaces in computer vision and geometry processing problems. Depending on the application domain, the quality of reconstructed surfaces are judged differently. Accurate surfaces, *ie* surfaces with a low error to the measurements, are expected in cultural heritage for instance, whereas compact surfaces are preferred in real time applications as SLAM. In an urban context, we believe the quality of a surface can be measured as a combination of three criteria.

- Geometric accuracy. The surface must be as close as possible of the data measurements. Many different (point-based or photo-consistency) error metrics can be considered with more or less robustness to defect-laden data.
- Complexity. The surface must be as light as possible in terms of storage and compaction.
- Structure. The surface must preserve geometric structures existing within the observed scene.

In particular, the quality of a surface can be measured as a trade-off between geometric accuracy and complexity under structural constraints. We have explored this idea through a series of works [LKB10, LM12, LKBV13, LA13] based on the concept of *hybrid surfaces*.

Hybrid surfaces are assembled as a mixture of canonical parts by geometric primitives and non-canonical part by free-form surfaces. Such a surface combines both structured canonical parts idealizing the regular elements, and free-form parts representing either the non-regular elements of the scene or the undetected yet canonical parts of the scene. Even in presence of under-detection of primitives, the reconstruction is handled gracefully as no parts are missing. On under-detected areas the final reconstruction is simply less structured. Such a hybrid framework suggests that it is possible to

cumulate some advantages of both worlds (primitive and smooth). In particular, it is more compact, more scalable and more structured than smooth representations. It is also more robust than primitive-based approaches.

After discussing the primitive extraction from different inputs in Section 3.2, we present two models for reconstructing hybrid surfaces, one from point clouds (Section 3.3) and the other from MVS imagery (Section 3.4). More details can be found in [LA13] and [LKBV13] respectively.

3.2 Primitive extraction

In our use, the extraction of geometric primitives can be seen as a preliminary step before reconstructing surfaces. Low running times are thus expected, even in presence of big input data.

Only the surface primitives are considered here. Planes constitutes the most commonly used primitives. Such shape allows the description of many urban objects or object parts, as roads, roofs or facade components. Several non-planar primitives are also used, in particular spherical, cylindrical, conical and toroidal shapes. However non-planar primitives are relatively marginal for urban reconstruction problems compared to reverse engineering applications [AP10, SDK09].

Note that volume primitives, *eg* cuboids, are also used in the literature by Constructive Solid Geometry based approaches. Previous works [LDZPD10] have been realized in this direction for building reconstruction where the volume primitives are parametric models of building portions. Such primitives are however more complex than surface primitives in terms of parameters, leading to fastidious detection and arrangement operations.

Primitives are usually extracted from point clouds and meshes, but rarely from MVS images as the photo-consistency computation on large surface elements is extremely time-consuming. One prefer computing point clouds from MVS images, and then quickly extracting primitives from this point clouds. Primitive extraction from meshes and from point clouds are closely similar when meshes are dense. Meshes can be seen as point clouds for which the vertex normals and adjacency graph are known. Meshes also offer more possibilities of error metrics (facet-to-primitive or edge-to-primitive).

Strategies. Among algorithms used for extracting primitives, two parameters are commonly involved: a minimal number of inlier, and a fitting tolerance. The former guarantees to find primitives whose size is large enough. The latter is a tolerance error checking whether a point (or facet) is close enough to the primitive hypothesis to be considered as an inlier. Euclidean distances are usually chosen as error metrics. Primitives are said to be detected under tolerance ϵ if there are no outliers within ϵ distance to the primitives. Fig.16 illustrates the impact of these two parameters in terms of number of found primitives, occupancy and running times.

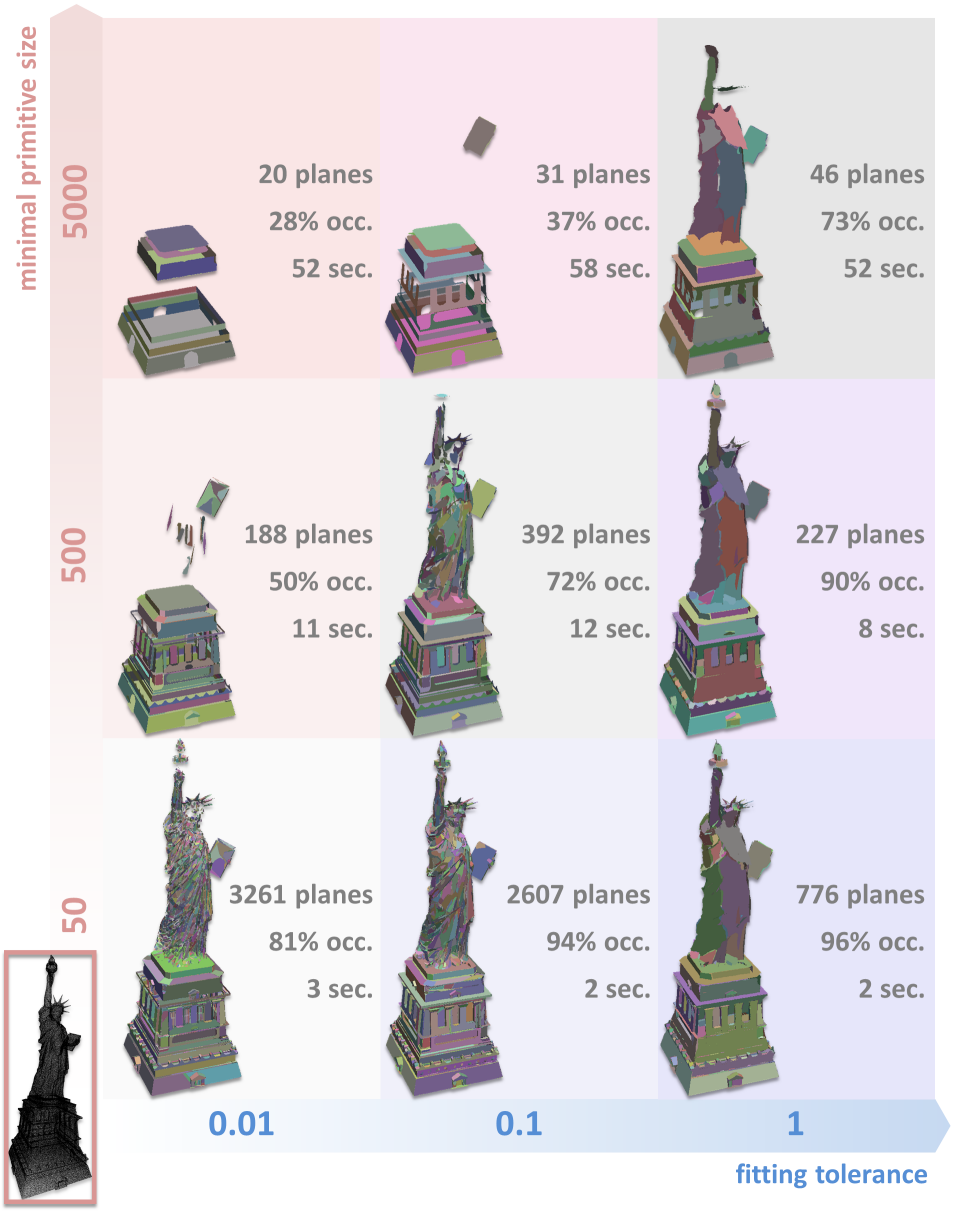


Figure 16: Plane extraction. Starting from a point cloud (1M points, bottom left), planes are extracted by the region growing procedure proposed in [LM12]. Fitting tolerance (in meter) and minimal primitive size (in number of inliers) are the two parameters impacting the plane extraction. A high fitting tolerance engenders a less accurate primitive extraction and a high occupancy (number of inliers to number of input points ratio). Increasing the minimal primitive size are more time-consuming and limits the number of found primitives.

We put our attention on four mechanisms among existing methods.

- *Ransac.* Ransac-based algorithms are probably the most popular for extracting primitives from point clouds containing outliers. They consists in proposing multiple primitive hypotheses from random subsets of the original data. The version proposed by Schnabel et al. [SWK07] presents competitive running times (few seconds for one million points), even if the results can be unstable with low fitting tolerances. Ransac-based algorithms constitute a good choice in presence of defect-laden data, in particular outliers and noise.

- *Region growing.* Region growing procedures consists in propagating a primitive hypothesis in a spatial neighborhood starting from a seed datum. The primitive hypothesis is progressively corrected during the propagation. These deterministic procedures are usually competitive with defect-free data, in particular with Laser/Lidar scans as shown in [LM12].

- *Accumulation space.* The projection of the data in accumulation spaces is also a conventional way for extracting primitives, in particular the Gaussian sphere and Hough accumulators. Primitive hypotheses are easily detected in the accumulation spaces by Mean Shift or similar clustering techniques. A notable advantage of this strategy is the possibility of regularizing the primitives at a global scale. However this strategy sometimes requires time-consuming post-processing to spatially distinguish the primitives, and cannot handle all types of primitives. In practice, Gaussian sphere accumulators are restricted to the extraction of planes [CC08].

- *Hypothesis-then-selection.* Other approaches address the problem from a more global point of view, typically in two successive steps. First primitive hypotheses are generated from the inputs which are eventually partitioned into clusters [PTJYS12]. Then one of these hypotheses is associated to each input point as a multi-label energy formulation, typically using Markov Random Fields. These approaches allow the insertion of model complexity considerations as well as geometric priors. Such approaches are well suited to homography detection which provides precious information to infer planar surfaces from images [FSB06]. In [LKB10], facets of an input mesh are grouped by curvature similarity using a Markov Random Field so that the optimal primitive can be found for each cluster using the first order approximation of the true Euclidean distance proposed by [MLM01].

Global regularities Imposing global regularities on plane hypotheses allows the models to be visually more consistent by re-adjusting plane positions

and orientations. Discovering global regularities is also of interest for reducing the number of primitives, and potentially the complexity of the space partition induced by primitives as illustrated on Fig. 17.

Several methods have been proposed in the literature for interactive geometric modeling [HK12], Lidar-based building reconstruction [ZN12], or more generally shape fitting under regularization constraints [LWC+11]. These methods are very efficient with individual objects containing few primitives, but are not adapted to the large-scale analysis, *eg* city modeling, leading to very high running times.

We present here a procedure based on barycentric operations for regularizing planar primitives in an urban context. Four types of pairwise interactions between planes are considered. By denoting P_1 and P_2, two planes having respective unit normals $\mathbf{n}_1$ and $\mathbf{n}_2$ and centroids c_1 and c_2, one can formulate these relationships under an orientation tolerance ϵ and an Euclidean distance tolerance d.

- *Parallelism.* P_1 and P_2 are ϵ-parallel if $|\mathbf{n}_1 \cdot \mathbf{n}_2| \geq 1 - \epsilon$
- *Orthogonality.* P_1 and P_2 are ϵ-orthogonal if $|\mathbf{n}_1 \cdot \mathbf{n}_2| \leq \epsilon$
- *Z-symmetry.* P_1 and P_2 are ϵ-Z-symmetric if $||\mathbf{n}_1 \cdot \mathbf{n}_z| - |\mathbf{n}_2 \cdot \mathbf{n}_z|| \leq \epsilon$, where $\mathbf{n}_z$ is the unit vector along the vertical axis
- *Coplanarity.* P_1 and P_2 are d-ϵ-coplanar if they are ϵ-parallel and $|d_\perp(c_1, P_2) + d_\perp(c_2, P_1)| < 2d$, where $d_\perp(c, P)$ represents the orthogonal distance between point c and plane P

The first three relationships are relating to the primitive orientations, and coplanarity is a particular case of parallelism with an additional relative positioning constraint. The notion of Z-symmetry matches the common assumption that connected components of roofs tend to share similar slope values.

Regularization proceeds as follows. We first regroup the primitives which are ϵ-parallel primitives into parallel clusters, and compute the average orientation of each cluster. For efficiency the next two steps act on clusters instead of individual primitive. We then construct an orthogonality graph with one node per parallel cluster, and one edge between two nodes when they are ϵ-orthogonal. We proceed similarly with a Z-symmetry graph. Without altering the structure of the graphs and the centroid of each primitive, we then alter the orientation of the nodes by propagating the orthogonality and Z-symmetry constraints greedily along the edges of the orthogonality graph (in general with a larger number of edges than the Z-symmetry graph), from large to small nodes (see numbers on the orthogonality graph depicted by Fig.17). Such greedy process propagates the regularization constraints without looping through the orthogonality graph.

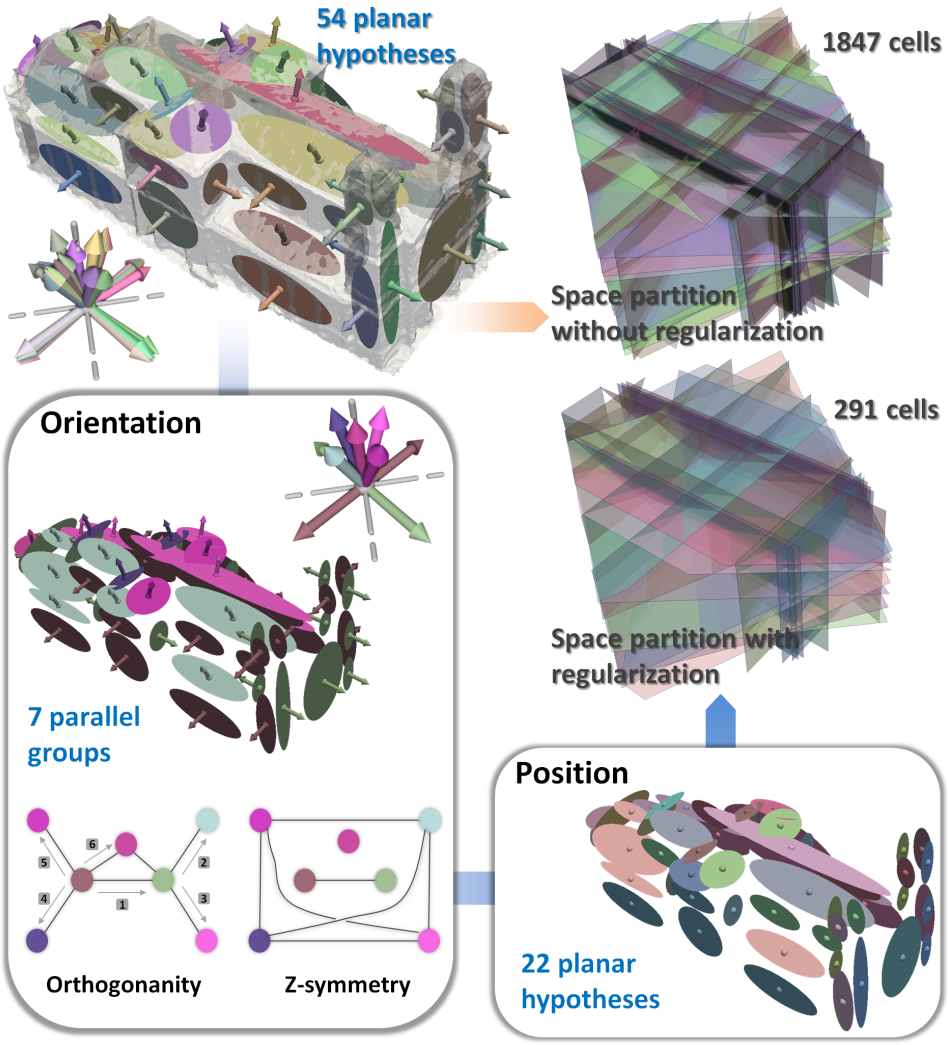


Figure 17: Regularization of planar primitives. Using a plane regularization procedure enforces structural coherence within plane hypotheses. Each color in the orientation step (respectively in the position step) represents a parallel group (resp. a set of coplanar planes). Plane normals are re-adjusted during the orientation step by propagating orthogonality and Z-symmetry constraints from the biggest parallel group to the smallest one (see numbers on the orthogonality graph). Additionally, in case of space partitioning from the planar hypotheses, this procedure avoids dense and irregular set of cells (top right), and strongly reduces the number of cells in the space partition.

