



**HAL**  
open science

# Detecting parametric objects in large scenes by Monte Carlo sampling

Yannick Verdie, Florent Lafarge

► **To cite this version:**

Yannick Verdie, Florent Lafarge. Detecting parametric objects in large scenes by Monte Carlo sampling. *International Journal of Computer Vision*, 2014, 106 (1), pp.57-75. 10.1007/s11263-013-0641-0. hal-00843022

**HAL Id: hal-00843022**

**<https://inria.hal.science/hal-00843022>**

Submitted on 10 Jul 2013

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

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

# Detecting parametric objects in large scenes by Monte Carlo sampling

Yannick Verdié · Florent Lafarge

Received: 28 August 2012 / Accepted: 9 July 2013

**Abstract** Point processes constitute a natural extension of Markov Random Fields (MRF), designed to handle parametric objects. They have shown efficiency and competitiveness for tackling object extraction problems in vision. Simulating these stochastic models is however a difficult task. The performances of the existing samplers are limited in terms of computation time and convergence stability, especially on large scenes. We propose a new sampling procedure based on a Monte Carlo formalism. Our algorithm exploits the Markovian property of point processes to perform the sampling in parallel. This procedure is embedded into a data-driven mechanism so that the points are distributed in the scene in function of spatial information extracted from the input data. The performances of the sampler are analyzed through a set of experiments on various object detection problems from large scenes, including comparisons to the existing algorithms. The sampler is also tested as optimization algorithm for MRF-based labeling problems.

**Keywords** Stochastic modeling · Monte Carlo sampling · object detection · large scenes · energy minimization · point processes · Markov Random Fields

## 1 Introduction

Markov point processes are probabilistic models introduced by Baddeley and Lieshout (1993) to extend the traditional Markov Random Fields (MRF) by using 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 in a scene. A point process is usually specified by three key elements:

**Some parametric objects.** They can be defined in discrete and/or continuous domains. They usually correspond to geometric entities, *e.g.* segments, rectangles, circles or planes, but can more generally be any type of multi-dimensional function. The complexity of the objects directly impacts on the size of the configuration space.

**An energy.** It is used to measure the quality of a configuration of objects. The energy is typically defined as a combination of a term assessing the consistency of objects to the data, and a term taking into account spatial interactions between objects in a Markovian context.

**A sampler.** It allows the search for the object configuration minimizing the energy. As the configuration space is of variable dimension and the energy is usually non-convex, Monte Carlo based samplers capable of exploring the whole configuration space are required, in most cases a Markov Chain Monte

Carlo (MCMC) algorithm (Hastings, 1970; Green, 1995; Liu, 2001).

### 1.1 Related works

The growing interest in these probabilistic models is motivated by the need to manipulate parametric objects interacting in complex scenes. Many works relying on point processes have been recently proposed to address the variety of image and vision problems listed below.

*Population counting.* Descombes et al. (2009) propose a point process for counting populations from aerial images, each entity being captured by an ellipse. Ge and Collins (2009) present a point process for a similar application, but dedicated to crowd detection from ground-based photos, for which objects are defined as a set of body shape templates learned from training data. Multi-view images are used by Utasi and Benedek (2011) to detect people by a point process in 3D where the objects are specified by cylinders.

*Structure extraction.* Sun et al. (2007) and Lacoste et al. (2005) propose point processes for extracting line-networks from images by taking into account spatial interactions between lines to favor the object connection and also certain types of line junctions more likely to appear in real networks. Stoica et al. (2007) extend these line-network models in third dimension for recovering the cosmic filament network from point clouds representing the map of the Universe. The junction point processes developed by Chai et al. (2013) allow the extraction of line-networks using a graph-based representation. Junction points are not associated to geometric shapes, but are instead marked by some angles indicating the directions of the adjacent points so that a junction-point configuration is equivalent to a planar graph. Ortner et al. (2008) and Chai et al. (2012) detect buildings by displacing and connecting rectangles from aerial images. The latter use an auxiliary point process of line-segments to reinforce the rectangle extraction, whereas the former embed the point process into a MRF model to provide a structure-driven segmentation of images.

*Texture analysis.* Nguyen et al. (2010) develop a model for texture recognition in which the spatial distribution of visual keypoints discriminates the textures. Zhu et al. (2005) describe natural textures by a layout of textons, which can be seen as a realization of a point process specified by a texton library. Lafarge et al. (2010) present a general model for extracting different types of geometric features from images, including line, rectangles and disks. A mixture of object interactions are considered such that the process can reconstruct a

large variety of textures.

*Object recognition.* Lieshout (2008) develops a point process for tracking rectangular colored objects from video. A mono-dimensional point process is proposed by Mallet et al. (2010) for modeling 1D-signals by mixtures of parametric functions while imposing physical constraints between the signal modes.

### 1.2 Motivations

The results obtained by these point processes are convincing and competitive with respect to other families of methods, but the performances are particularly limited in terms of computation time and convergence stability, especially on large scenes. These drawbacks explain why industry has been reluctant until now to integrate these mathematical models in their products. Indeed, the works mentioned in Section 1.1 emphasize complex model formulations by proposing parametrically sophisticated objects (Ge and Collins, 2009; Lafarge et al., 2010), advanced techniques to fit objects to the data (Utasi and Benedek, 2011), and non-trivial spatial interactions between objects (Mallet et al., 2010; Ortner et al., 2008; Sun et al., 2007). However, these works usually rely on standard sampling procedures, mainly on the Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm (Green, 1995). The computation time generated by such a sampler is reasonable only from data of small size. For example, the building extraction algorithm proposed by Ortner et al. (2008) requires around six hours from an image portion of size  $1000 \times 1000$  pixels only ( $0.25 \text{ km}^2$  area). Such a solution is obviously not reasonable when dealing with entire aerial and satellite images.

In the literature, few works have addressed the optimization issues from such complex models. The proposed solutions are mainly based on some improvements of the traditional RJMCMC sampler.

*Jump-Diffusion.* Proposed by Grenander and Miller (1994), this algorithm has been designed to speed-up the MCMC sampling by combining diffusion dynamics with a RJMCMC sampler. Both mechanisms play different roles: the former performs reversible jumps between the different subspaces, whereas the latter conducts stochastic diffusion within each continuous subspace, the global process being controlled by a common relaxation parameter. However this algorithm is restricted to specific energy forms (Srivastava et al., 2002; Han et al., 2004; Lafarge et al., 2010).

*Data-Driven MCMC.* Data considerations can also be used to drive the MCMC sampling with more efficiency (Tu and Zhu, 2002). The idea consists in mod-

eling 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.

*Parallelization mechanisms.* Some works have also proposed parallelization procedures by using multiple chains simultaneously (Harkness and Green, 2000) or decomposition schemes in configuration spaces of fixed dimension (Byrd et al., 2010; Gonzalez et al., 2011). However they are limited by border effects, and are not designed to perform on large scenes. In addition, the existing decomposition schemes cannot be used for configuration spaces of variable dimension, and as a consequence, they are not adapted to sample point processes. Parallel tempering (Earl and Deem, 2005) 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 all configuration space.