The size of a node refers to the total area of its primitives. Denote respectively by source and target node, a pair of nodes is altered by the propagation. The initial orientation of the target node is altered by constraining its normal to match the constraints (orthogonal and sometimes also Z-symmetry) with respect to the source node. When only one constraint is propagated we choose the constrained orientation that best aligns to the initial orientation. When the two constraints are propagated there is in general a unique orientation that is both orthogonal and Z-symmetric. When no solution exists due to relationships contradictions along edges of the graph or when the constrained orientation deviates too much from the initial orientation (dot product between initial and altered normals lower than $1-\epsilon$) we undo the alteration and restore the initial normal. Finally, we also detect d-ϵ-coplanarity among parallel clusters and compute new primitive positions by clustering, in 1D along the normal of each parallel cluster, the area-weighted centroid of each primitive after projection onto the said normal.

3.3 Reconstruction by point set structuring

The surface reconstruction algorithm proposed in [LA13] considers as inputs a raw point set and a configuration of planar primitives extracted under a tolerance ϵ. The algorithm relies on a two-step strategy. First, the input point set is structured from the extracted primitives. Second, the surface is reconstructed from the structured point set using a min-cut formulation over a 3D Delaunay partitioning of the space.

Structuring. Given a configuration of planar primitives extracted under a tolerance ϵ, the structuring process turns the input point set into another point set in which each point is associated to one of the four structural types, *ie planar*, *crease*, *corner* and *clutter*, as depicted by Fig. 18. A point labeled as *planar* (resp. *crease* and *corner*) is associated to one (resp. two and three or more) planar primitives. The structuring process acts on three key points:

- *Meaning insertion:* Each point is enriched with structural information related to its associated extracted primitives (zero, one or more) and their adjacencies. This information is used in subsequent reconstruction processes.
- *Structure preservation under space partitioning:* Points are sampled so that the structures induced from the extracted primitives are preserved when subdividing the space with a partitioning scheme, here a Delaunay 3D-triangulation.
- *Simplification:* The point set is re-sampled on canonical parts using the primitives, without losing the details on the free-form parts which are kept untouched.

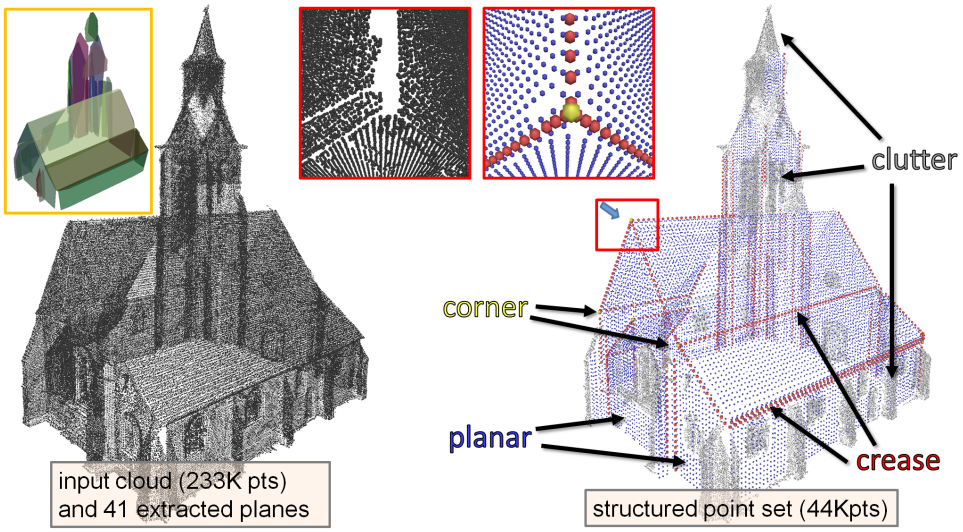


Figure 18: Structuring principle. Plane anchors (blue), creases (red) and corners (yellow) are positioned in the new point set to describe the main structures of the building. The other components such as windows or doors are defined as clutter points (grey).

The structuring process consists in replacing the points fitted to the primitives, *ie* inliers, by an ideal layout of points, both light and preserving the primitive surfaces in the Delaunay triangulation. An occupancy binary 2D-grid projected in the planar primitive is created. The width (side length) of a unitary square surface element of the grid is denoted by L_p. A surface element of the grid is marked occupied if at least one fitted point orthogonally projects within its domain or, subsequently, if it is surrounded by only occupied elements. The centers of all occupied elements form the new layout of points whose structural type is *planar*. In order for the detected planes to appear in the final model, the width L_p must be chosen so that the subsequent Delaunay triangulation will link the *planar* points with triangles. This linking condition is guaranteed when the equatorial circumsphere of these triangles is empty, *ie* when $L_p < \sqrt{2}\epsilon$. Once *planar* points are sampled, *crease* points are created between adjacent primitives in order to consolidate their connection. The adjacency relationship between two primitives is defined using the K-nearest neighbor (KNN) graph of the input points. Two primitives are said adjacent if at least two points fitted each to one of the two primitives are mutual neighbors in the KNN graph. *Crease* points are sampled uniformly along the intersection line of each pair of adjacent primitives by a similar process of the planar point sampling. *Corner* points are also positioned when detecting 3−cycles from the primitive adja-

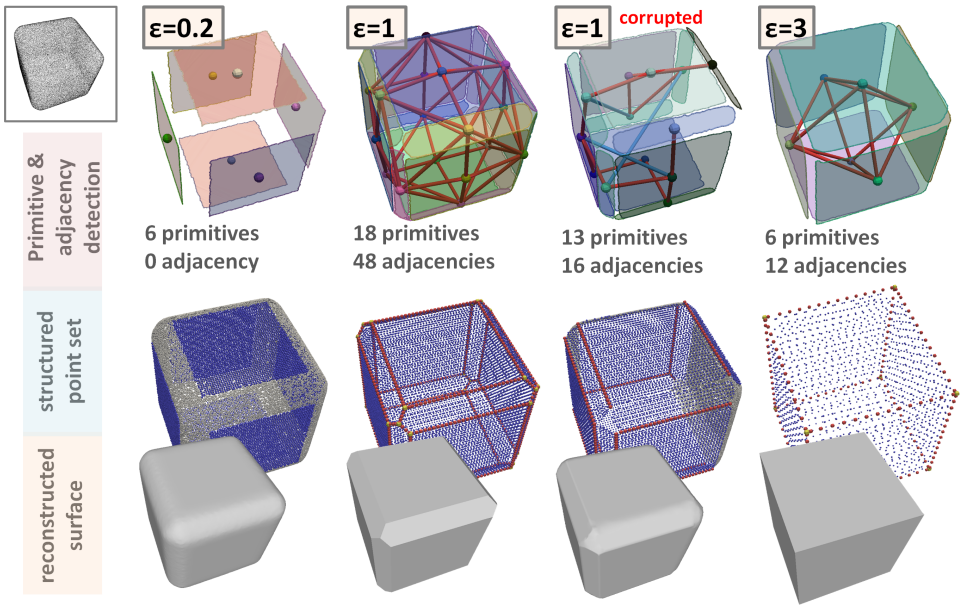


Figure 19: *Smooth cube* structured with different ϵ-values. Increasing ϵ progressively structures the input point set (top left) while maintaining a coherent reconstructed surface. In the third example, both primitives and adjacencies have been randomly corrupted (5 primitives and 6 adjacencies are removed, and 3 wrong adjacencies depicted as blue segments are added). While under- and over-detection of primitives and adjacencies reduce the quality of the structuring, by either omitting or overly creating creases and corners, this does not hamper the reconstruction thanks to the free-form components.

cency graph. The input points which have not been detected as belonging to planar primitives are inserted into the structured point set with the label *clutter*. The point set structuring is controlled by the tolerance parameter ϵ. As illustrated by Fig.19, increasing ϵ progressively structures the point set while reducing the amount of clutter points. When the point sets are ideally and fully structured (*eg*, second and fourth examples in Fig. 19), the surface can be straightforwardly extracted by basic polygonalization. However, such cases rarely occur from real-world data as free-form elements and under/over-detections of primitives and adjacencies are common. A robust procedure is required for extracting the surfaces.

Surface extraction. The surface extraction step relies on the structured point set. The general framework builds on the creation of a space partition from which each volume element is labeled either inside or outside the

inferred surface.

The space subdivision is obtained by constructing a 3D Delaunay triangulation from the structured point set. Such a space partition provides us with several relevant properties. Constructed from the structured point set, the triangulation preserves the structures both in terms of geometry (tetrahedra do not intersect the surfaces induced from the primitives) and in terms of meaning (each vertex of the 3D-triangulation inherits from a structural type assigned during structuring). The partitioning also has the advantage of being light as in general the structured point set comprises fewer points than the input point set.

In order to extract the surface from the 3D Delaunay triangulation, a min-cut formulation is used to find the inside/outside labeling of the tetrahedra and to deduce the surface as the interface between inside and outside. This graph-cut method has been commonly used in surface reconstruction either from regular space partitions [HK06, LB07] or from data-driven partitions [LPK09]. This method guarantees a hole-free and intersection-free surface, as well as low running times.

Let us consider a graph $(\mathcal{C}, \mathcal{F})$. $\mathcal{C} = \{c_1, ..., c_n\}$ is the set the cells (or tetrahedra) induced by the 3D Delaunay triangulation, corresponding to the nodes of the graph. $\mathcal{F} = \{f_1, ..., f_m\}$ is the set of triangular facets existing between two cells of the Delaunay triangulation, representing the edges of the graph. A cut in the graph $(\mathcal{C}, \mathcal{F})$ consists in separating the set of cells $\mathcal{C}$ in two disjoint sets $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$ such that $\mathcal{C} = \mathcal{C}_{in} + \mathcal{C}_{out}$ and $\mathcal{C}_{in} \cap \mathcal{C}_{out} = \emptyset$. The set of edges between $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$ corresponds to a set of triangular facets forming a surface $\mathcal{S} \subset \mathcal{F}$.

In order to measure the quality of the surface $\mathcal{S}$ induced by the cut $(\mathcal{C}_{in}, \mathcal{C}_{out})$, we introduce a cost function C of the form

$$C(\mathcal{S}) = \sum_{f_i \in \mathcal{S}} a(f_i)\, Q(f_i) + \sum_{c_k \in \mathcal{C}_{in}} P_{out}(c_k) + \sum_{c_k \in \mathcal{C}_{out}} P_{in}(c_k) \qquad (18)$$

where $Q(f_i)$ is a non-negative quality function of the facet f_i weighted by its area $a(f_i)$. Q penalizes facets whose vertex labels are structurally non coherent. The product $a(f_i)\, Q(f_i)$ represents the weight put on the edge f_i in the graph. P_{in} and P_{out} are prediction functions penalizing unexpected cell labels according to visibility considerations. They allow the insertion of weights between the nodes of $\mathcal{C}$ and two artificial nodes called the source and the sink so that the optimal surface is not reduced to the trivial empty solution. These functions act as a data term in the conventional energy formulations. Details on the formulation of these terms are given in [LA13]. The optimal cut minimizing the cost $C(\mathcal{S})$ is obtained using the max-flow algorithm [BK04].

Surface simplification. The complexity of the obtained hybrid surface depends on both the free-form/canonical area ratio and the tolerance param-

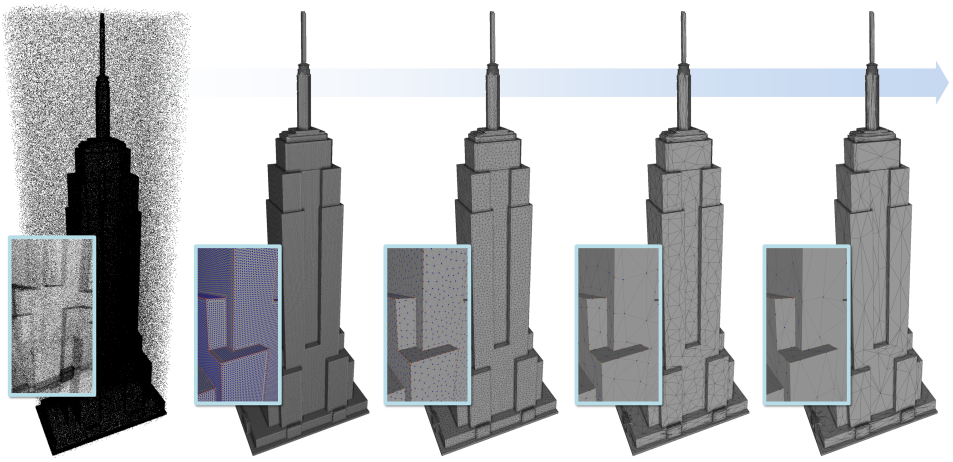


Figure 20: Surface simplification. An edge collapse procedure exploits the structural type of vertices so that the surface is simplified without loss of geometric accuracy (see close-ups).

eter ϵ. However the surface can be easily simplified by exploiting the semantic information of the vertices. An edge-collapse procedure is then specified by attributing either an edge length based cost to the edges connecting two identical planar or crease vertices, the created vertex being determined by the edge mid-point, or an infinite cost value to the other edges. As illustrated by Fig. 20, this procedure allows us to reduce the complexity without any loss of accuracy as the free-form components are preserved.

Results. The algorithm is designed to generate consistent results even in case of defect-laden primitive detection. Through the hybrid aspect, the reconstructed surface remains coherent even in high under-detection situations where only a few primitives are detected. In presence of noise, the canonical parts are nicely preserved, albeit the free-form parts are hampered with noise. In particular, the main planar components, *ie* the walls, are structured on *MVS1* and *MVS2* models in Fig. 22 whereas the details above the tolerance ϵ, *ie* the ornaments, are recovered. Running times are provided in Fig. 22 and 21 on two models as a function of ϵ. Our algorithm takes an order of 30 seconds to reconstruct a surface from one million points with a high ϵ-tolerance, after consuming few seconds to extract the planar primitives. The surface extraction step is often both more time and memory consuming than the structuring step. The running times strongly depend on the structuring level of the input point set.

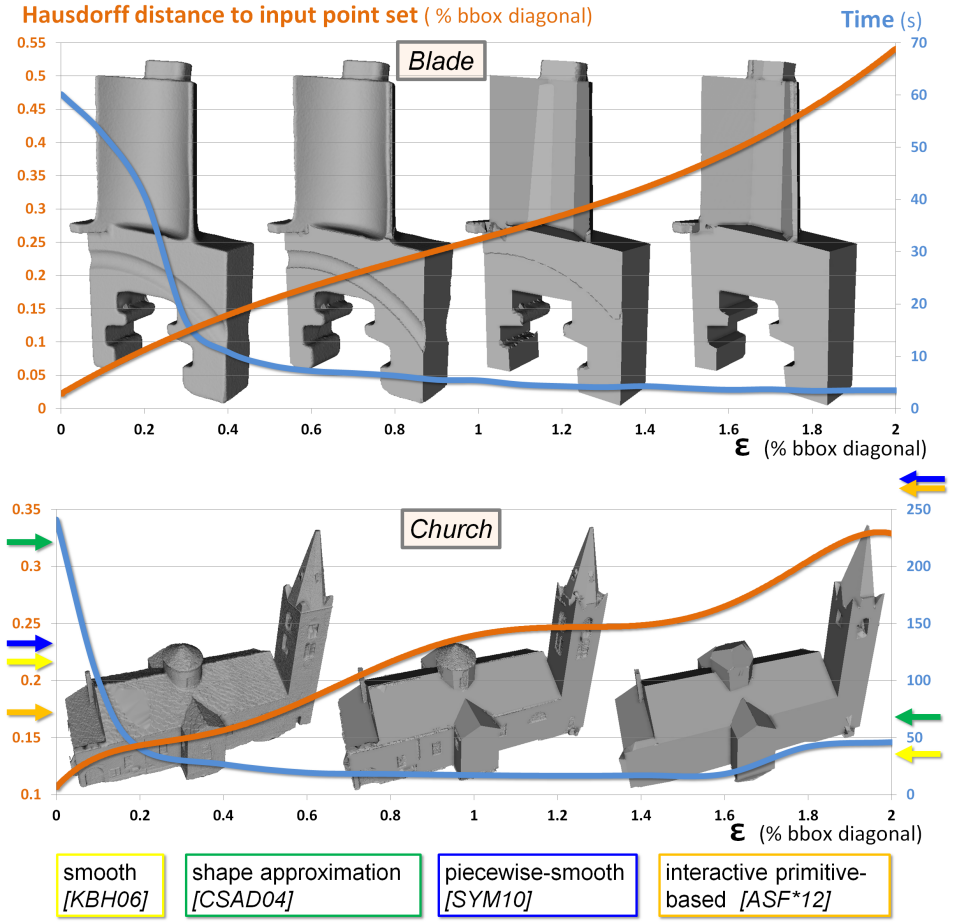


Figure 21: Impact of ϵ. When ϵ is close to zero, few primitives are detected; the obtained surfaces being similar to smooth reconstructions. Increasing ϵ progressively structures the surface while preserving the details over an ϵ tolerance. The running time decreases as the structured point set becomes lighter. For smooth shapes, *ie Blade*, the Hausdorff distance to the input point set increases relatively proportionally to ϵ. In case of urban scenes, *ie Church*, the distance evolves in function of the different urban scales, the stagnation around $\epsilon = 1$ occurring after that the minor elements as windows or doors have been digested in the major components as walls or roof sections. The hybrid model at $\epsilon = 0.2$ is particularly interesting, competing well with existing approaches in terms of accuracy and running time (see colored arrows).

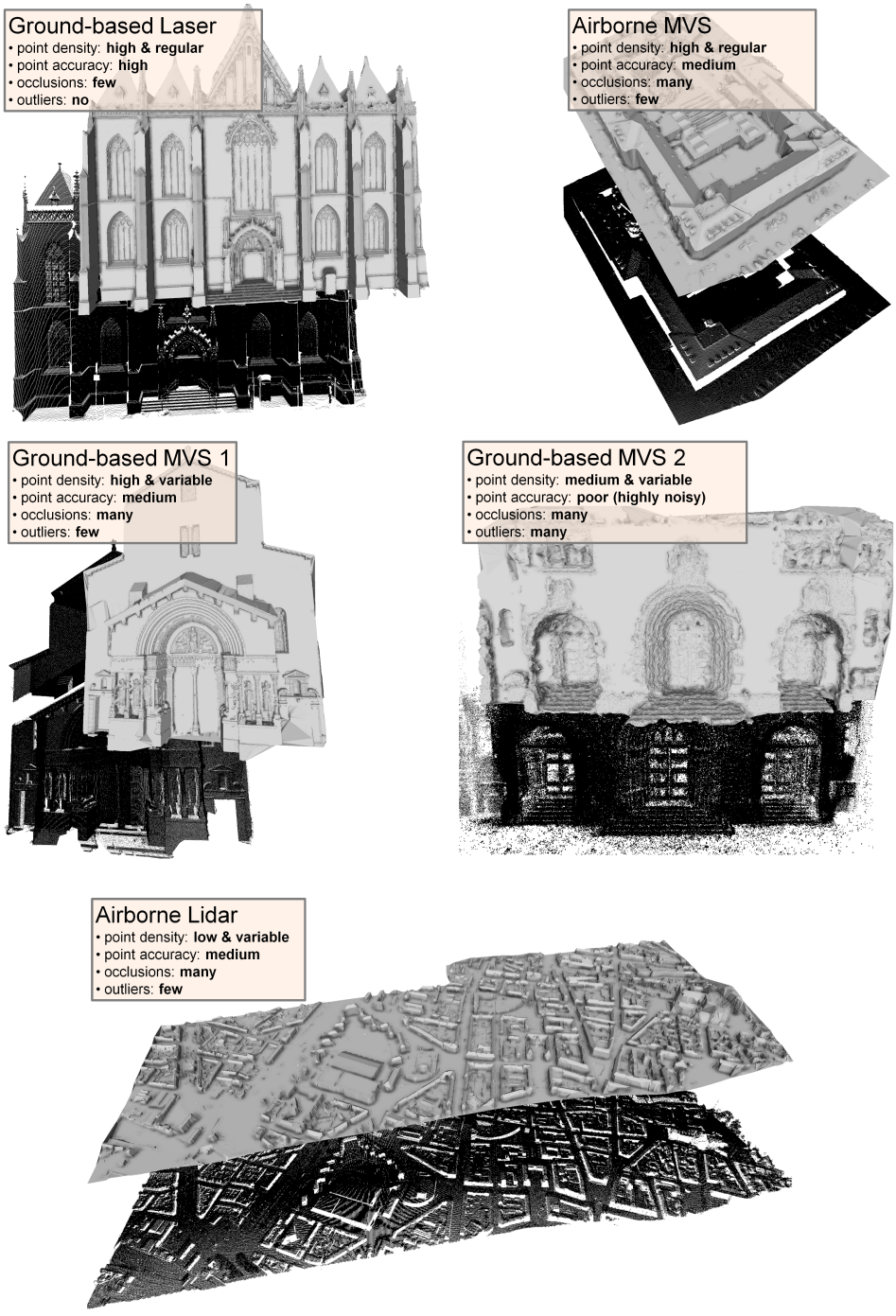


Figure 22: Reconstruction of urban scenes from different acquisition systems. The obtained hybrid surfaces recover the main structure of the scenes while preserving the details.

	Ground Laser	Airborne MVS	Airborne Lidar	Ground MVS 1	Ground MVS 2
# input points	1.49M	1.97M	1.69M	2.43M	0.57M
# primitives	166	154	3257	55	4
# structured points	1.19M	0.24M	0.87M	1.26M	0.36M
Structuring time (sec)	19	15.4	51.6	18.9	3.4
extraction time (sec)	216	31.6	353.1	434	40.3
Memory peak (Mb)	3,300	1,120	2,220	3,700	990

Table 1: Performances in terms of running time and memory of models from Fig. 22. The tests have been performed on an Intel Core i7 clocked at 2GHz. Data loading and computation of the Riemannian graph are excluded from the running times.

Discussion. This algorithm is not designed to reconstruct non-manifold surfaces nor surfaces with complex occlusions or invisible parts from the scanning directions. The algorithm is also not suited to missing data (large holes in the input point set) where the reconstruction problem is even more ill-posed. In its current form the user-specified error tolerance provides a way to vary the level of details. However, the accuracy curves obtained in Fig. 21 whose shapes make appear different plateaus suggest it can be possible to automatically select the error tolerance given a targeted urban scale. In particular, we wish to research on ways to construct a scale-space with smooth transitions between the scales.

3.4 Reconstruction by multiple shape sampling

The second surface reconstruction algorithm, presented in [LKBV13], considers as inputs some multi-view stereo images and an initial rough mesh whose topology is supposed to be correct. A two-step strategy is adopted, consisting first in segmenting the initial mesh-based surface and second in sampling primitives and mesh patches simultaneously on the obtained partition. A preliminary segmentation is important because it allows us to significantly reduce the complexity of the problem. These two stages are embedded into a general iterative procedure which provides, at each iteration, an increasingly more refined hybrid surface. Extracted 3D-primitives are collected along iterations whereas mesh patches are subdivided and used as the initialization of the next iteration. The procedure stops when the mesh subdivision generates facets which are too small to be accurately matched with the images.

We focus here on the second step of the algorithm, *ie* the simultaneous sampling of primitives and mesh patches. More details on the overall algorithm can be found in [LKBV13].

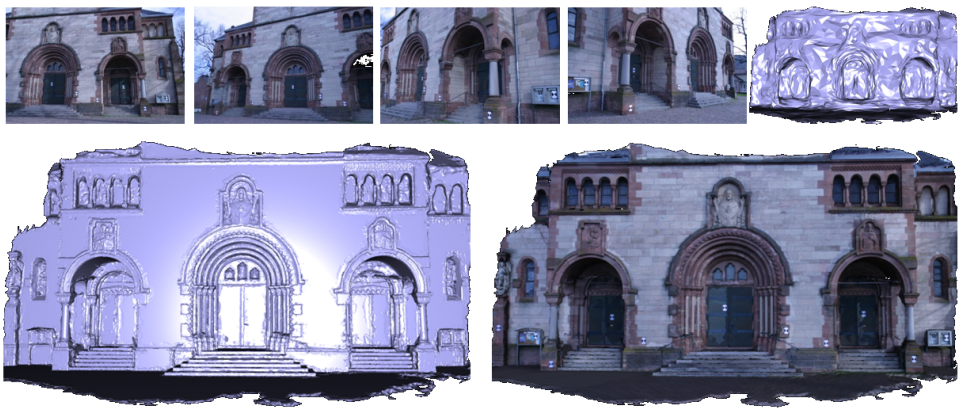


Figure 23: Herz-Jesu-P25. (top) The inputs are composed of multi-view stereo images and an initial rough mesh. (bottom) the hybrid surface is reconstructed by mesh-patches describing free-form elements as ornaments or statues, and by geometric primitives recovering regular components as walls or facade columns.

Energy formulation. Let $x^{(0)}$ be the initial rough mesh based surface segmented in N clusters by using the algorithm presented in [LKB10]. Let x be a hybrid surface defined as a set of N_m mesh patches and N_p primitives, each of them associated with an above-mentioned cluster such that $x = (x_i)_{i\in[1,N]} = ((m_i)_{i\in[1,N_m]}, (p_i)_{i\in[N_m+1,N]})$ and $N_m + N_p = N$. m_i represents the mesh patch associated with the cluster i. The primitive p_i is defined by a primitive type chosen among a set of basic geometric shapes (*plane, cylinder, cone, sphere* and *torus*), and its parameter set. The solution space is defined as a union of 6^N continuous subspaces, each containing a predefined object type per cluster (*ie* 5 primitive types and 1 mesh with triangular facets). In the following, we call *object*, an element x_i of x which can be a primitive or a mesh patch. The quality of a hybrid surface is measured through an energy formulation as detailed below.

Formulating an energy U from the configuration space is not a conventional problem because several kinds of objects must be simultaneously taken into account. In addition, U must verify certain requirements, in particular the differentiability in order to guide the exploration of solution space with gradient considerations. The energy is expressed as an association of three

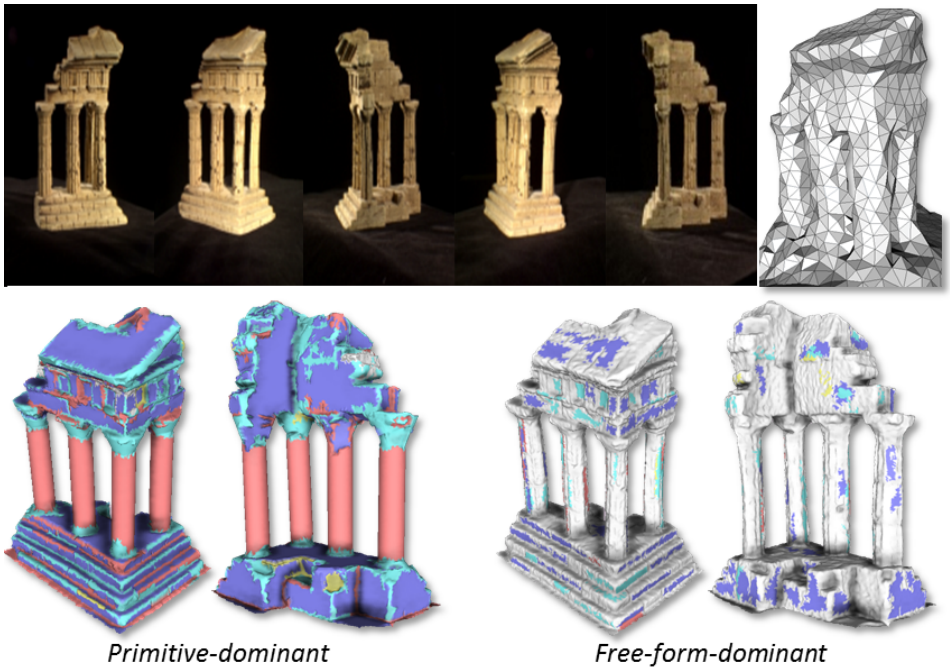


Figure 24: Temple model. (top) input images and a rough visual hull as initial surface; (bottom): two results, one with primitive dominance and one with free-form dominance. Color code: *purple=plane, pink=cylinder, blue=cone, yellow=sphere, green=torus, gray=mesh*. The free-form dominant version has a 0.48 mm accuracy and a 99.7% completeness on the Middlebury benchmark [SCD+06] whereas the primitive dominant version has a 1.03 mm accuracy and a 95.7% completeness. Indeed the reconstruction of non-regular columns and pieces of walls by cylinders and planes can engender a loss of accuracy.

terms by:

$$U(x) = \sum_{i=1}^{N} U_{pc}(x_i) + \beta_1 \sum_{i=1}^{N_m} U_s(m_i) + \beta_2 \sum_{i \bowtie i'} U_a(p_i, p_{i'}) \qquad (19)$$

where U_{pc} measures the coherence of an object surface with respect to the images, U_s imposes some smoothness constraints on the mesh based objects, U_a introduces structural knowledge on urban scenes for placing the 3D-primitives, $i \bowtie i'$ represents the primitive pairwise set and (β_1, β_2) are parameters weighting these three terms.

- *Photo-consistency* U_{pc}. This term, based on the work of [PKF07], computes the image re-projection error with respect to the object surface. A parameter is introduced in this term to tune the 3D-primitive to mesh-patch ratio, as illustrated in Fig. 24.

- *Mesh smoothness* U_s. This term allows the regularization of mesh patches by introducing smoothness constraints. We use the thin plate energy E_{TP} proposed in [KCVS98] which penalizes strong bending. In particular, this local bending energy is efficient for discouraging degenerated triangles.

- *Structural priors* U_a. This term aims to improve both the visual representation by realistic layouts of 3D-primitives, and compensate for the lack of information contained in the images by favoring regular structures. It is expressed through a pairwise interaction potential which favors parallel and orthogonal primitives.

Sampling by Jump-Diffusion The search for an optimal configuration of objects is performed using Jump-Diffusion [GM94b]. Jumps between the subspaces are performed by switching the type of an object in the configuration x, *eg* a mesh patch to a cylinder or a torus to a plane. The new object is proposed randomly. This dynamic, which consists in creating bijections between the parameter sets of the different object types [Gre95], is sufficient to explore the various subspaces of our problem.

Two diffusion dynamics are considered to explore each subspace. The Mesh adaptation dynamic allows the evolution of mesh based objects using variational considerations. The energy gradient restricted to mesh based objects (*ie* $\nabla_{m_i} U = \nabla_{m_i}(U_{pc} + \beta_1 U_s)$) is computed by using the discrete formulation proposed by [PKF07]. Brownian motions, which drive the diffusion equations, allow us to ensure the convergence towards the global minimum but make the process extremely slow. In practice, we found that the Brownian motion is not necessary to explore mesh based object configurations because the switching dynamic which proposes random mesh patches is efficient enough to escape from local minima. The primitive competition dynamic selects relevant parameters θ_i of primitive based objects p_i without changing their types. It is particularly efficient to accelerate the primitive structuring while keeping the object coherent to the images. The gradient related to this dynamic is given by $\nabla_{\theta_i} U = \nabla_{\theta_i}(U_{pc} + \beta_2 \sum U_a)$ where $\nabla_{\theta_i} U_{pc}$ is approximated using [MLM01].

Results The obtained hybrid surfaces provide interesting representations of the scenes. The main regular components such as walls, columns, vaultings or roofs are largely reconstructed with 3D-primitives during the first iterations of the refinement procedure (*ie* on the models at low resolution). The method has been compared to the standard multi-view stereo algorithms from a facade reconstruction benchmark [SVHVG+08]. The cumulative error histograms presented in Fig. 25 show that we obtain the first and second best accuracies for Herz-Jesu-P25 and Entry-P10 respectively.