*Multiple births and deaths.* A mechanism based on multiple creation and destruction of objects has also been developed to address population counting problems (Descombes et al., 2009; Utasi and Benedek, 2011). 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 gains in terms of computation time remain weak and are usually realized at the expense of convergence stability, especially in large scenes. Finding a fast efficient sampler for general Markov point processes clearly represents a challenging problem.

### 1.3 Contributions

We present an original solution to address this problem and to drastically reduce computation times while guaranteeing convergence stability and quality of the reached configurations. Our algorithm presents several important contributions to the field.

**Sampling in parallel.** Contrary to the conventional MCMC sampler which makes the solution evolve by successive perturbations, our algorithm can perform a large number of perturbations simultaneously using a unique chain. The Markovian property of point

processes is exploited to make the global sampling problem spatially independent in a local neighborhood.

**Data-driven mechanism.** Point processes mainly use uniform proposition kernels which are computationally easy to simulate, but make the sampling particularly slow. We propose an efficient mechanism allowing the modifications, creations or removals of objects by taking into account spatial information extracted from the observed data. Contrary to the data-driven solutions proposed by Tu and Zhu (2002) and Ge and Collins (2009), our proposition kernel is not built directly from image likelihood, but is created via a space-partitioning tree in order to guarantee the sampling parallelization.

**Efficient GPU implementation.** We propose an implementation on GPU which significantly reduces computation times with respect to existing algorithms, while increasing stability and improving the quality of the obtained solution. The potential of GPU is efficiently exploited in both optimizing the number of operations in parallel, and limiting the memory transfer between GPU and CPU.

**Original models for object extraction.** To evaluate the performance of the sampler, we propose original point processes for vision problems. In particular, a model for detecting complex 3D objects in large-scale point clouds is designed. This model is applied to tree recognition from laser scans of large urban and natural environments. To our knowledge, it is the first point process sampler to date to perform in such highly complex state spaces.

This paper extends the work presented in (Verdie and Lafarge, 2012) by detailing the proposed sampler and its implementation, by presenting new results, comparisons and performance tests, as well as proposing a version dedicated to the optimization of MRF-based energies for labeling problems.

The paper is organized as follows. Background on point processes is presented in Section 2. Section 3 demonstrates how a RJMCMC-based algorithm can sample point process in parallel. Section 4 details how to create data-driven proposition kernels while keeping the parallelization characteristics of the sampler. Our new sampling procedure is then detailed in Section 5. Experiments with point processes in 2D and 3D are presented and discussed in sections 6 and 7 respectively, including comparisons with the existing algorithms. Note that datasets, results and evaluation tools are available on-



line (Benchmark, 2013). Section 8 demonstrates the interests of our sampler for optimizing energies in case of MRF-based labeling problems, before presenting conclusions in Section 9.

## 2 Point Process background

### 2.1 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

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

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

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  $h(\cdot)$  defined in  $\mathcal{P}$  and a reference measure  $\mu(\cdot)$  under the condition that the normalization constant of  $h(\cdot)$  is finite:

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

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

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

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

$$h(\cdot) \propto \exp -U(\cdot) \quad (4)$$

### 2.2 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  $h(\mathbf{p}) > 0$ ,

- (i)  $\forall \tilde{\mathbf{p}} \subseteq \mathbf{p}, h(\tilde{\mathbf{p}}) > 0$ ,
- (ii)  $\forall u \in K, h(\mathbf{p} \cup \{u\})/h(\mathbf{p})$  only depends on  $u$  and its neighbors  $\{p \in \mathbf{p} : u \sim p\}$ .

The expression  $h(\mathbf{p} \cup \{u\})/h(\mathbf{p})$  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.

### 2.3 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. We denote by  $\mathcal{C}$ , the corresponding space of object configurations where each configuration is given by  $\mathbf{x} = \{x_1, \dots, x_{n(\mathbf{x})}\}$ . 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.

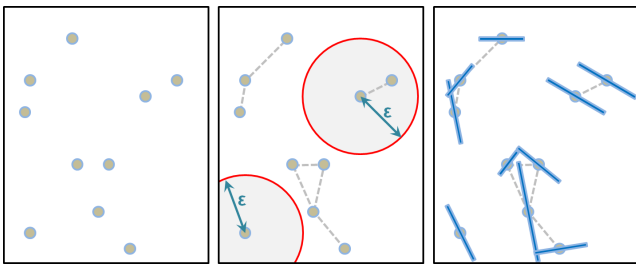
The most popular family of point processes corresponds to the Markov point processes of objects specified by Gibbs energies on  $\mathcal{C}$  of the form

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

where  $\sim$  denotes the symmetric neighborhood relationship of the Markov point process,  $D(x_i)$  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  $\epsilon$  between points such that

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

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 (Szeliski et al., 2008). Our problem can indeed be seen as a generalization of these MRF models. This particular case is detailed in Section 8.



**Fig. 1** 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 grey dashed lines represent the pairs of points interacting with respect to the neighboring relationship which is specified here by a limit distance  $\epsilon$  between two points (Eq. 6).

## 2.4 Simulation

Point processes are usually simulated using a RJMCMC sampler (Green, 1995) 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  $\mathbf{x}$  of the chain is locally perturbed to a configuration  $\mathbf{y}$  according to a density function  $Q(\mathbf{x} \rightarrow \cdot)$ , called a proposition kernel. The perturbations are local, which means that  $\mathbf{x}$  and  $\mathbf{y}$  are very close, and differ by no more than one object in practice. The configuration  $\mathbf{y}$  is then accepted as the new state of the chain with a certain probability depending on the energy variation between  $\mathbf{x}$  and  $\mathbf{y}$ , 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

$$Q(\mathbf{x} \rightarrow \cdot) = \sum_m q_m Q_m(\mathbf{x} \rightarrow \cdot) \quad (7)$$

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 (*e.g.* 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, *i.e.* able to propose the inverse perturbation. Details on kernel computation for image and vision problems can be found in (Descombes, 2011).

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 de-

---

### Algorithm 1 RJMCMC sampler (Green, 1995)

---

- 1- Initialize  $X_0 = \mathbf{x}_0$  and  $T_0$  at  $t = 0$ ;
- 2- At iteration  $t$ , with  $X_t = \mathbf{x}$ ,
  - Choose a sub-kernel  $Q_m$  according to probability  $q_m$
  - Perturb  $\mathbf{x}$  to  $\mathbf{y}$  according to  $Q_m(\mathbf{x} \rightarrow \cdot)$
  - Compute the Green ratio

$$R = \frac{Q_m(\mathbf{y} \rightarrow \mathbf{x})}{Q_m(\mathbf{x} \rightarrow \mathbf{y})} \exp\left(\frac{U(\mathbf{x}) - U(\mathbf{y})}{T_t}\right) \quad (8)$$

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

crease which gives an approximate solution close to the optimum (Baddeley and Lieshout, 1993; Salamon et al., 2002).