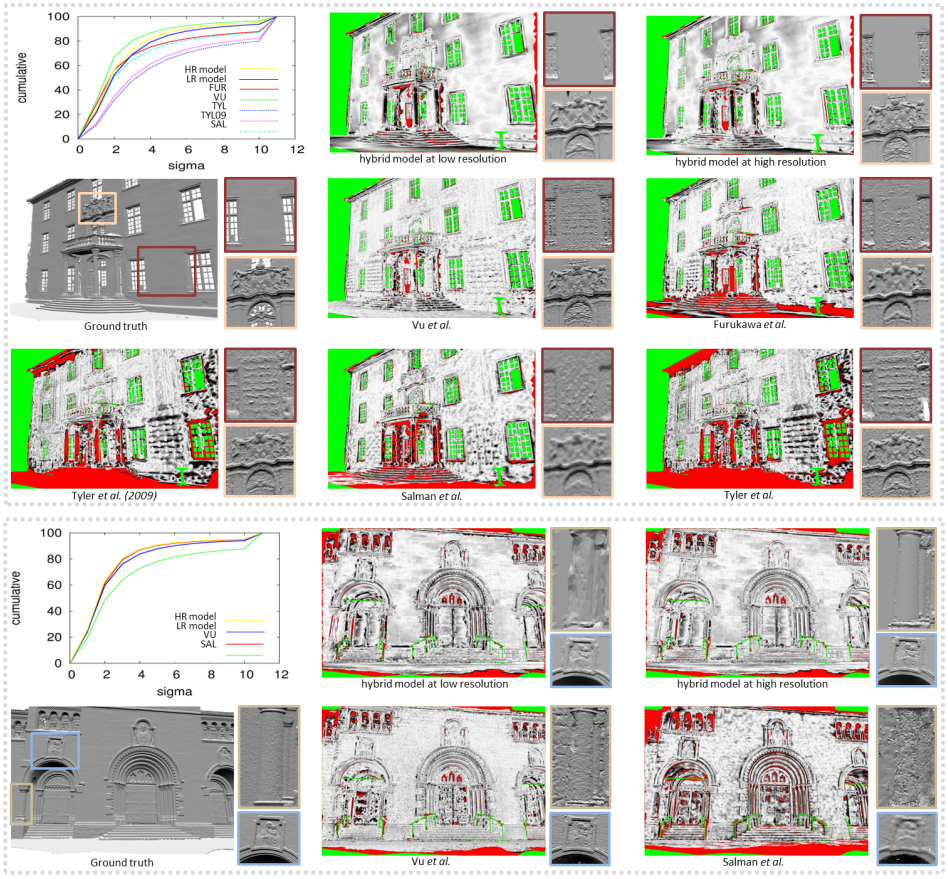


Figure 25: Accuracy evaluation on Entry-P10 (top) and Herz-Jesu-P25 (bottom). The cumulative error histograms are measured with respect to the standard deviation Σ of the ground truth accuracy [SVHVG+08]. The error maps of the different models are established with respect to the ground truth (white=low, black=high, red=off-the-scale). Our high resolution hybrid models obtain the first and second best accuracies for Herz-Jesu-P25 and Entry-P10 respectively.

The experiments reveal several interesting points illustrated in the close-ups of Fig. 25. First, the regular structures which are partially occluded in the images are more accurately reconstructed by using 3D-primitives than by standard multi-view stereo algorithms, as shown on the column crops. Second the *trompe l'oeil* structures (see for example the textures representing fake ornaments on the walls of Entry-P10) are correctly reconstructed contrary to the mesh-only models which are based on a local analysis of the scene. Another advantage of this hybrid representation is the model com-

plexity. The level of compaction depends on the type of scenes: the more regular the scene structures, the more compact the hybrid surface. Compaction is not obtained at the expense of the visual quality as shown in Fig. 23. The presence of large primitives in our models even simplifies the texturing process in comparison with the mesh-based representations. In particular, these results offer interesting perspectives for integrating both detailed and compact models in public visualization softwares.

Discussion Structural priors remain relatively simple. In particular, they are not designed to recover geometric repetitions or symmetries (see, for example, the set of small windows on the background tower of the Entry-P10 which are not represented by similar primitives). The initial surface has also to be topologically close to the real surface to guarantee good results. A more complex approach would be possible by embedding the segmentation stage into the sampling procedure. New kernels dealing with partition changes, *eg* object merging/splitting dynamics, could be introduced in order to unify segmentation and object sampling. However, such an algorithm would be performed at the expense of robustness and running times.

4 Large-scale city modeling

We address in this section the specific problem of city modeling from airborne data, in particular from multiview stereo imagery and Lidar scanning. Some works have explored building reconstruction directly from multiview stereo images, *eg* [BZ00, SMVG02]. This strategy is ambitious as the algorithmic complexity is particularly high. No entirely convincing results have been provided by such approaches at the scale of a city. Depth maps generated from multiview stereo images constitute an indirect but more efficient way of modeling cities. These 2.5D view-dependent inputs are usually highly noisy. The works of [ZBKB08] and [LDZPD10] propose compact 3D-models of buildings from depth maps; the former labels a 2D space partition driven by geometric primitives whereas the later assembles 3D-blocks of urban structures using Monte Carlo sampling. The major limitation of depth maps comes from the difficulty in distinguishing buildings and high vegetation.

Lidar data became very popular in the mid-2000 leading to a series of works mainly focused on parsing building components and extracting accurately building contours, *eg* [VKH06, PY09, TMT10, VLZ11, ZN12, LGZ+13]. Points generated by Lidar acquisition systems are usually very accurate, but contrary to depth maps, they are geometrically unstructured and are free of radiometric information. Planar primitives represent the favorite geometric tools for recovering roofs and facades. Efforts have been made towards the parsing of planes, *eg* [TMT10], the discovery of global regularities [ZN12], or the geometric decomposition of building components [LGZ+13].

Matei *et al.* [MSS+08] and Poullis *et al.* [PY09] propose flat roof models adapted to "Manhattan-world" environments [CY00]. Both approaches put efforts in segmenting the buildings and simplifying their boundaries, either by estimating building orientations [MSS+08] or by using statistical considerations [PY09]. A more general building representation is proposed in [ZN10] where a mesh simplification procedure based on dual contouring is used. Although this approach increases flexibility, semantic information are lost: a simple planar roof section can be described by many triangular facets with different normal orientations. These methods mainly focus on building reconstruction from a geometric point of view. Semantics contained in the urban scenes is poorly exploited, except in [TMT10] where tree detection is also considered.

Existing methods are usually designed for a specific type of urban scenes, *eg* American residential areas [LGZ+13] and financial districts of big cities [MSS+08, PY09]. Most of these approaches rely on strong geometric assumptions limiting the reconstruction flexibility. "Manhattan-world", flat-roof, and facade parallelism constitute the most commonly used geometric assumptions. Even if such assumptions reduce algorithmic complexity, they remain too specific to reconstruct random urban landscapes. Also, these methods provide a sparse description of urban scenes. They are focused on the building modeling task and disregard all the other objects which can be found in an urban scene such as trees, or even sometimes ground surfaces by assuming a constant altitude over the global scene.

4.1 General strategy

we presents two algorithms for modeling cities, one from Lidar data, and the other from MVS imagery. These algorithms are designed to be suitable to any kind of urban scenes. They follow a similar strategy described in the following.

- *Object identification.* Input data are classified into different objects of interest, *ie* buildings, trees and ground. A fourth class called clutter can be considered to identify outliers contained in the data and small urban components which temporarily perturb the scene such as cars, fences, wires, roof antennas or cranes. A Markov Random Field (MRF) with pairwise interactions is used to label each datum by one of the classes of interest. The quality of a label configuration l is measured by the energy U of the standard form:

$$U(l) = \sum_{i \in S} D_i(l_i) + \gamma \sum_{\{i,j\} \in E} V_{ij}(l_i, l_j) \tag{20}$$

  where S and E corresponds to the set of elements to label and their adjacency graph respectively, D_i the unary data term formulated as

a combination of local geometric features, and V_{ij} the propagation constraints which bring spatial consistency in the labeling. A Graph-Cut based algorithm [BVZ01] is used to quickly reach an approximate solution close to the global optimum of the energy.

- *geometric primitive extraction from buildings.* Geometric primitives are extracted from elements labeled as building. A region growing detailed in [LM12] is considered to extract planes and other types of primitives. Optionally, the regularization procedure described in Section 3.2 is performed subsequently to primitive extraction.

- *Building modeling by primitive arrangements.* The extracted primitives are arranged in 3D or 2.5D to generate compact meshes that model buildings. This part constitute the main contribution to the proposed systems.

- *Tree and ground modeling.* Trees are modeled either by sampling 3D templates of trees with point process (see Section 2), or, more simply, by template matching. The template is generally a simple ellipsoidal tree model whose compaction and rendering are well adapted to large urban scenes. For a street-view representation, one can imagine proposing a more realistic tree modeling, *eg* [XGC07]. A standard meshing procedure is used to model the ground. A grid of 3D-points is created from a spatial sub-sampling of the cells labeled as *ground*. It allows an accurate description without imposing any geometric constraints on the surface.

Both object identification and building modeling by primitive arrangements for Lidar-based and MVS-based algorithms are presented next.

4.2 Airborne Lidar

The first algorithm, proposed in [LM12], considers airborne Lidar scans as input data. These point clouds can have different densities varying from 2 to 17 pts/m^2.

Object identification. Input Lidar scans are classified by labeling each point by one of four classes of interest, *ie building*, *vegetation*, *ground*, and *clutter*, as shown in Fig. 26. The energy formulated in Eq. 20 is specified by S, the set of input points, and E the set of edges in the Riemannian graph, *ie* pairs of neighboring points. The potential V_{ij} is defined by the standard Potts model [Li01], whereas the unary data term E_d is based on the combination of several local geometric features used considered to discriminate the different classes of interest. The features are in value in the interval $[0, 1]$.

- *Local non-planarity* f_p represents the Euclidean distance between the point and the optimal 3D-plane computed among its neighbors. The response to this feature is supposed to be low in the case of buildings and ground.

- *Elevation* f_e allows the distinction between the ground and the other classes. This feature corresponds to the altimetric variation between the point and its planimetric projection on an estimated elevation map of the ground.

- *Scatter* f_s measures the local height dispersion of the points. It provides a high value in the case of trees and also some undesirable urban components. This feature is defined as the minimal principal curvature mean of the considering point and its neighbors.

- *Clutter* f_c is devoted to outliers and undesirable components having a linear structure. It is defined as the Euclidean distance between the point and the optimal 3D-line among its neighbors, weighted by its number of neighbors.

The unary data term E_d is then expressed as

$$E_{di}(x_i) = \begin{cases} (1-f_e).f_p.f_s & \text{if } x_i = \textit{building} \\ (1-f_e).(1-f_p).(1-f_s) & \text{if } x_i = \textit{vegetation} \\ f_e.f_p.f_s & \text{if } x_i = \textit{ground} \\ (1-f_p).f_s.f_c & \text{if } x_i = \textit{clutter} \end{cases} \tag{21}$$

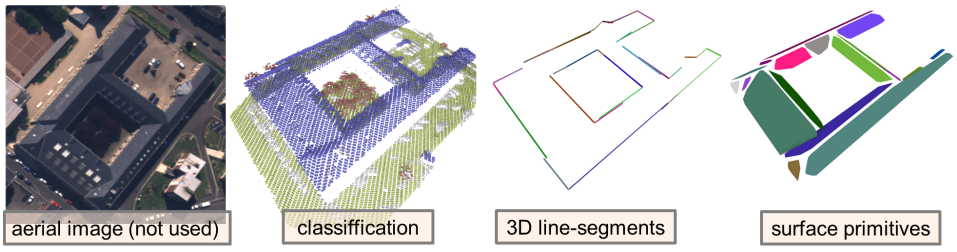


Figure 26: Classification and primitive extraction from Lidar scans. Point clouds are classified into four classes [color code: blue=*building*, red=*vegetation*, yellow= *ground* and white= *clutter*]. Both 3D line-segments and surface primitives are then extracted from the set of points classified as *building*. The main regular roof sections of the buildings are detected as well as the global building contours. Note that the planes are visually represented by their convex envelopes.

Planimetric arrangement. The idea consists in arranging both the geometric primitives and the other urban components identified during classification in a common dense representation. The proposed solution relies on a label propagation in a grid of X and Y axis under geometric constraints. Performing the arrangement on such a grid, called a planimetric map, allows us to substantially reduce the algorithmic complexity by assuming a 2.5D representation of urban scenes, and also to combine two distinct types of geometric tools, *ie* primitives and mesh patches, in a common framework.

Each point of the cloud is associated with the label *ground*, *vegetation*, *clutter*, $plane^{(k)}$, $cylinder^{(l)}$, $sphere^{(m)}$, $cone^{(n)}$ or $roof$. *Clutter* points are not taken into account in the following. The label $roof$ corresponds to the points classified as *'building'*, which have not been fitted to geometric primitives. The point labels are projected on a 2D-grid G. We denote by $G^{(proj)}$, the subset of G composed of the cells on which at least one point label has been projected, and $G^{(empty)}$ its complementary subset on G. The labels are then propagated on the entire grid G under structure arrangement constraints.

The label propagation procedure is performed using a Markov Random Field (MRF) with pairwise interactions, whose sites are specified by the cells of the 2D-grid G, and whose adjacency set E is given by a breakline-dependent neighborhood. $l = (l_i)_{i \in G} \in L$ represents a configuration of labels of the MRF, where L is the configuration space:

$$L = \{\text{ground}, \text{vegetation}, \text{plane}^{(l)}, \text{cylinder}^{(m)}, \text{sphere}^{(n)}, \text{cone}^{(o)}, \text{roof}\}^{card(G)} \tag{22}$$

The quality of a configuration l is measured by the energy U of the standard form:

$$U(l) = \sum_{i \in G} D_i(l_i) + \beta \sum_{\{i,j\} \in E} V_{ij}(l_i, l_j) \tag{23}$$

where D_i and V_{ij} constitute the data term and propagation constraints respectively, balanced by the parameter $\beta > 0$.

The neighborhood relationship is not defined by an isotropic area, but takes into account the extracted 3D line-segments extracted in order to stop the propagation beyond building contours. It is given by:

$$\{i,j\} \in E \Leftrightarrow \begin{cases} \|i - j\|_2 \leq r \\ \mathcal{O}(i, \mathcal{L}_k) = \mathcal{O}(j, \mathcal{L}_k) \end{cases} \tag{24}$$

where $\mathcal{L}_k$ is the 2D-line obtained by projecting the k^{th} 3D-segment interacting with the pair $\{i,j\}$. $\mathcal{O}(i, \mathcal{L})$ is the oriented side in which the cell i is located with respect to the line $\mathcal{L}$, and r is the maximal distance between two neighboring cells.

D_i checks the coherence of the label l_i at the cell i with respect to the input point cloud. The term is given by

$$D_i(l_i) = \begin{cases} c & \text{if } l_i = roof \\ \min(1, |z_{l_i} - z_{p_i}|) & \text{else if } i \in G^{(proj)} \\ 0 & \text{otherwise} \end{cases} \quad (25)$$

where $c \in [0, 1]$ is a coefficient penalizing the labels $roof$ in order to favor the primitive-based description of buildings. z_{l_i} is the height associated with l_i, and z_{p_i} the maximal height of the input 3D-points contained in the cell i.