## 3 Sampling in parallel

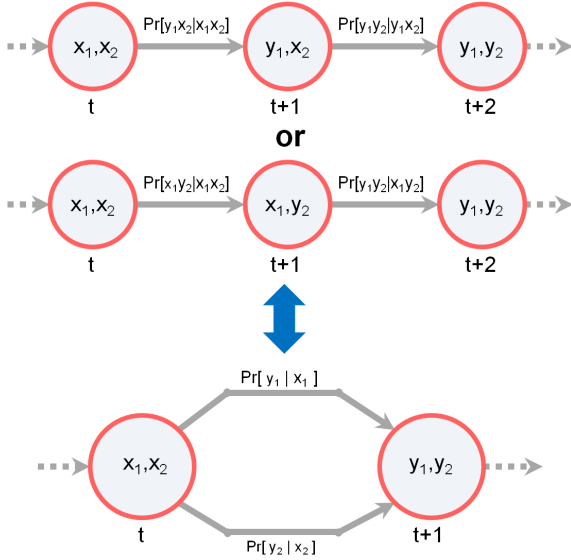
The conventional RJMCMC sampler performs successive perturbations on objects. Such a procedure is obviously long and fastidious, especially for large scale problems. A natural idea but still unexplored for Markov point processes consists in sampling objects in parallel by exploiting their conditional independence outside the 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.

### 3.1 From sequential to parallel sampling

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. In other words, realizing two successive perturbations on independent cells at the same temperature is equivalent to performing them in parallel. To do so, let us consider two cells  $c_1$  and  $c_2$  independent on  $(X_t)_{t \in \mathbb{N}}$ . We denote by  $\mathbf{x}$ , a realization of the point process such that  $\mathbf{x} = (x_1, x_2, u)$  where  $x_1$  (respectively  $x_2$ ) represents the set of points falling in the cell  $c_1$  (respectively  $c_2$ ), and  $u$  is the remaining set

of points falling in  $K - \{c_1, c_2\}$ . Let  $\mathbf{y}$  be a new configuration of points obtained from  $\mathbf{x}$  by two perturbations on the cells  $c_1$  and  $c_2$  so that  $\mathbf{y} = (y_1, y_2, u)$ , as illustrated on Fig. 2.



**Fig. 2** Illustration of the equivalence between two successive perturbations on independent cells  $c_1$  and  $c_2$ , and two simultaneous perturbations on each cell.

The transition probability  $\Pr[X_{t+2} = \mathbf{y} | X_t = \mathbf{x}]$  of moving from the state  $\mathbf{x}$  at time  $t$  to the state  $\mathbf{y}$  at time  $t + 2$  can be expressed as

$$\begin{aligned} \Pr[X_{t+2} = \mathbf{y} | X_t = \mathbf{x}] &= \Pr[X_{t+2} = (y_1, y_2, u) | X_{t+1} = (y_1, x_2, u)] \\ &\quad \times \Pr[X_{t+1} = (y_1, x_2, u) | X_t = \mathbf{x}] \\ &\quad + \Pr[X_{t+2} = (y_1, y_2, u) | X_{t+1} = (x_1, y_2, u)] \\ &\quad \times \Pr[X_{t+1} = (x_1, y_2, u) | X_t = \mathbf{x}] \end{aligned} \quad (9)$$

or, the temperature parameter is constant between  $t$  and  $t + 2$ , which means that

$$\begin{aligned} \Pr[X_{t+2} = \mathbf{y} | X_{t+1} = (y_1, x_2, u)] &= Q((y_1, x_2, u) \rightarrow \mathbf{y}) \\ &\quad \times \min \left[ 1, \frac{Q(\mathbf{y} \rightarrow (y_1, x_2, u))}{Q((y_1, x_2, u) \rightarrow \mathbf{y})} \exp \left( \frac{U((y_1, x_2, u)) - U(\mathbf{y})}{T_{t+1}} \right) \right] \\ &= Q((y_1, x_2, u) \rightarrow \mathbf{y}) \\ &\quad \times \min \left[ 1, \frac{Q(\mathbf{y} \rightarrow (y_1, x_2, u))}{Q((y_1, x_2, u) \rightarrow \mathbf{y})} \exp \left( \frac{U((y_1, x_2, u)) - U(\mathbf{y})}{T_t} \right) \right] \\ &= \Pr[X_{t+1} = \mathbf{y} | X_t = (y_1, x_2, u)] \end{aligned} \quad (10)$$

The cells  $c_1$  and  $c_2$  being independent, the transition probability for the perturbation  $y_2$  falling in  $c_2$  does

not depend, by definition, on  $x_1$  and  $y_1$ . Thus we have in particular

$$\begin{aligned} \Pr[X_{t+1} = (y_1, y_2, u) | X_t = (y_1, x_2, u)] \\ = \Pr[X_{t+1} = (x_1, y_2, u) | X_t = (x_1, x_2, u)] \end{aligned} \quad (11)$$

This leads to the equation

$$\begin{aligned} \Pr[X_{t+2} = \mathbf{y} | X_{t+1} = (y_1, x_2, u)] \\ = \Pr[X_{t+1} = (x_1, y_2, u) | X_t = \mathbf{x}] \end{aligned} \quad (12)$$

Similarly, one can demonstrate that

$$\begin{aligned} \Pr[X_{t+2} = \mathbf{y} | X_{t+1} = (x_1, y_2, u)] \\ = \Pr[X_{t+1} = (y_1, x_2, u) | X_t = \mathbf{x}] \end{aligned} \quad (13)$$

Finally, by inserting Eq. 12 and 13 in Eq. 9, the expected result is obtained

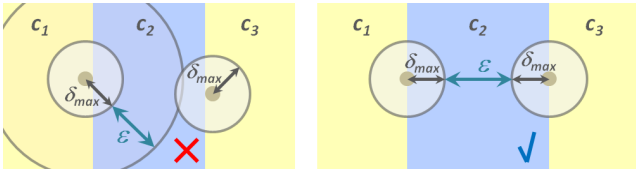
$$\begin{aligned} \Pr[X_{t+2} = \mathbf{y} | X_t = \mathbf{x}] \\ = 2! \Pr[X_{t+1} = (y_1, x_2, u) | X_t = \mathbf{x}] \\ \quad \times \Pr[X_{t+1} = (x_1, y_2, u) | X_t = \mathbf{x}] \end{aligned} \quad (14)$$

where  $2!$  is the combinatorial coefficient corresponding to the number of permutations of perturbations in the sequential chain. This proof can be easily extended by recurrence in case of  $n$  cells mutually independent on  $(X_t)_{t \in \mathbb{N}}$ , with  $n > 2$ .

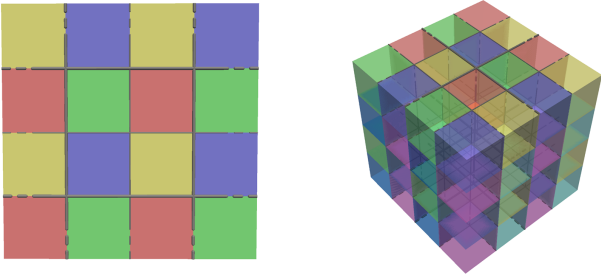
### 3.2 How to guarantee cell independence?

By definition, two cells are independent if the transition probability for any random perturbation falling in the first cell does not depend on the objects and perturbations falling in the second cell, and vice versa. It implies that the two cells must be located at a minimum distance from each other. As illustrated in Fig. 3, this distance must take into account the width  $\epsilon$  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 as object perturbation, denoted by  $\delta_{\max}$ . Considering these two constraints, the independence between two cells  $c_1$  and  $c_2$  can then be guaranteed when

$$\min_{p_1 \in c_1, p_2 \in c_2} \|p_1 - p_2\|_2 \geq \epsilon + 2\delta_{\max} \quad (15)$$



**Fig. 3** 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. 15 is satisfied.



**Fig. 4** Regular partitioning scheme of  $K$ . In dimension two (left), the cells are squares of identical size regrouped into 4 mic-sets (yellow, blue, red and green). Each cell is adjacent to cells belonging to different mic-sets. In dimension three (right), the cells are cubes regrouped into 8 mic-sets.

### 3.3 How to construct a cell partition?

Knowing a condition for insuring the cell independence (Eq. 15), the objective is now to find a partitioning of the space  $K$  optimizing the performance of the sampling in parallel.

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, *i.e.* 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. Fig. 4 illustrates the partitioning scheme for  $\dim K = 2$  and  $\dim K = 3$ . This partitioning scheme guarantees the mutual independence between all the cells of a same set. 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.

However, the number of cells which can be perturbed simultaneously is limited by the computer architecture, in our case by the number of threats available in GPU. When sampling in large scenes, this number is usually much lower than the number of cells in a mic-set if a cell width of  $\epsilon + 2\delta_{\max}$  is imposed. One solution could consist in fixing the cell width so that the number of

cell in a mic-set cannot exceed the maximum number of simultaneous perturbations. This option does not take advantage of the Markovian property, and leads to average performances in practice. In order to fully exploit both the potential of the computer architecture and the Markovian property, a non-regular partitioning of  $K$  is required. This problem is addressed in the next section where spatial information from input data is exploited to ideally partition  $K$ .

## 4 Data-driven mechanism

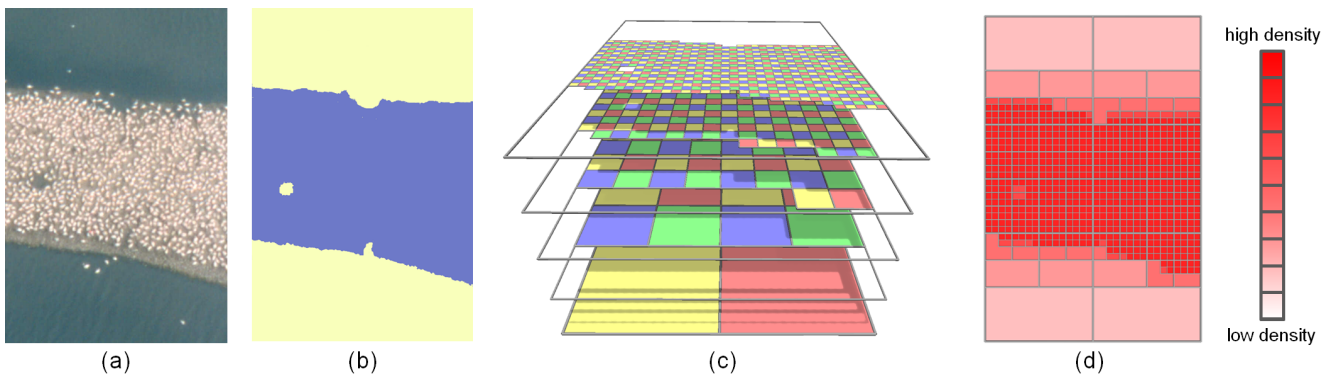
The creation of a non-regular partitioning of  $K$  leads us to consider a proposition kernel which distribute the points non-uniformly in the space  $K$ . In other words, perturbations do not have the same occurrence according to their locations in  $K$ . To design such a proposition kernel, one needs to take into account the characteristics of observed scenes. Contrary to the data-driven solution proposed by Tu and Zhu (2002), our mechanism must be compatible with the parallelization constraints mentioned in Section 3.

### 4.1 Principle

Our idea consists in creating a proposition kernel as an accumulation of uniform sub-kernels spatially restricted on the domain of the cells. Indeed, such a mixture of sub-kernels has the interesting property of still guaranteeing the parallelization of the sampling. In the literature, mixtures of sub-kernels are frequently used in MCMC dynamics to simulate point processes, but with a restricted role where each sub-kernel is dedicated to a perturbation type (*e.g.* birth and death, translation, rotation, etc).

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  $\{c_s\}^{(1)}, \dots, \{c_s\}^{(L)}$ , 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



**Fig. 5** 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.

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. 5.

#### 4.2 Data-driven space-partitioning tree

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. The extraction of such a class is not addressed in this paper, and is supposed to be done by a segmentation algorithm of the literature adapted to the considered application. 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  $\epsilon + 2\delta_{\max}$ , *i.e.* before that the cell independence condition (Eq. 15) is not longer valid.

The space-partitioning tree allows the creation of a proposition kernel in an elegant way as points are naturally and efficiently distributed in  $K$ . On the area of interest, the intensity of the density map is maximal. When moving far from the class of interest, the intensity progressively decreases as shown in Fig. 5, while being ensured to be non-null. In addition, the sampling is not severely affected when the class of interest is inaccurately extracted.

#### 4.3 Proposition kernel formulation

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

$$\forall \mathbf{x} \in \mathcal{C}, Q(\mathbf{x} \rightarrow \cdot) = \sum_{c \in \mathcal{K}} \sum_{t \in \mathcal{T}} q_{c,t} Q_{c,t}(\mathbf{x} \rightarrow \cdot) \quad (16)$$

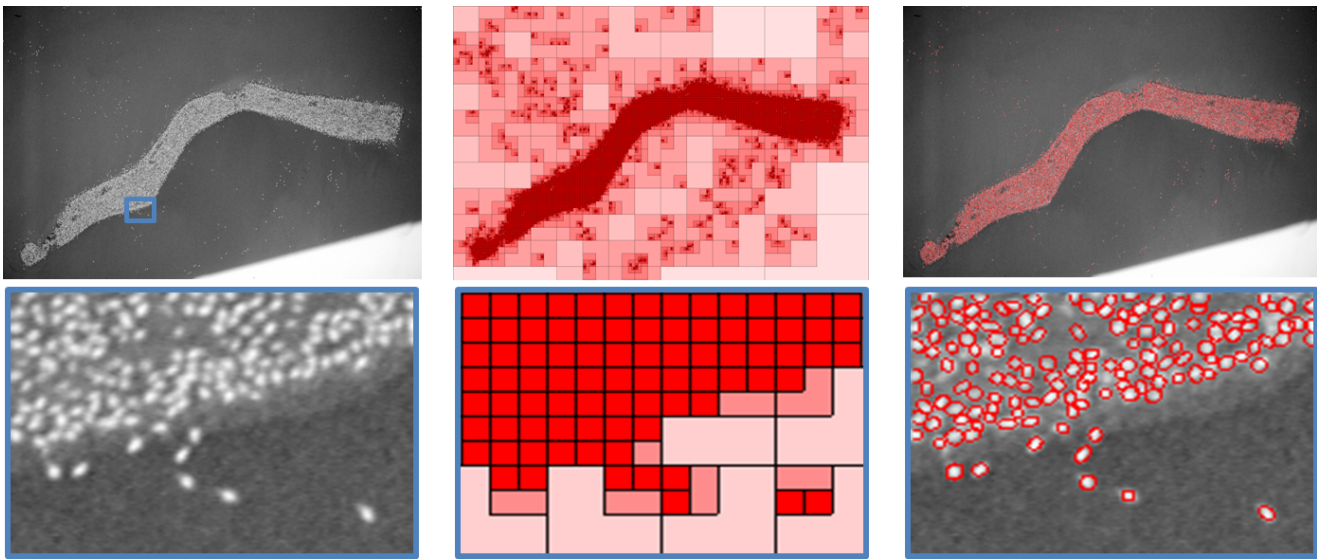
where  $q_{c,t} > 0$  is the probability of choosing the sub-kernel  $Q_{c,t}(\mathbf{x} \rightarrow \cdot)$ . The probability  $q_{c,t}$  allows us to specify the intensity of the density map, given the space-partitioning tree. In practice, this probability is chosen as

$$q_{c,t} = \frac{\Pr(t)}{\#\text{cells in } \mathcal{K}} \quad (17)$$

where  $\Pr(t)$  denotes the probability of choosing the perturbation type  $t \in \mathcal{T}$ . The expression of  $q_{c,t}$  (Eq. 17) 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. As discussed later in Section 6.1, other possible expressions may become more interesting when the class of interest cannot be reliably extracted.