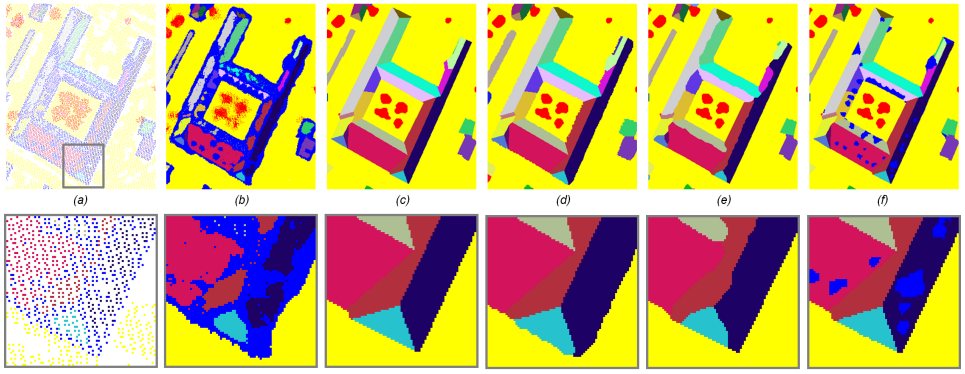


Figure 27: Planimetric arrangement. *(a)* the grid $G^{(proj)}$ of the projected point labels, *(b)* the initial label map, *(c)* the label map after minimizing U, *(d)* the label map after minimizing a variant of U where the breakline-dependent neighborhood is substituted by a standard isotropic neighborhood, *(e)* the label map after minimizing a variant of U where the $\bowtie$-law is not taken into account, and *(f)* the label map after minimizing U whose parameter c has been significantly decrease.. One can notice that the label propagation is correctly stopped beyond building contours and neighboring primitives. The $\bowtie$-law allows the optimal arrangement of the roof sections, and the breakline-dependent neighborhood avoids the wavy building contours [color code: white=empty cell, yellow=ground, red=vegetation, blue=roof, other colors=primitives].

V_{ij} allows both the label smoothness and a coherent arrangement of the primitives. To do so, an arrangement law, denoted by $\bowtie$, is introduced to test whether two labels, l_i and l_j, of neighboring cells, i and j, are spatially coherent:

$$l_i \bowtie l_j \Leftrightarrow \mathcal{O}(i, \mathcal{I}_{l_i,l_j}) \neq \mathcal{O}(j, \mathcal{I}_{l_i,l_j}) \quad (26)$$

where $\mathcal{I}_{l_i,l_j}$ is the XY-intersection between the two objects l_i and l_j, and $\mathcal{O}(i, \mathcal{I})$ is the oriented side in which the cell i is located with respect to the

curve $\mathcal{I}$. In other words, the intersection of the two objects must be spatially located in between the two cells i and j. Finally the pairwise interaction is formulated by:

$$V_{ij}(l_i, l_j) = \begin{cases} \epsilon_1 & \text{if } l_i \bowtie l_j \\ \epsilon_2 & \text{if } l_i = l_j \\ 1 & \text{otherwise} \end{cases} \tag{27}$$

where ϵ_1 and ϵ_2 are real values in $[0, 1]$ with $\epsilon_1 < \epsilon_2$. They tune the label smoothness with respect to the coherent object arrangement considerations.

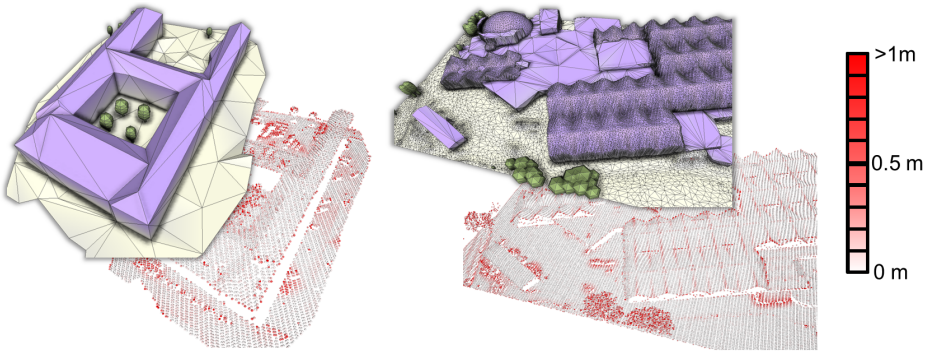


Figure 28: Building reconstruction - (*top*) obtained 3D-model and (*bottom*) input cloud ($2\,\text{pts/m}^2$) with the points colored according to their distance to the 3D-model. The high errors correspond to points from trees (the points of a tree do not obviously describe a perfect ellipsoidal shape) and from small urban components such as cars or roof superstructures. The left building is highly regular, the roof being entirely explained by planar primitives. The right building is more complex as it contains atypical roof forms, *ie* the undulating roof sections and the spherical dome. An hybrid surface is particularly relevant in such a case.

In order to get reasonable running times, a parallelization scheme is proposed on the entire scene, relying on the two following assumptions: (i) the labels cannot be propagated between two non-overlapped urban objects in the scene (*eg* the label corresponding to the roof section of a building cannot be used for an other building), (ii) the point labels originally projected in the 2D-grid G are of quality, *ie* they are probably correct. Thus, the configuration space can be significantly reduced by considering the minimization of U as a set of N local independent (and thus parallelizable) energy minimization problems over a partition of the grid G. The α-expansion algorithm [BVZ01] is used to solve each local independent optimization problem. The generation of 3D models from label maps relies then on standard projection operations detailed in [LM12].

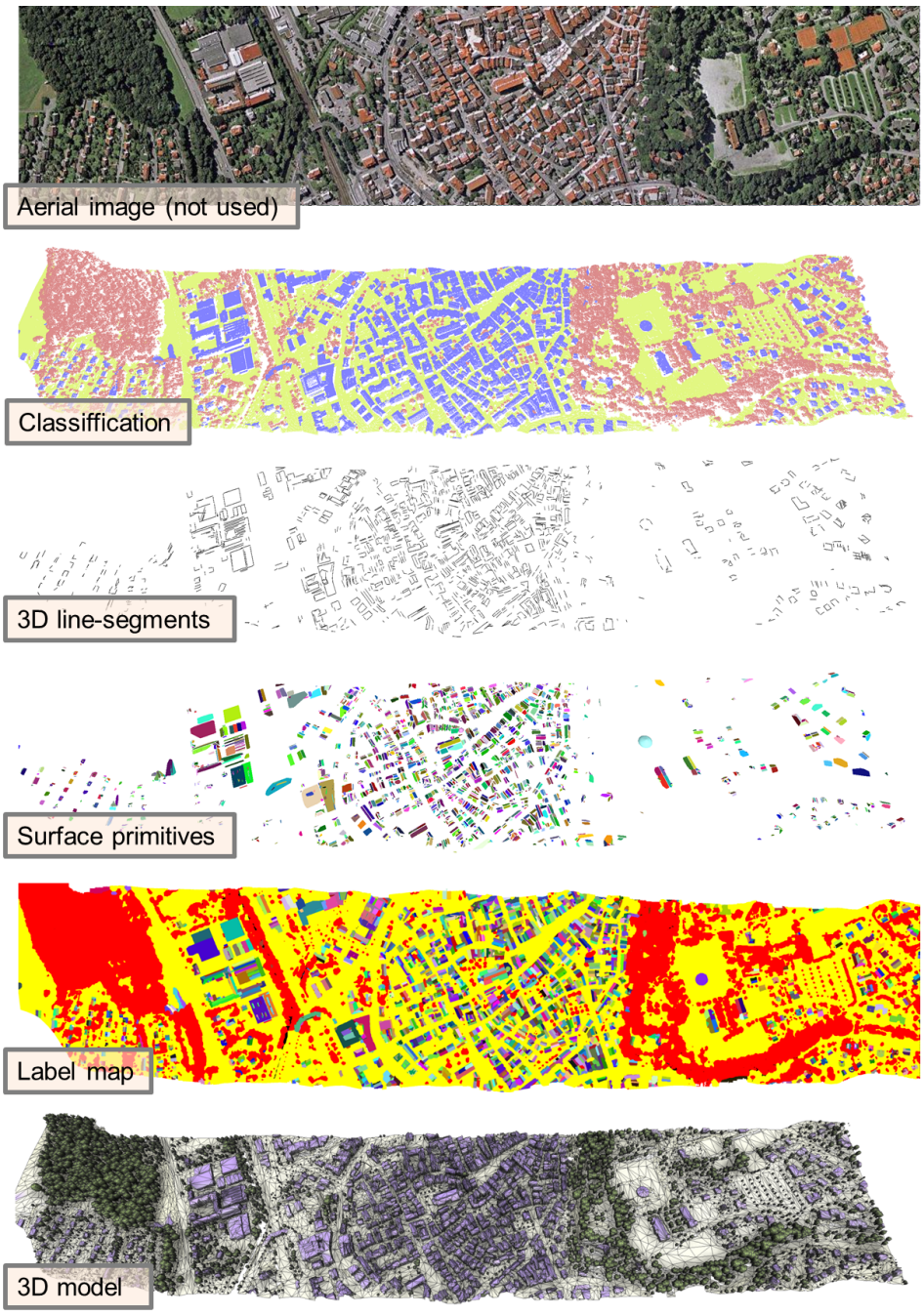
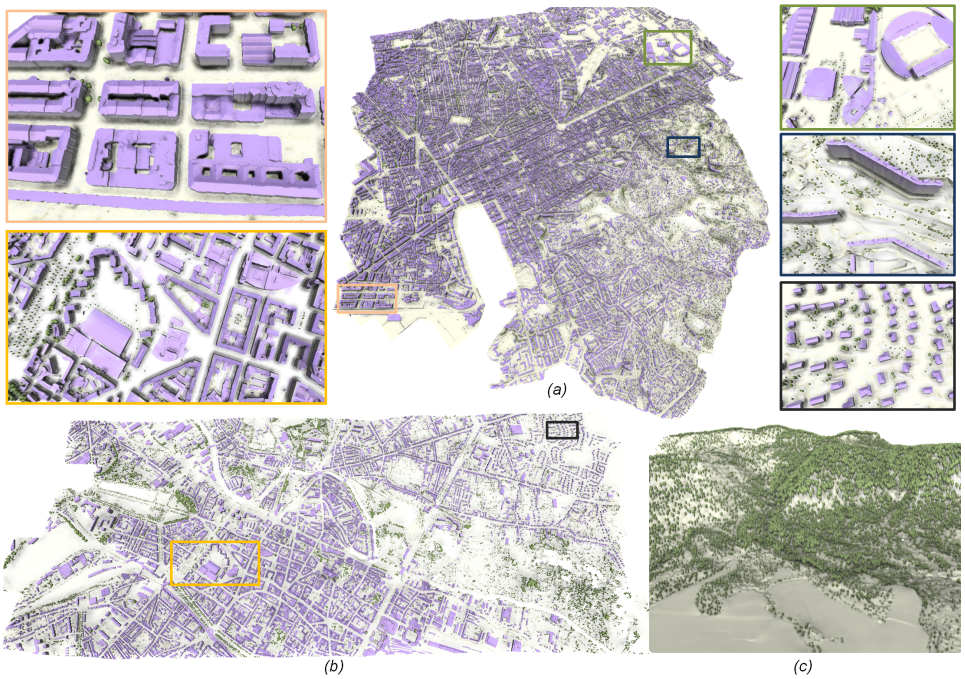


Figure 29: 1 km^2 area urban modeling from a 2.3M Lidar scan (Biberach, Germany). The intermediate results of the algorithm are shown from top to bottom. The final result (3D model) is obtained in approximatively 10 minutes.

Results. The algorithm has been tested on various types of urban landscapes including business districts with large and tall buildings, historic towns with a high concentration of both small buildings and trees, and hilly areas with high altimetric variations and dense forests (see Fig. 30).



	#input points ($\times 10^6$)	area (km^2)	running time (hour)	compaction (Mo)
Marseille, France *(a)*	38.67	19.8	2.52	131
Amiens, France *(b)*	24.52	11.57	1.34	93
Mountain area *(c)*	22.67	3.41	0.31	34

Figure 30: Reconstruction of three large scenes with some performance statistics and crops on various types of urban landscapes.

One of the main advantages of this hybrid representation for city modeling is that the potential primitive under-detection does not affects the surface accuracy. Indeed the regular roof sections missed during the geometric primitive extraction stage are completed by mesh-patches. The final 3D-model remains coherent and correct even if it loses in terms of compaction. The eventual under-detection of 3D-segments is more penalizing, especially when the input cloud has both a spatially heterogeneous point distribution and a low point density. In such a case, 3D-models can have wavy contours which correspond to the shape induced by the bordering points of the building as shown in Fig. 28. One solution can be then to simplify the mesh but this engenders a loss of accuracy. On the other hand, over-detecting primitives

would increase the number of labels during the planimetric arrangement, and thus, the running times as well as the model complexity.

Discussion. Some urban components are not taken into account in our representation. In particular, the bridges and the elevated roads which are local planar structures elevated above the ground are frequently detected as buildings (see Fig. 30, top right crop). This problem can be solved by considering additional urban components in the point cloud classification. Note that in this perspective, the energy formulation of the planimetric arrangement can be easily adapted. This algorithm is also not optimal when both the altimetric accuracy of the input points is poor and the point density is weak, typically with low resolution Digital Surface Models, *ie* >0.5 m. In such cases, the use of less generic methods constrained by high order geometric assumptions is recommended to compensate for the low quality of data, *eg* the structural-based approach of [LDZPD10].

4.3 Airborne imagery

The second algorithm starts from raw meshes generated from airborne imagery. With the recent advances in Multi-View Stereo (MVS) [SCD+06], some efficient techniques have been developed for creating dense meshes from high resolution images. Contrary to depth maps and Lidar scans, these triangular meshes, that we call *MVS meshes*, constitute real-3D representations in the sense that information is also retrieved on the vertical components of the scene. As illustrated on Fig. 31, MVS meshes are particularly accurate and provide an impressive amount of detail, outclassing existing 3D models for visualization-based applications.

MVS meshes are, however, not optimal in certain application domains such as urban planning, wireless propagation simulation or navigation for which 3D-models must either be compact or contain semantical information allowing the identification of urban objects within the scene. In addition, MSV meshes contain geometric and topological defects; the main problems coming from the presence of holes and self-intersecting facets, spatial heterogeneous distributions of vertices and facets, the loss of accuracy in presence of reflecting surfaces as glass, and the merging of different urban components such as trees with facades.

Object identification. Four classes are considered: *ground*, *tree*, *facade* and *roof*. The classification relies on simple but efficient geometric assumptions: *(i) ground* is characterized by locally flat surfaces located below the other classes, *(ii) tree* has irregular curved surfaces, *(iii) facade* are vertical structures connecting *roof* and *ground* and *(iv) roof* are mainly composed of piecewise-planar surfaces.

Three different geometric attributes are computed from each facet of the mesh for distinguishing the different classes of interest.

- *Elevation* a_e is defined as a function of the relative height (z coordinate) of the facet centroid, similarly to the Lidar case.
- *Planarity* a_p denotes the planarity of the superfacet containing f_i, derived from the surface variation.
- *Horizontality* a_h measures the deviation of the unit normal $\mathbf{n}_i$ to facet f_i with respect to the vertical axis.

As MVS meshes are extremely dense, classifying each triangular facet would lead to both high running time and poor spatial regularization. Instead, groups of connected facets - that we call *superfacets* - are considered. Superfacets are obtained by clustering the facets with similar mean curvatures [BKP+10]. A region growing is used to efficiently regroup facets; the propagation is relatively fast as the facet adjacency is known. This clustering procedure preserves the planar components. From these geometric attributes defined for each facet, all taking values within $[0, 1]$, we compute the geometric attribute for each superfacet as the area-weighted sum of the geometric attributes of its facets.

The energy formulated in Eq. 20 is specified by S, the set of superfacets, and E the set of adjacent superfacets, two superfacets being adjacent if they share at least one edge in the input mesh. The data term combines the above-described attributes weighted by the area A_i of the superfacet i:

$$D_i(l_i) = A_i \times \begin{cases} 1 - a_p \cdot a_h \cdot \overline{a_e} & \text{if } l_i = \textit{ground} \\ 1 - \overline{a_p} \cdot a_h & \text{if } l_i = \textit{tree} \\ 1 - a_p \cdot \overline{a_h} & \text{if } l_i = \textit{facade} \\ 1 - a_p \cdot a_h \cdot a_e & \text{if } l_i = \textit{roof} \end{cases} \tag{28}$$

where $\overline{a_\cdot} = 1 - a_\cdot$. The pairwise interaction V_{ij} between two adjacent superfacets i and j favors label smoothness away from sharp creases:

$$V_{ij}(l_i, l_j) = C_{ij} \cdot w_{ij} \cdot 1_{\{l_i \neq l_j\}}, \tag{29}$$

where $1_{\{.\}}$ denotes the characteristic function, and C_{ij} denotes the length of the interface between superfacets i and j (sum of interface edge lengths). Weight w_{ij} is introduced to lower the label propagation over sharp creases that often appear when two classes meet, *eg* for trees adjacent to facades. w_{ij} is defined as the angle cosine between the estimated normals of two superfacets. As the unary data term and pairwise potential are weighted by the superfacet areas and interface lengths, this energy formulation behaves similarly to a facet-based energy with grouping constraints.

The aforementioned geometric rationale alone is not sufficient to solve the ill-posed classification problem. Two types of errors frequently occur

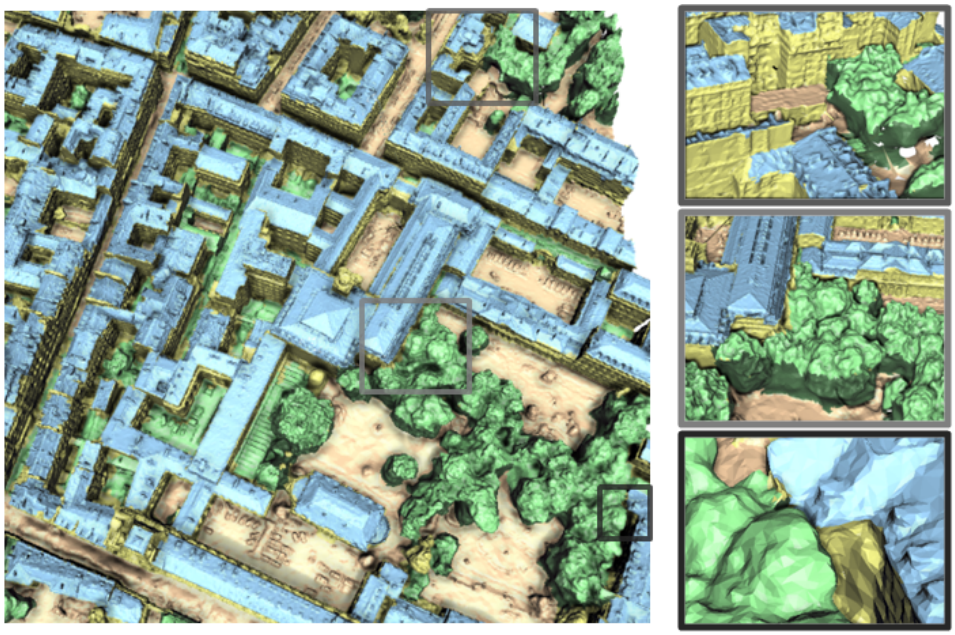


Figure 31: Classification into four classes of urban elements: roof (blue), facade (yellow), ground (brown) and trees (green). The regularizing term of the energy as well as the semantic rules improve spatial consistency. The close-ups depict how roofs and facades, as well as trees adjacent to facades, are adequately separated.

when dealing with complex urban scenes: *(i)* roof superstructures such as chimneys or dormer-widows may be wrongly labeled as *tree*, these elements being too small and irregular to be considered locally planar, and *(ii)* vertical components of large trees may be labeled as *facade*. We thus add the following semantic rules:

- *Rule 1.* superfacets labeled as *tree* and adjacent to only superfacets labeled as *roof* are re-labeled *roof*. This rule relies on the common assumption that large trees are not located on top of roofs.
- *Rule 2.* superfacets labeled as *facade* and adjacent to superfacets labeled as *tree* and *ground* are turned to *tree*.

As illustrated by Fig.31, these two rules bring greater contextual coherence to the semantic labeling in presence of small irregular roof superstructures and trees with cylindrical shapes. Finally, after classification we decompose the scene into connected components: isolated buildings or blocks of connected buildings are extracted by searching for connected components of superfacets labeled as $roof$ and $facade$.

Modeling at various LOD. Most of city models are generated at a given urban scale in terms of geometry and structure, making them suitable to a restricted application domain only. The idea here is to generate multiple variants of city models controllable through an urban Level Of Detail (LOD) formalism. General mesh simplification or approximation approaches are effective but often merge objects of different classes, *eg* a tree and a roof, and fragment structural features such as the boundary of a roof. Most of these approaches rely on a pure geometric error metric and are thus oblivious to semantic and structural considerations for urban scenes. Some error metrics are more feature-preserving than others, which indirectly helps preserve the structure, but the structure itself is scale-dependent and hence can hardly be decoupled from semantic labels specific to urban LODs.

The LOD generation proceeds by filtering the planar primitives and abstracting the icons, in accordance to the urban LODs used by CityGML:

- LOD0: ground mesh is not used as the representation is planar. Trees are depicted as discs computed as vertical projection of tree icons, and buildings are depicted by 2D regions bounded by polylines computed only from the primitives labeled as *facade* using a 2D instance of the min-cut formulation. Superstructures are omitted.

- LOD1: ground mesh, enriched with vertical cylinders for trees and a LOD0-building elevated in 3D with horizontal primitives as roofs whose height is computed by median of corresponding superfacet height.

- LOD2: ground mesh, enriched with tree icons, and building reconstructed with all primitives to generate piecewise-planar roofs.

- LOD3: LOD2 enriched with roof superstructure. We do not attempt at reconstructing details on facades such as doors and windows as we are dealing with airborne measurement data with insufficient resolution on vertical structures.

3D arrangement with discrete partitioning. The final reconstruction step turns the planar primitives filtered by the choice of LOD into watertight buildings. We instantiate a series of 3D arrangements, and label the cells of these arrangements as inside or outside via a min-cut formulation

Even when restricting it to each building component, computing the complete, exact arrangement leads to very high computational complexity (we experimented with scenes containing hundreds of components, each containing on average hundred planes). Previous work based on arrangements attempted to reduce such complexity by restricting to axis-aligned planes [FCSS09] or by computing a two-level hierarchy made up of a rectilinear volumetric grid combined with a convex polyhedral cell decomposition [CLP10]. The former work is too restrictive and the latter approach exceeds half an

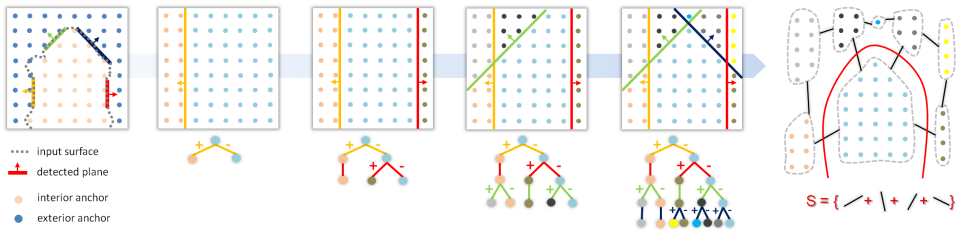


Figure 32: Discrete space partitioning. Anchors are labeled as interior or exterior to the input mesh by ray casting (left). At each plane insertion, both the anchors of the discrete space (top) and the BSP tree (bottom) are updated. After the last plane insertion, the anchor set is decomposed into discrete cells (right, colored points) from which one can compute a discrete volume or the ratio of interior/exterior anchors, as well as identify the adjacent cells. Once the optimal cut of our discrete problem is found, the surface can be extracted by computing the exact geometry of facets from the BSP tree (black edges crossed by the red cut).

hour when dealing with more than few hundred planes. In [OLA14], the 3D problem is turned to multiple 2D problems by slicing the scene under verticality assumptions valid only for indoor context. Observing that only a very small subset of the faces of the arrangement contribute to the output after solving for a min-cut surface, we postpone the exact geometric computation operations to the final surface extraction step after min-cut solve. We rely instead on a transient discrete approximation of the arrangement so as to avoid the compute-intensive exact geometric operations required to insert each plane into the arrangement.

For each sub-part of the input MVS corresponding to a building component we first compute an object-oriented bounding box B. We then sample uniformly B by placing sample points at the corners of a uniform grid aligned to B. Each of these sample points, referred to as *anchors*, is associated to (i) a Boolean localization flag specifying whether the anchor is estimated to be inside or outside the inferred building, and (ii) an integer index denoting the cell of the arrangement containing the anchor. We iterate over each anchor and guess its inside/outside flag with respect to the inferred building by casting rays and counting the intersection parity of these rays against the input MVS mesh using an AABB tree data structure. Five rays have shown sufficient in all experiments: four towards the upper corners of B and one towards the barycenter of these corners. We then compute an approximate arrangement of planes by inserting iteratively all planes of B, then the planar proxies, while refining an arrangement tree. Instead of computing the exact geometry of the arrangement cells for each plane insertion we update

the anchor cell indices, see Fig.32. The anchors are used as unit volume elements to approximate geometric information of cells (volume, facet areas and adjacency).

For each arrangement a min-cut formulation is used to find an inside/outside labeling of the cells, the output surface being defined as the interface facets between inside and outside. Consider a graph $(\mathcal{C}, \mathcal{F})$ where $\mathcal{C} = \{c_1, \ldots, c_n\}$ denotes the nodes relating to the cells induced by the space partition, and $\mathcal{F} = \{f_1, \ldots, f_m\}$ denotes edges relating to the facets separating all pairs of adjacent cells. A cut in the graph consists of separating the cells $\mathcal{C}$ into two disjoint sets $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$. The edges between $\mathcal{C}_{in}$ and $\mathcal{C}_{out}$ correspond to a set of facets forming a surface $\mathcal{S} \subset \mathcal{F}$.

In order to quantize the quality of the solution, *ie* the surface $\mathcal{S}$ induced by the cut $(\mathcal{C}_{in}, \mathcal{C}_{out})$, we introduce the following cost function C:

$$C(\mathcal{S}) = \sum_{c_k \in \mathcal{C}_{out}} V_{c_k}\, g(c_k) + \sum_{c_k \in \mathcal{C}_{in}} V_{c_k}\, (1 - g(c_k)) + \beta \sum_{f_i \in \mathcal{S}} A_{f_i}, \qquad (30)$$

where V_{c_k} denotes the volume of cell c_k, $g(c_k)$ denotes the function estimating the label likelihood of cell c_k with respect to the ratio of its inside/outside anchors, and A_{f_i} denotes the discrete area of facet f_i. The first two terms of the cost function C are data terms whereas the third term weighted by parameter $\beta \geq 0$ acts as a regularization term in order to favor solutions with small area. The optimal cut minimizing the cost $C(\mathcal{S})$ is found via the max-flow algorithm [BK04].

Function $g(c_k)$, defined in the interval $[0, 1]$, quantizes the coherence of assigning label *inside* to cell c_k with ratio r_{in} of inside anchors contained in c_k:

$$g(c_k) = \frac{(2r_{in} - 1) \times |2r_{in} - 1|^{\alpha} + 1}{2}, \qquad (31)$$

where α is a model parameter tuning the data sensitivity of function g. The optimal cut corresponds to a subset of facets separating the inside and outside cells, as depicted by Fig.32. The final geometry of these interface facets is then computed with exact arithmetic by intersecting the set of corresponding planes. By construction each interface facet is thus a planar convex polygon. For LOD0 and LOD1 we create a 2D instance of such discrete arrangement and min-cut formulation by sampling a single horizontal layer of anchors.

Results. This pipeline digests input meshes with several million triangle facets. On average a block of buildings is fully processed in around 30 seconds for LOD1 and 3 minutes for LOD2. Fig.34 depicts a variable density urban scene covering 1km square of Paris with 235 building components, 3.3K roofs and 1.3K trees. For this complex model the total running time is less than 20 minutes for LOD1 and around 2 hours for LOD2 (175K facets), with a sequential implementation of the plane arrangement per block.

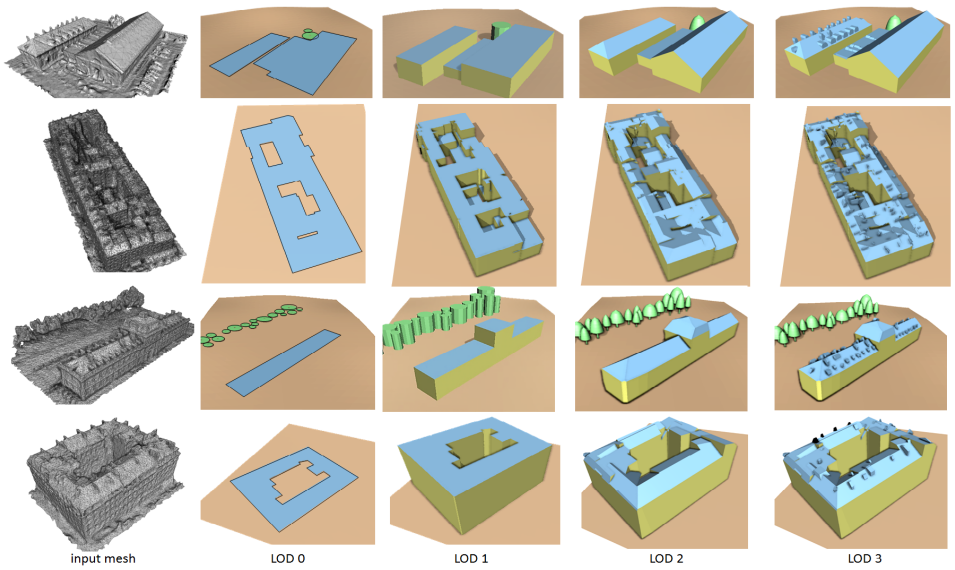


Figure 33: LOD generation of several buildings. First row: on this simple residential scene all facades and roofs are well classified and the Z-symmetry relationships between the two types of roof (2 and 4 slopes) enables abstraction. Second row: on this dense urban component each roof is simple but all roofs form a complex arrangement as the buildings have been built at different times with little coherence. Third row: on this architectural building both Z-symmetry and orthogonal relationships cooperate to abstract the central part of the roof. Fourth row: this building contains complex and thin roof superstructures. Despite a limited accuracy of the input MVS mesh our method recovers the main facades and roofs, and most superstructures.