Four types of perturbations are usually considered in practice so that  $\mathcal{T} = \{\text{birth and death, translation, rotation, scale}\}$ . When objects have several possible models, the perturbations consisting in switching the model of an object can also be used. Note that the proposition kernel  $Q$  is reversible as a sum of reversible sub-kernels. Note also that such a proposition kernel allows us to visit the whole configuration space  $\mathcal{C}$ , as guaranteed by the sub-kernels of the coarsest level of  $\mathcal{K}$ .





**Fig. 6** Bird counting by a point process of ellipses. (right) More than ten thousand birds are extracted by our algorithm 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.

## 5 New sampling procedure

The kernel defined in Eq. 16 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.

### 5.1 Algorithm

The proposed sampler, detailed in Algorithm 2, can be seen as a parallelized extension of the traditional RJMCMC with data-driven proposition kernel. As illustrated on Fig. 7, each perturbation is completely independent of the other perturbations simultaneously proposed in the cells of the considered mic-set.

Note that the temperature parameter is updated after each series of simultaneous perturbations such that the temperature decrease is equivalent to a cooling schedule by plateau in a standard sequential MCMC sampling. Note also that the space-partitioning tree protects the sample from mosaic effects. In practice, the sampling is stopped when no perturbation has been accepted during a certain number of iterations.

### 5.2 Implementation

The algorithm has been implemented on GPU using CUDA. A thread is dedicated to each simultaneous perturbation so that operations are performed in parallel for each cell of a mic-set. The sampler is thus all

---

### Algorithm 2 Our data-driven parallel sampler

---

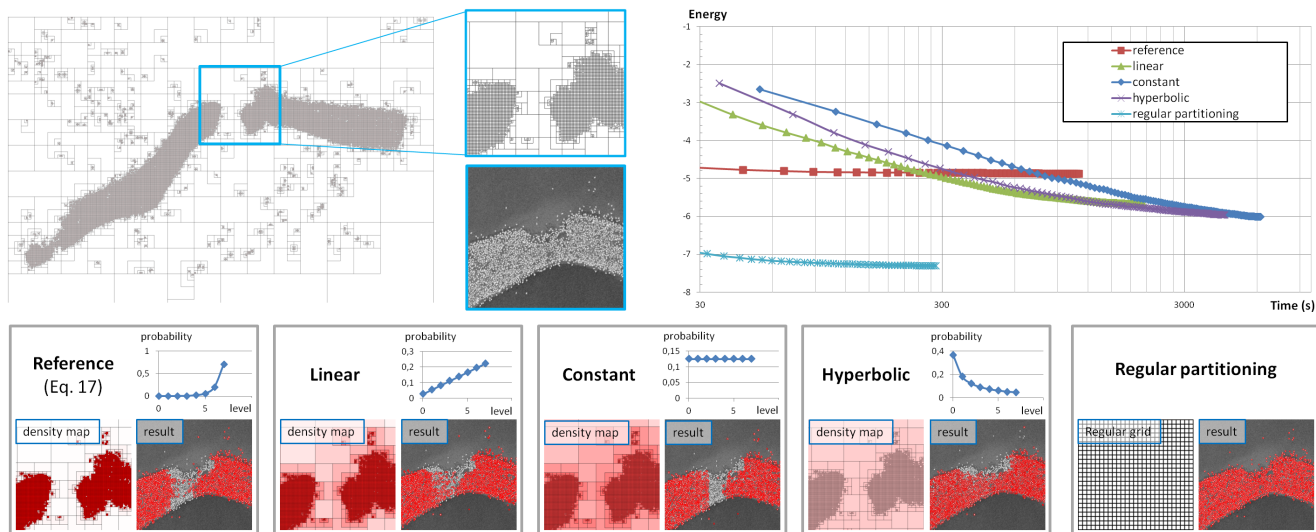
- 1-Initialize  $X_0 = \mathbf{x}_0$  and  $T_0$  at  $t = 0$ ;
- 2-Compute a space-partitioning tree  $\mathcal{K}$ ;
- 3-At iteration  $t$ , with  $X_t = \mathbf{x}$ ,

- 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  $\mathbf{x}$  in the cell  $c$  to a configuration  $\mathbf{y}$  according to  $Q_{c,t}(\mathbf{x} \rightarrow \cdot)$
  - Calculate the Green ratio

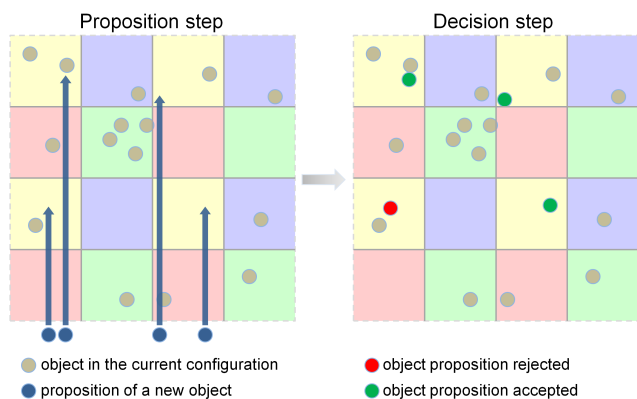
$$R = \frac{Q_{c,t}(\mathbf{y} \rightarrow \mathbf{x})}{Q_{c,t}(\mathbf{x} \rightarrow \mathbf{y})} \exp\left(\frac{U(\mathbf{x}) - U(\mathbf{y})}{T_t}\right) \quad (18)$$

- Choose  $X_{t+1} = \mathbf{y}$  with probability  $\min(1, R)$ , and  $X_{t+1} = \mathbf{x}$  otherwise
  - Update  $T_{t+1} = \alpha T_t$
- 

the more efficient as the mic-set contains many cells, and generally speaking, as the scene supported by  $K$  is large. Moreover, the code has been programmed to avoid time-consuming operations. In particular, the threads do not communicate between each other, and memory coalescing permits fast memory access. The memory transfer between CPU and GPU has also been minimized by indexing the parametric objects. The experiments presented in this section have been performed on a 2.5 Ghz Xeon computer with a Nvidia graphics card (Quadro 4800, architectures 1.3 and 2.0).