Cases that challenge the robustness of our algorithms include input meshes with insufficient density and defects such as noise, holes and overlaps. Fig.33 (fourth row) shows that small scale roofs may not be reconstructed in LOD2 but are recovered in LOD3 as roof superstructures. Imperfect input data often lead to over- or under-detected planar proxies. We observe that over-detection is often compensated by the proxy regularization procedure that merges nearly-coplanar proxies. Under-detection however leads to very few primitives as observed on free-form architectural buildings, and hence to an overly abstracted reconstruction. Nevertheless, in the worst case where no primitives are detected for a building component, the output LOD is abstracted as its bounding box. Data that challenge the classification step include merged objects such as a tree touching a facade, and clutter elements such as cars or hedges digested by the four classes of interest. The

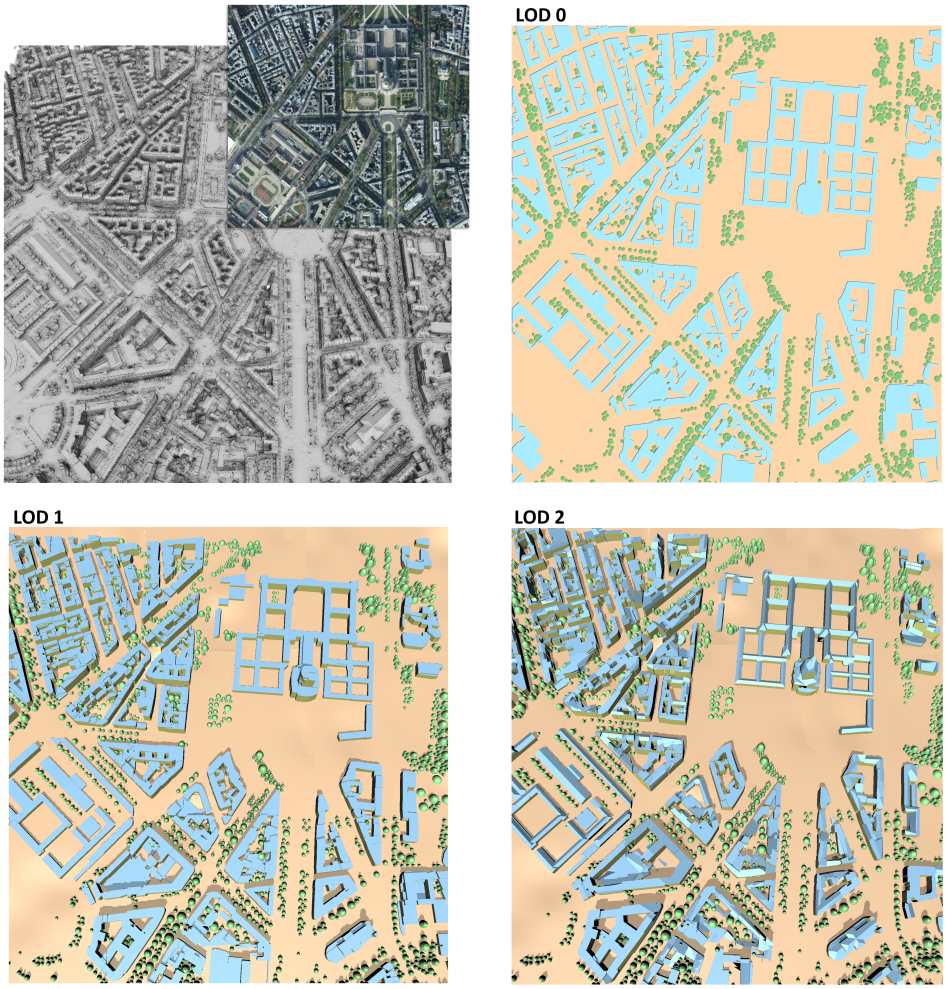


Figure 34: Reconstruction on large-scale urban scene. The input mesh (11M triangle facets) was generated from 600 airborne images. LOD1 and LOD2 comprise 10K and 175K polygon facets respectively, excluding tree and ground meshes.

regularization term of the energy together with the semantic rules improve spatial consistency and reduce the number of classification errors. Fig.35 evaluates the robustness of the primitive detection on an input mesh with variable scale features, noise and smoothed features.

Fig.36 evaluates the accuracy of the reconstructed LODs against the input meshes, albeit our approach is designed to provide a trade-off between faithfulness to input data and structure-aware abstraction. The comparisons against two mesh approximation approaches [GH97, CSAD04], referred to

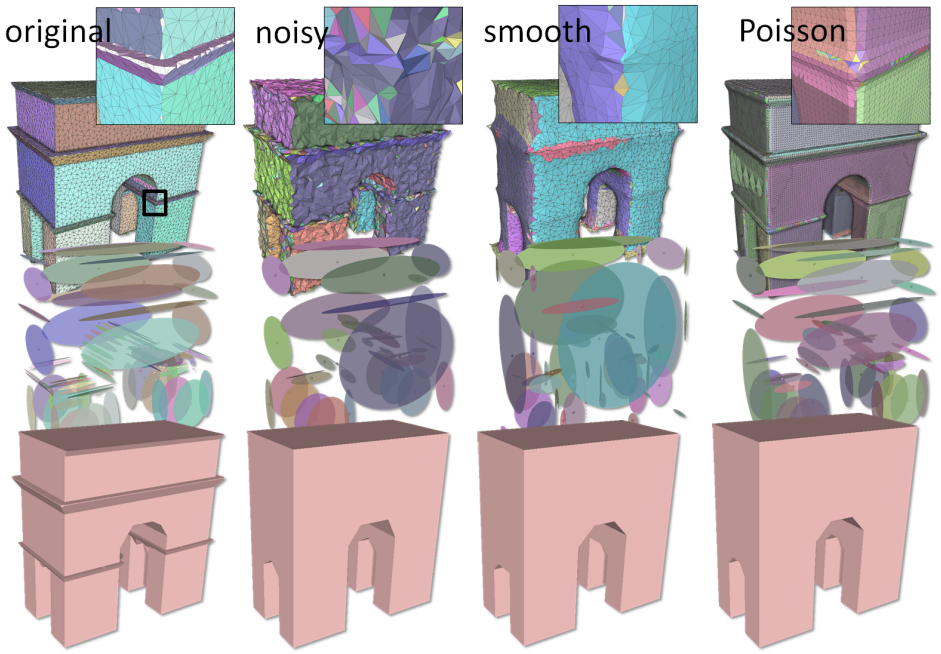


Figure 35: Robustness. Left: "defect-free" input mesh colored by superfacets (top) and its planar primitives (middle). Our reconstruction algorithm (applied here with no classification to evaluate only the primitive detection and abstraction steps) recovers most features (bottom). Notice the curved area reconstructed by planar polygons. Middle left: when noise is added the small scale features are filtered out and the vault is overly simplified. Right: when fed with the output of the Poisson reconstruction method the behavior of the algorithm is similar to the one on the smoothed mesh (middle right).

as QEM and VSA respectively, show comparable approximation errors, better resilience to holes and topological artifacts of the input mesh through the arrangement of planes, and better coherence and preservation of thin structures across LODs such as the square church towers. Notice how LOD3 represents roof details such as chimneys and dormer-windows while keeping a low polygon count.

Discussion We limited the classification to four common classes of urban objects. At first glance such a low class number may appear restrictive in terms of semantics, but these four classes match CityGML and the requirements of several application needs. They provide a satisfactory trade-off between robustness and quality of the reconstruction (we found only few errors during visual inspection of the large scale scene at LOD1 and LOD2 against the airborne tiled image, see Fig. 34). Similarly to the method pre-

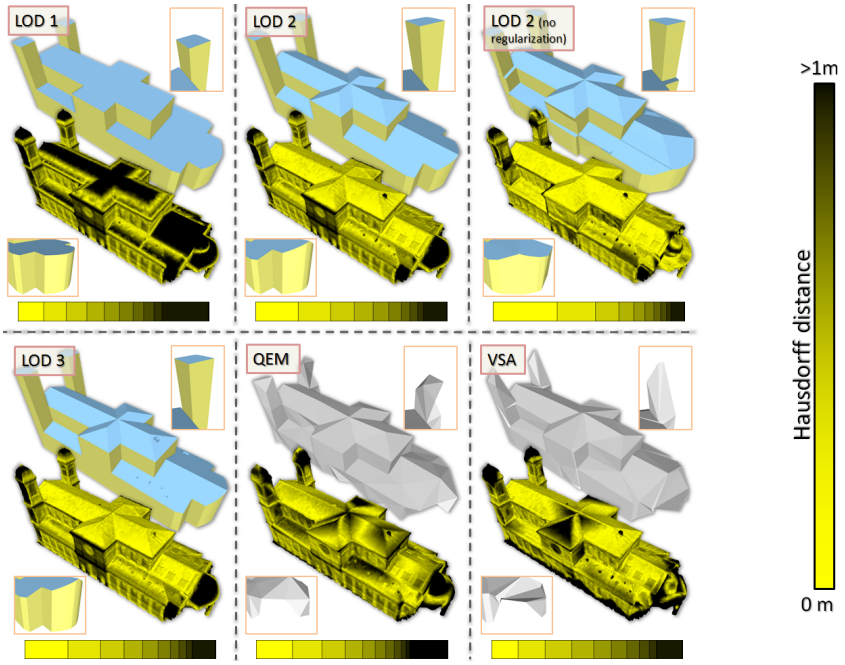


Figure 36: Geometric accuracy and structure-awareness. We compare the LODs to two mesh approximation algorithms by measuring the Hausdorff distance (color scale from yellow to black) to the input mesh. The complexity of the LOD2, QEM and VSA models is identical (190 facets). The root mean square error (RMS) of LOD1 is higher than for QEM and VSA (0.47 vs 0.4 and 0.43 respectively), the roof being poorly approximated. LOD2 without plane regularization has a lower RMS than LOD2 with planar regularization but is less abstracted and consumes less time to reconstruct (0.33s vs 0.39s). In terms of structure-awareness, thin components such as the church towers are correctly preserved in the different LODs, which is not the case for mesh approximation algorithms (see top close-ups). In addition, QEM and VSA do not fill the holes contained in the input mesh (see bottom close-ups).

sented in Section 4.2, this choice hampers the reconstruction of less common urban structures such as bridges or elevated roads. Our 3-step pipeline is however amenable to inserting additional classes with new labels in the MRF-based classification, and new objects to abstract and reconstruct. The use of planar proxies is also a limitation when dealing with free-form architecture buildings such as the dome of *Les Invalides* depicted by Fig.34.

5 Conclusion

5.1 Summary

In this habilitation thesis, a series of works in the field of geometric modeling of urban environments from physical measurements has been presented. We briefly summarize the contributions brought by these works before discussing their limitations.

Advances on Spatial point processes. Several contributions for improving flexibility and applicability of point processes have been presented. Concepts to extend point processes to the manipulation of libraries of geometric shapes [LGD10] and graphs [CFL13] have been formulated and applied to various vision problems. Two optimization techniques for sampling point processes from large-scale spaces have also been developed in order to improve performances and stability of state-of-the-art samplers [VL14]. The potential of these contributions has been tested on various applications, including population counting, parametric object extraction or shape recognition.

Hybrid surfaces. Beyond conventional surface representations used in vision and geometry processing, the concept of hybrid surfaces collects the advantages of both free-form and primitive worlds, *ie* high robustness, low model complexity and structure-awareness. This concept has been applied to surface generation problems in different contexts, in particular, surface reconstruction from unorganized point clouds [LA13] and from Multi-View Stereo images [LKBV13], as well as surface approximation [LKB10]. Experimental results suggest such surface representations constitutes an interesting solution in terms of accuracy and performance for modeling urban environments and man-made objects.

Large-scale city models. A general automatic pipeline for reconstructing large-scale cities from airborne data acquisitions, in particular from Lidar sans [LM12, VLZ11] and dense meshes obtained from Multi-View Stereo systems (Section 4.3), have been proposed. Objects contained in the observed scene are identified among three categories (buildings, trees and ground), and then modeled by specialized representations. In particular, we proposed original building modeling strategies that lie on geometric primitive arrangements: a planimetric arrangement scheme [LM12] and a discretized volumetric labeling that explores Levels Of Detail.

Limitations. The responses brought by these works to the geometric modeling of urban environments are, however, not fully-satisfactory. Several important restrictions remind us that many challenges still exist in the field.

The Automatic selection of object categories and geometric primitives is not addressed in this habitation thesis. Our algorithms rely on the assumption that the observed scene can be explained by a predefined library of categories or geometric primitives. If hybrid surfaces are, by definition, a robust solution to wrong geometric primitive selection, the object identification procedures cannot generate and select categories. For instance, the city modeling algorithm proposed in [LM12] fails to correctly describe bridges or elevated roads that are not considered as object categories.

Evaluation tools for measuring the geometric quality of models against Ground Truth are sill largely undeveloped compared to other vision and graphics application fields. The recent benchmark proposed in [RSJ+12] constitutes a preliminary step in this direction. The evaluation problem arises from the difficulties to (i) create accurate Ground Truth at the scale of a city, (ii) share non-public datasets, and (iii) propose relevant quantitative criteria that combined both geometric and semantic considerations. In our case, output models have been quantitatively evaluated from a restricted number of objects for which geometric and semantic information were available.

The specialization of urban modeling algorithms, that makes them very sensitive to input data, remains a big issue. In particular, algorithms designed for Multi-View Stereo imagery do not adapt well to Lidar scans, and *vice versa*. Even for a same type of data, algorithms are usually very sensitive to the quality of data resolution. To fight against this dependence, the key might be to go further in the use of geometric primitives. It seems quite crucial to create a new generic geometric language for modeling urban environments, in particular a geometric vocabulary common to the different data sources and a geometric grammar that provides structural rules to infer 3D models from descriptor vocabulary.

5.2 Perspectives

Looking into the future, important challenges are arising. In addition to the problems of evaluation tools and category selection discussed previously, we believe that five research directions will be of high interest in the coming years.

Urban scale-space exploration. The structure within urban environments is not a fixed entity, but it evolves depending on the scale at which the scene is analyzed. Beyond the detection of geometric primitives and the discovery of structural relationships, it seems crucial to design algorithms that provides control upon the structure. In particular, one of the main challenges is to explore the solution space across scales and automatically select the structure, for instance as the best trade-off between complexity

and faithfulness to input data, where complexity relates to the enumeration of structural rules and their parameters.

Physical coherence. To constrain the solution space of ill-posed urban modeling problems, one interesting research direction is to take into account physical principles in addition to geometric, semantic or structural considerations. Exploiting physical coherence is not only a means for reinforcing the method efficiency; it is also a goal for producing 3D models that conform to physical principles such as self-supporting masonry structures, or constraints related to manufacturing and 3D-printing. In free-form architecture modeling, some recent works explore this research direction, for instance for tiling a surface with specific manufacturing and machining constraints.

Functionality. Beyond modeling of objects and scenes as sets of structured objects, discovering their function is another important challenge. Form follows function is a common principle in design and architecture: the shape of an object should be primarily based upon its intended function. Geometry, structure, semantic and physical coherence contribute to characterizing the nature of objects, and can be further exploited to understand their utility and to specialize their computerized modeling. For urban modeling, one objective is to understand the function of a building by analyzing these different characteristics. Some preliminary works have been proposed at the scale of individual objects [GGVG11], and it is still a scientific challenge to extend some of these ideas to large scale scenes.

Community data. During the last decades, geometric modeling issues on urban environments have been largely tackled from specialized sensors as airborne/satellite stereoscopic imagery or Laser scanning. Today, this data acquisition paradigm is completely reassessed with the emergence of new acquisition procedures that allow non-specialized people to freely access and enrich big datasets. An increasing variety of sensors is progressively disseminated everywhere; the best illustration is probably the 1.5 billion smartphones interacting in the world. The emergence of such "community data" coupled with the expanding computational resources constitute a great opportunity to propose efficient solutions to two big recurrent limitations in urban modeling: the low coverage of the specialized data (only a hundred cities are digitalized in 3D in the world) and the lack of flexibility of existing methods that are designed from a specific type of data to produce standardized 3D models. This new paradigm leads to an entire rethinking of the existing algorithm designs towards more flexibility and different data specificities, going from specialized, rare, private, expensive and accurate to multi-sourced, massive, public, free and defect-laden measurements.

Dynamical urban environments. The fact urban environments permanently evolve in time provokes considerable efforts for detecting changes and updating models. Beyond change detection and city model updating, which are both traditional problems largely addressed in the literature, one major scientific challenge is to understand tendencies and even anticipate the evolution of urban environments in terms of geometry. Recent works in vision [KZBH12] have demonstrated that prediction functions can be efficiently designed to forecast human actions from image sequences. In urban modeling, such prediction functions could be created by analyzing geometric variations along time from a big flux of information, typically community data. Many urban indicators could be studied, going from the expansion/shrinking directions of cities to evolution of architectural style through to the road network complexity.

References

[ABK98] N. Amenta, M. Bern, and M. Kamvysselis. Crust: A new Voronoi-based surface reconstruction algorithm. In *SIGGRAPH*, 1998.

[AP10] M. Attene and G. Patane. Hierarchical structure recovery of point- sampled surfaces. *Computer Graphics Forum*, 29(6), 2010.