**Fig. 10** Behavior of the sampler with a non relevant space-partitioning tree. (top left) When the class of interest cannot be correctly extracted, the space-partitioning tree is of low quality, as shown on the closeup in which a large area of interest is missed. (bottom left frame) The probability  $q_{c,t}$  defined in Eq. 17 does not perform well. (bottom center frames) More interesting choices for  $q_{c,t}$  can then be used, as the linear, constant and hyperbolic formulations which progressively favors the selection of the coarsest levels in the space-partitioning tree. The graph in the frames represents the probability of selecting the space-partitioning tree levels, the value 0 being the coarsest level of the tree. (bottom right frame) Note that the use of a regular partitioning can become especially efficient in such a situation. (top right) the energy decrease graph summarize the performances of these different formulations.



**Fig. 7** Mechanism of the sampler. At a given iteration, a mic-set is chosen in the space-partitioning tree (here, the set of yellow cells), as well as a kernel type (here, births illustrated by the insertion of blue dots). (left) During the proposition step, a perturbation will be proposed simultaneously in each cell of the selected mic-set, independently from each other. (right) Each perturbation will then be either accepted (green dot) or rejected (red dot) during the decision step. The decisions are independent, relying on the Green ratio computations and numbers randomly chosen in the interval  $[0, 1]$ , as formulated in Eq. 18.

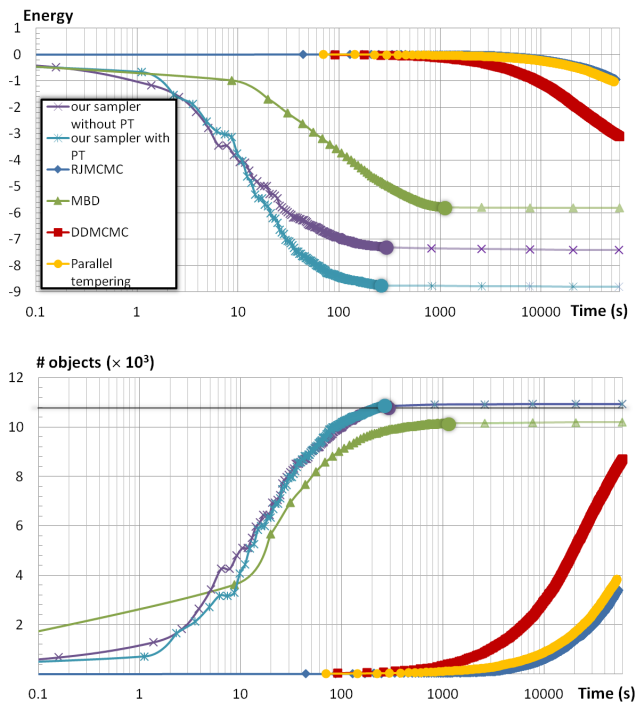
## 6 Experiments with 2D-Point processes

### 6.1 Population counting

The algorithm has been evaluated on population counting problems from large-scale images using a point process in 2D, *i.e.* with  $\dim K = 2$ . The problem presented

in Fig. 6 consists in detecting migrating birds to extract information on their number, their size and their spatial organization. The point process is marked by ellipses which are simple geometric objects defined by a point (center of mass of an ellipse) and three additional parameters. This object shape is well adapted to capture the bird contours. 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 are given in Appendix A. The probability  $\Pr(t)$  of choosing the perturbation type  $t$  is set to 0.2 if  $t = \text{'birth and death'}$ , 0.4 if  $t = \text{'translation'}$ , 0.2 if  $t = \text{'rotation'}$ , and 0.2 if  $t = \text{'scale'}$  in our experiments.

Computation time, quality of the reached energy, and stability are the three important criteria used to evaluate and compare the performance of samplers. As shown on Fig. 8, our algorithm obtains the best results for each of the criteria compared to the existing samplers. In particular, we reach a better energy ( $-8.76$  *vs*  $-5.78$  for Descombes et al. (2009), and  $-2.01$  for Green (1995)) while significantly reducing computation times (269 sec *vs* 1078 sec for Descombes et al. (2009), and  $> 10^5$  sec for Green (1995), Tu and Zhu (2002), and Earl and Deem (2005)). Fig. 8 also underlines an important limitation of the point process sampler of reference for population counting (Descombes et al., 2009) compared to

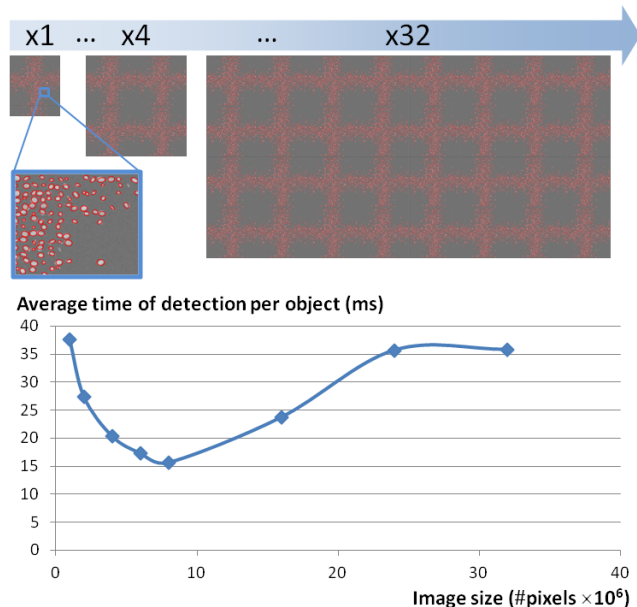


**Fig. 8** Performances of the various samplers. The top graph describes the energy decrease over time from the bird image presented in Fig. 6 (the colored dots correspond to algorithm convergence). Note that time is represented using a logarithmic scale, and that the slow convergence of RJMCMC (Green, 1995), DDMCMC (Tu and Zhu, 2002), and parallel tempering (Earl and Deem, 2005) algorithms is not displayed on the graph. The bottom 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). Note that estimating the correct number of objects does not mean that the objects are correctly fitted to the data, but it is an important criterion for population counting problems as underlined by Lempitsky and Zisserman (2010).

**Table 1** Stability of the various samplers. The coefficients of variation of the energy, time and number of objects reached at the convergence are computed over 50 simulations.

	Coefficient of variation		
	energy	time	#objects
RJMCMC (Green, 1995)	7.3%	4.2%	1.7%
Parallel tempering (Earl and Deem, 2005)	5.0%	20%	4.4%
DDMCMC (Tu and Zhu, 2002)	8.1%	0.7%	1.8%
multiple birth and death (Descombes et al., 2009)	5.0%	2.1%	1.3%
our sampler with regular partitioning	7.4%	6.2%	1.6%
our sampler with space-partitioning tree	4.4%	1.8%	1.1%

our algorithm. Indeed, the discretization of the object parameters required in (Descombes et al., 2009) causes approximate detection and localization of objects which explains the average quality of the reached energy. The stability is analyzed by the coefficient of variation, defined as the standard deviation over mean, and known to be a relevant statistical measure for comparing methods having different means. Our sampler provides a better stability than the existing algorithms, including the multiple birth and death sampler which is supposed to be particularly stable thanks to its semi-deterministic mechanism. Note that the DDMCMC algorithm is particularly stable in terms of time, but is more likely to be stuck in local minimums as energy variation is high.



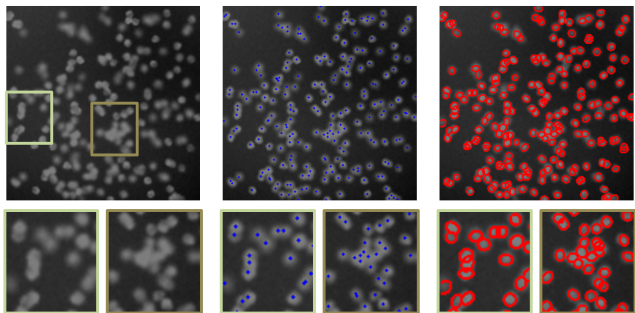
**Fig. 9** Impact of the data size on the computation times. (top) The performances of our sampler have been tested from a set of simulated images of ellipses corrupted by blur and noise, whose size progressively increases. (bottom) The results are displayed on the graph representing the evolution of the average time of detection per object in function of the image size. The minimum average time is met (here, 15 ms) when the optimal occupancy conditions of the GPU architecture are reached. In this experiment, note that the image size is directly proportional to the number of used threads as the cell size at the finest level of the trees is identical for all simulated images.

This gap with the others algorithms in terms of performances becomes more marked when the input scene is larger. Contrary to the existing samplers, the average time of detection per object of our sampler does not explode when increasing the data size, but it falls until reaching the optimal occupancy conditions of the GPU architecture (see Fig. 9). During this stage, there is in-



deed no extra cost in terms of computation time for using additional threads, *i.e.* for increasing the number of simultaneous perturbations. The average time of detection per object then slightly increases before becoming stable.

The impact of the data-driven mechanism is also measured by performing tests with a proposition kernel based on a regular partitioning of the space  $K$  (see Fig. 4). The performances decrease but remain better than the existing algorithms. In particular, the sampler loses stability, and the objects are detected and located less accurately than by using a relevant space-partitioning tree. However the use of a regular partition of  $K$  can become an interesting solution when the class of interest is not correctly extracted from the data, more precisely when entire parts of the class of interest are omitted. Indeed this leads to generate space-partitioning trees of low quality. Fig. 10 shows the behavior of the sampler in such a situation. In particular, one can see that the formulation of the probability  $q_{c,t}$  proposed initially in Eq. 17 is not relevant anymore, and needs to be modified in order to favor the selection of the coarsest levels in the space-partitioning tree. A constant or hyperbolic formulations for  $q_{c,t}$  then become interesting solutions. Because of the construction of the density map, note that the sampler is not affected when the contours of the class of interest are rough.



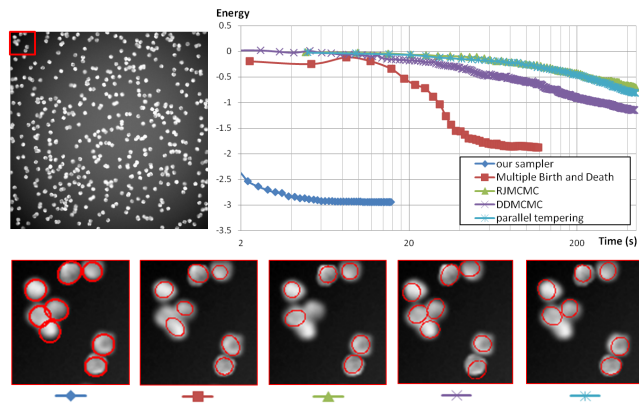
**Fig. 11** Cell counting from (left) the microscope image *cell17*. (middle) Ground Truth shows the location of each cell through a blue cross. (right) Our 2D point process of ellipses captures the cells with very few errors. As illustrated on the right crop, omissions can appear when many cells are regroupped in a tiny area. Note that, in such a case, it is very difficult to visually detect the cells, even for an expert.

Fig. 11 and 12 and Tab. 2 show results on cell counting from microscope images. These images have been simulated by Lehmußola et al. (2007) and are provided with ground truth, *i.e.* the exact number and location of cells are known for each image. Our algorithm has been compared to the supervised approach proposed by

Lempitsky and Zisserman (2010) in Tab. 2. Note that, contrary to our algorithm, their approach just delivers an estimated number of cells per images without locating and delineating them. As shown in Tab. 2, the root mean square error (RMSE) in terms of number of cells from our sampler is more than twice lower than from (Lempitsky and Zisserman, 2010).

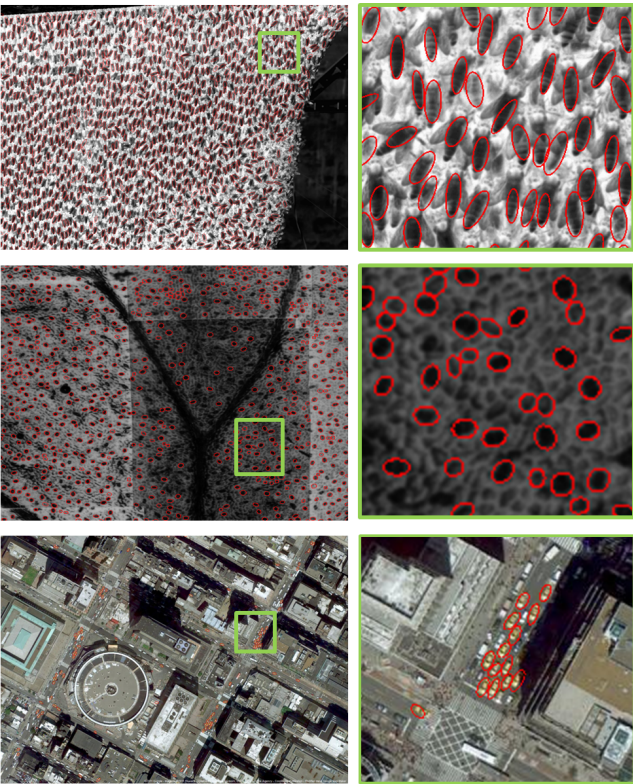
**Table 2** Comparisons with the cell counting approach proposed by Lempitsky and Zisserman (2010) from the microscope images simulated by Lehmußola et al. (2007). The given values correspond to the number of cells detected in a set of images. Our algorithm provides a better estimation of the number of cells than both the  $L1$ - and Tikhonov-regularization versions of (Lempitsky and Zisserman, 2010). In particular, our algorithm is more accurate where cells are highly concentrated.

	our method	Lempitsky ( $L1$ -reg.)	Lempitsky (Tikhonov-reg.)	Ground Truth
<i>cell17</i>	<b>209</b>	202.9	194.1	213
<i>cell18</i>	184	<b>184.6</b>	175.9	185
<i>cell19</i>	<b>187</b>	192.2	180.1	188
<i>cell20</i>	<b>169</b>	174.1	170.4	169
<i>cell21</i>	147	<b>148.6</b>	144.4	149
<i>cell22</i>	<b>184</b>	182.6	176.5	184
<i>cell23</i>	<b>159</b>	158.3	157.6	161
RMSE	<b>1.93</b>	4.71	9.21	-



**Fig. 12** Performances of various samplers on cell counting. 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. Our sampler allows both low running time and good configuration quality.