[ASF+12] M. Arikan, M. Schwarzler, S. Flory, M. Wimmer, and S. Maierhofer. O-snap: Optimization-based snapping for modeling architecture. *Trans. on Graphics*, 2012.

[ASS+09] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building Rome in a day. In *ICCV*, 2009.

[BBH03] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in Computational Stereo. *Trans. on Pattern Analysis and Machine Intelligence*, 25(8), 2003.

[BDZ12] C. Benedek, X. Descombes, and J. Zerubia. Building development monitoring in multitemporal remotely sensed image pairs with stochastic birth-death dynamics. *Trans. on Pattern Analysis and Machine Intelligence*, 34(1), 2012.

[Bis06] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[BJB10] J. Byrd, S. Jarvis, and A. Bhalerao. On the parallelisation of mcmc-based image processing. In *International Symposium on Parallel and Distributed Processing*, 2010.

[BK04] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Trans. on Pattern Analysis and Machine Intelligence*, 26(9), 2004.

[BKP+10] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon Mesh Processing*. AK Peters, 2010.

[BL93] A. J. Baddeley and M. Van Lieshout. Stochastic geometry models in high-level vision. *Journal of Applied Statistics*, 20(5-6), 1993.

[BLN+13] M. Berger, J. Levine, L. Nonato, G. Taubin, and C. Silva. A benchmark for surface reconstruction. *Trans. on Graphics*, 32(2), 2013.

[BVZ01] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Trans. on Pattern Analysis and Machine Intelligence*, 23(11), 2001.

[BZ00] C. Baillard and A. Zisserman. A plane-sweep strategy for the 3d reconstruction of buildings from multiple images. In *ISPRS Congress*, 2000.

[CC08] J. Chen and B. Chen. Architectural modeling from sparsely scanned range data. *International Journal of Computer Vision*, 78(2-3), 2008.

[CFL13] D. Chai, W. Forstner, and F. Lafarge. Recovering line-networks in images by junction-point processes. In *CVPR*, 2013.

[CLP10] A.-L. Chauve, P. Labatut, and J.-P. Pons. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In *CVPR*, 2010.

[CSAD04] D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. In *SIGGRAPH*, 2004.

[CY00] J. M. Coughlan and A. L. Yuille. The Manhattan world assumption: Regularities in scene statistics which enable Bayesian inference. In *NIPS*, 2000.

[Des11] X. Descombes. *Stochastic geometry for image analysis*. Wiley, 2011.

[DMZ09] X. Descombes, R. Minlos, and E. Zhizhina. Object extraction using a stochastic birth-and-death dynamics in continuum. *Journal of Mathematical Imaging and Vision*, 33(3), 2009.

[DWB06] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *IEEE Robotics and Automation Magazine*, 2, 2006.

[ED05] D. Earl and M. Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 23(7), 2005.

[Fau93] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, 1993.

[FCOS05] S. Fleishman, D. Cohen-Or, and C. Silva. Robust moving least-squares fitting with sharp features. In *SIGGRAPH*, 2005.

[FCSS09] Y. Furukawa, B. Curless, S.M. Seitz, and R. Szeliski. Manhattan-world stereo. In *CVPR*, 2009.

[FFGG+10] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building Rome on a cloudless day. In *ECCV*, 2010.

[FP10] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *Trans. on Pattern Analysis and Machine Intelligence*, 32(8), 2010.

[FSB06] F. Fraundorfer, K. Schindler, and H. Bischof. Piecewise planar scene reconstruction from sparse correspondences. *Image and Vision Computing*, 24(4), 2006.

[FZ03] C. Frueh and A. Zakhor. Constructing 3D City Models by Merging Ground-Based and Airborne Views. In *CVPR*, 2003.

[GGVG11] H. Grabner, J. Gall, and L. Van Gool. What makes a chair a chair? In *CVPR*, 2011.

[GH86] S. Geman and C.R. Huang. Diffusion for global optimization. *SIAM Journal on Control and Optimization*, 24(5), 1986.

[GH97] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH*, 1997.

[GKF09] A. Golovinskiy, V.G. Kim, and T. Funkhouser. Shape-based recognition of 3D point clouds in urban environments. In *ICCV*, 2009.

[GLGG11] J. Gonzalez, Y. Low, A. Gretton, and C. Guestrin. Parallel Gibbs sampling: From colored fields to thin junction trees. *Journal of Machine Learning Research*, 2011.

[GM94a] C. J. Geyer and J. Moller. Simulation and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics*, Series B(21), 1994.

[GM94b] U. Grenander and M.I. Miller. Representations of Knowledge in Complex Systems. *Journal of the Royal Statistical Society*, 56(4), 1994.

[GP12] G. Groger and L. Plumer. CityGML - interoperable semantic 3d city models. *Journal of Photogrammetry and Remote Sensing*, 71, 2012.

[Gre95] P.J. Green. Reversible Jump Markov Chains Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4), 1995.

[GSH+07] R. Gal, A. Shamir, T. Hassner, M. Pauly, and D. Cohen-Or. Surface reconstruction using local shape priors. In *SGP*, 2007.

[GZW03] C. E. Guo, S.C. Zhu, and Y.N. Wu. Modeling visual patterns by integrating descriptive and generative models. *International Journal of Computer Vision*, 53(1), 2003.

[HG00] M. Harkness and P. Green. Parallel chains, delayed rejection and reversible jump mcmc for object recognition. In *BMVC*, 2000.

[HK06] A. Hornung and L. Kobbelt. Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information. In *SGP*, 2006.

[HK10] N. Haala and M. Kada. An update on automatic 3d building reconstruction. *Journal of Photogrammetry and Remote Sensing*, 65(6), 2010.

[HK12] M. Habbecke and L. Kobbelt. Linear analysis of nonlinear constraints for interactive geometric modeling. In *Eurographics*, 2012.

[HTZ04] F. Han, Z.W. Tu, and S.C. Zhu. Range Image Segmentation by an Effective Jump-Diffusion Method. *Trans. on Pattern Analysis and Machine Intelligence*, 26(9), 2004.

[HZ04] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[HZC+13] Christian Hane, Christopher Zach, Andrea Cohen, Roland Angst, and Marc Pollefeys. Joint 3D scene reconstruction and class segmentation. In *CVPR*, 2013.

[IKH+11] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *ACM Symposium on User Interface Software and Technology*, 2011.

[JWS08] P. Jenke, M. Wand, and W. Straber. Patch-graph reconstruction for piecewise smooth surfaces. In *Proc. of the Vision, Modeling, and Visualization Conference*, 2008.

[KBH06] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *SGP*, 2006.

[KCVS98] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH*, 1998.

[KF98] R. Keriven and O. Faugeras. Complete dense stereovision using level set methods. In *ECCV*, 1998.

[KZBH12] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert. Activity forecasting. In *ECCV*, 2012.

[LA13] F. Lafarge and P. Alliez. Surface reconstruction through point set structuring. In *Eurographics*, 2013.

[LB07] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *CVPR*, 2007.

[LDZ05] C. Lacoste, X. Descombe, and J. Zerubia. Point processes for unsupervised line network extraction in remote sensing. *Trans. on Pattern Analysis and Machine Intelligence*, 27(10), 2005.

[LDZPD06] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny. An automatic building reconstruction method: a structural approach using hr images. In *ICIP*, 2006.

[LDZPD08] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny. Automatic building extraction from DEMs using an object approach and application to the 3d-city modeling. *Journal of Photogrammetry and Remote Sensing*, 63(3), 2008.

[LDZPD10] F. Lafarge, X. Descombes, J. Zerubia, and M. Pierrot-Deseilligny. Structural approach for building reconstruction from a single DSM. *Trans. on Pattern Analysis and Machine Intelligence*, 32(1), 2010.

[LGD10] F. Lafarge, G. Gimel'farb, and X. Descombes. Geometric feature extraction by a multi-marked point process. *Trans. on Pattern Analysis and Machine Intelligence*, 32(9), 2010.

[LGZ+13] H. Lin, J. Gao, Y. Zhou, G. Lu, M. Ye, C. Zhang, L. Liu, and R. Yang. Semantic decomposition and reconstruction of residential scenes from lidar data. In *SIGGRAPH*, 2013.

[Li01] S.Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer, 2001.

[LIP+10] F. Leberl, A. Irschara, T. Pock, P. Meixner, M. Gruber, S. Scholz, and A. Wiechert. Point clouds: Lidar versus 3d vision. *Photogrammetric Engineering and Remote Sensing*, 76(10), 2010.

[LKB10] F. Lafarge, R. Keriven, and M. Bredif. Insertion of 3D-primitives in mesh-based representations: Towards compact models preserving the details. *Trans. on Image Processing*, 19(7), 2010.

[LKBV13] F. Lafarge, R. Keriven, M. Bredif, and H. H. Vu. A hybrid multi-view stereo algorithm for modeling urban scenes. *Trans. on Pattern Analysis and Machine Intelligence*, 35(1), 2013.

[LM12] F. Lafarge and C. Mallet. Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation. *International Journal of Computer Vision*, 99(1), 2012.

[LPK09] P. Labatut, J.-P. Pons, and R. Keriven. Robust and efficient surface reconstruction from range data. *Computer Graphics Forum*, 28(8), 2009.

[LWC+11] Y. Li, X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. J. Mitra. Globfit: Consistently fitting primitives by discovering global relations. In *SIGGRAPH*, 2011.

[LZ10] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *NIPS*, 2010.

[LZS+11] Y. Li, Q. Zheng, A. Sharf, D. Cohen-Or, B. Chen, and N. Mitra. 2d-3d fusion for layer decomposition of urban facades. In *ICCV*, 2011.

[May08] H. Mayer. Object extraction in photogrammetric computer vision. *Journal of Photogrammetry and Remote Sensing*, 63(2), 2008.

[MBH12] D. Munoz, J. Bagnell, and M. Hebert. Co-inference machines for multi-modal scene analysis. In *ECCV*, 2012.

[MLM01] D. Marshall, G. Lukacs, and R. Martin. Robust segmentation of primitives from range data in the presence of geometric degeneracy. *Trans. on Pattern Analysis and Machine Intelligence*, 23(3), 2001.

[MPWC12] N. Mitra, M. Pauly, M. Wand, and D. Ceylan. Symmetry in 3d geometry: Extraction and applications. In *EUROGRAPHICS State of the Art Reports*, 2012.

[MSS+08] B. Matei, H. Sawhney, S. Samarasekera, J. Kim, and R. Kumar. Building segmentation for densely built urban regions using aerial lidar data. In *CVPR*, 2008.

[MWA+12] P. Musialski, P. Wonka, D. Aliaga, M. Wimmer, L. Van Gool, and W. Purgathofer. A survey of urban reconstruction. In *EUROGRAPHICS State of the Art Reports*, 2012.

[MWZ+13] N. Mitra, M. Wand, H. Zhang, D. Cohen-Or, and M. Bokeloh. Structure-aware shape processing. In *EUROGRAPHICS State of the Art Reports*, 2013.

[ODZ08] M. Ortner, X. Descombes, and J. Zerubia. A marked point process of rectangles and segments for automatic analysis of digital elevation models. *Trans. on Pattern Analysis and Machine Intelligence*, 30(1), 2008.

[OLA14] S. Oesau, F. Lafarge, and P. Alliez. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS Journal of Photogrammetry and Remote Sensing*, 90, 2014.

[PKF07] J.-P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2), 2007.

[PNF+08] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78(2-3), 2008.

[PTJYS12] T. T. Pham, Chin T.-J., J. Yu, and D. Suter. The random cluster model for robust geometric fitting. In *CVPR*, 2012.

[PWZ08] J. Porway, L. Wang, and S.C. Zhu. A hierarchical and contextual model for aerial image understanding. In *CVPR*, 2008.

[PY09] C. Poullis and S. You. Automatic reconstruction of cities from remote sensor data. In *CVPR*, 2009.

[RSJ+12] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf. The ISPRS benchmark on urban object classification and 3d building reconstruction. In *Proc. of the ISPRS congress*, 2012.

[SCD+06] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006.

[SDK09] R. Schnabel, P. Degener, and R. Klein. Completion and reconstruction with primitive shapes. In *Eurographics*, 2009.

[SGJM02] A. Srivastava, U. Grenander, G. Jensen, and M. Miller. Jump-Diffusion Markov processes on orthogonal groups for object pose estimation. *Journal of Statistical Planning and Inference*, 103(1/2), 2002.

[SMS07] R. S. Stoica, V. Martinez, and E. Saar. A three dimensional object point process for detection of cosmic filaments. *Journal of the Royal Statistical Society*, 56(4), 2007.

[SMVG02] S. Scholze, T. Moons, and L. Van Gool. A probabilistic approach to building roof reconstruction using semantic labelling. In *DAGM-Symposium*, 2002.

[SS02] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense 2-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1), 2002.

[SSF02] P. Salamon, P. Sibani, and R. Frost. *Facts, Conjectures, and Improvements for Simulated Annealing*. SIAM Monographs on Mathematical Modeling and Computation, 2002.

[SVHVG+08] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008.

[SWK07] R. Schnabel, R. Wahl, and R. Klein. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26(2), 2007.

[SYM10] N. Salman, M. Yvinec, and Q. Merigot. Feature preserving mesh generation from 3D point clouds. In *SGP*, 2010.

[SZS+08] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. Comparative study of energy minimization methods for markov random fields with smoothness-based priors. *Trans. on Pattern Analysis and Machine Intelligence*, 30(6), 2008.

[TMT10] A. Toshev, P. Mordohai, and B. Taskar. Detecting and parsing architecture at city scale from range data. In *CVPR*, 2010.

[TPL07] O. Tournaire, N. Paparoditis, and F. Lafarge. Rectangular road marking detection with marked point processes. In *Proc. of conference on Photogrammetric Image Analysis*, 2007.

[TZ02] Z. Tu and S.C. Zhu. Image Segmentation by Data-Driven Markov Chain Monte Carlo. *Trans. on Pattern Analysis and Machine Intelligence*, 24(5), 2002.

[UB11] A. Utasi and C. Benedek. A 3-D marked point process model for multi-view people detection. In *CVPR*, 2011.

[VAB12] C. Vanegas, D. Aliaga, and B. Benes. Automatic extraction of manhattan-world building masses from 3d laser range scans. *Trans. on Visualization and Computer Graphics*, 18(10), 2012.

[VAM+09] C. Vanegas, D. Aliaga, P. Mueller, P. Waddell, B. Watson, and P. Wonka. Modeling the appearance and behavior of urban spaces. In *EUROGRAPHICS State of the Art Reports*, 2009.

[VKH06] V. Verma, R. Kumar, and S. Hsu. 3D building detection and modeling from aerial LIDAR data. In *CVPR*, 2006.

[VKLP09] H. Vu, R. Keriven, P. Labatut, and J.P. Pons. Towards high-resolution large-scale multiview. In *CVPR*, 2009.

[VL14] Y. Verdie and F. Lafarge. Detecting parametric objects in large scenes by monte carlo sampling. *International Journal of Computer Vision*, 106(1), 2014.

[VLZ11] Y. Verdie, F. Lafarge, and J. Zerubia. Generating compact meshes under planar constraints: an automatic approach for modeling buildings lidar. In *ICIP*, 2011.

[XGC07] H. Xu, N. Gossett, and B. Chen. Knowledge and heuristic-based modeling of laser-scanned trees. *Trans. on Graphics*, 26(4), 2007.

[ZBKB08] L. Zebedin, J. Bauer, K.F. Karner, and H. Bischof. Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In *ECCV*, 2008.

[ZLAK14] H. Zimmer, F. Lafarge, P. Alliez, and L. Kobbelt. Zometool shape approximation. *Graphical Models*, xx, 2014.

[ZN10] Q.Y. Zhou and U. Neumann. $2.5d$ dual contouring: A robust approach to creating building models from aerial lidar point clouds. In *ECCV*, 2010.

[ZN12] Q.Y. Zhou and U. Neumann. 2.5D building modeling by discovering global regularities. In *CVPR*, 2012.