Note finally that the proposed model is general and can be used for many different population counting problems, as illustrated on Fig. 13. This model does not just count a population but it also provides helpful information on its spatial organization in order to track and analyze its behavior. Note also that ellipses can be



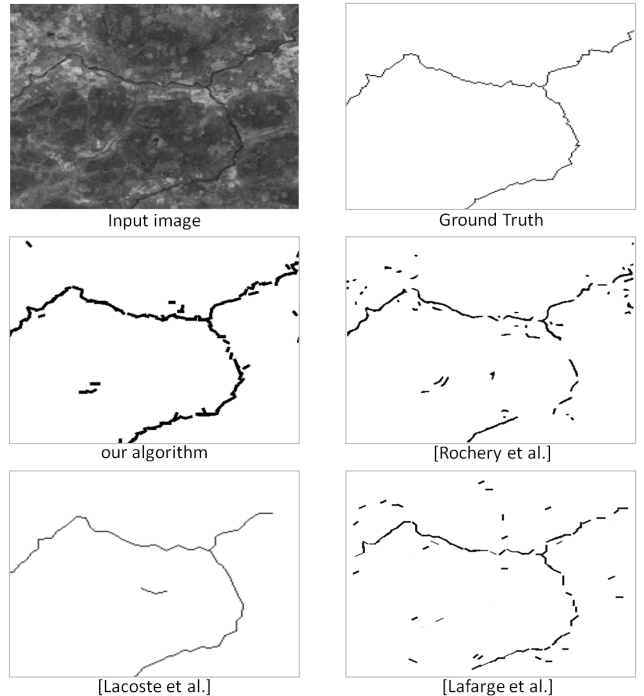
**Fig. 13** Various population counting problems. Our algorithm captures different objects of interest by ellipses in large scenes, as (top) bees from beehive pictures, (middle) opened stomata from microscope images of leaf, and (bottom) yellow cabs from aerial images. 1167 bees (respectively 757 stomata and 87 taxis) are detected in 12 minutes (respectively 168 seconds and 165 seconds). Note that the computation time is higher for bee detection because the partitioning scheme contains few cells, *i.e.* 75. As shown on the close-ups, the objects of interest are globally well detected in spite of the high concentration and overlap of objects.

substituted by any type of parametric 2D-objects in the model formulation.

## 6.2 Structure extraction

The algorithm has also been tested for recovering specific structures from images, in this case line-networks. The parametric objects are specified by line-segments defined as a point (center of mass of a line-segment) and two additional parameters (length and orientation). Contrary to the population counting model previously used, the pairwise potential includes a connection interaction to link the line-segments. This constraint allows the object configurations to be structured as line-networks, each connection between line-segments representing a junction in the line-network. Details on this energy formulation can be found in Appendix B. The probability  $\Pr(t)$  of choosing the perturbation type  $t$  is

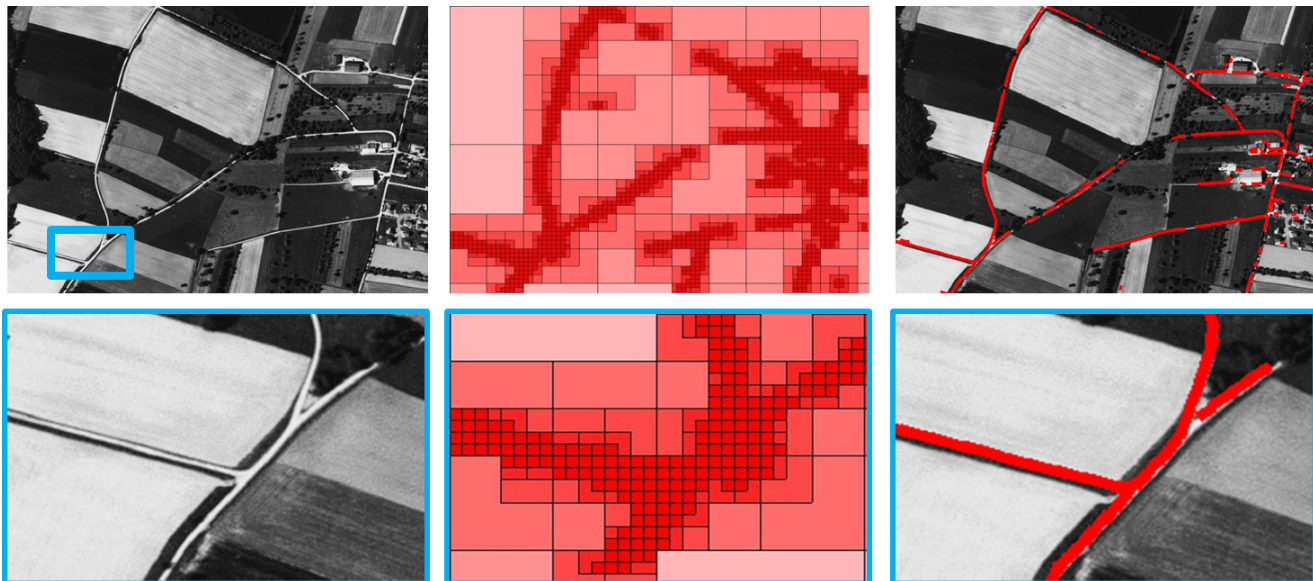
set to 0.4 if  $t = \text{'birth and death'}$ , 0.1 if  $t = \text{'translation'}$ , 0.4 if  $t = \text{'rotation'}$ , and 0.1 if  $t = \text{'scale'}$  in our experiments.



	Time (s)	FPR	FNR	model
our sampler	16.8	0.02	0.25	line
Lafarge et al. (2010)	108	0.02	0.45	line
Lacoste et al. (2005)	2700	0.01	0.35	line
Rochery et al. (2006)	600	0.01	0.50	pixel

**Fig. 15** Extraction of a river network from a satellite image by different methods. The result obtained from our method is visually competitive with respect to existing methods. The structure of the river is correctly recovered by connected line-segments. The accuracy of our algorithm in terms of False Positive / False Negative rates is relatively competitive, being better than Lafarge et al. (2010) and Rochery et al. (2006), and similar to Lacoste et al. (2005). Note that our algorithm significantly improves the computation times.

Figure 14 shows a road network extraction result obtained from a satellite image. Our method significantly improves the computation times with respect to existing methods, as detailed on Benchmark (2013). 16 seconds are required in our case, compared to 7 minutes by a Jump-Diffusion algorithm (Lafarge et al., 2010), 155 minutes for a RJMCMC-based method (Lacoste et al., 2005), and 60 minutes for an Active Contour approach (Rochery et al., 2006). One can see in Fig. 15 that the visual quality of our extracted networks is relatively inferior to those of Lacoste et al. (2005), even if the quantitative results are similar. That said, their



**Fig. 14** Line-network extraction by a point process of line-segments. (middle) Even with a rough density map, (right) the road network is recovered (red segments) by our algorithm in 16 seconds from (left) a satellite image. Similarly to existing methods, some parts of the network can be omitted when roads are hidden by trees at some locations, as shown on the close-up.

method relies on a heavy formalism, *i.e.* the *Quality Candy* model, requiring many parameters (*i.e.* around fifteen) whose tuning by trial and error is a difficult task as several parameters have unstable behaviors.

## 7 Experiments with 3D-Point processes

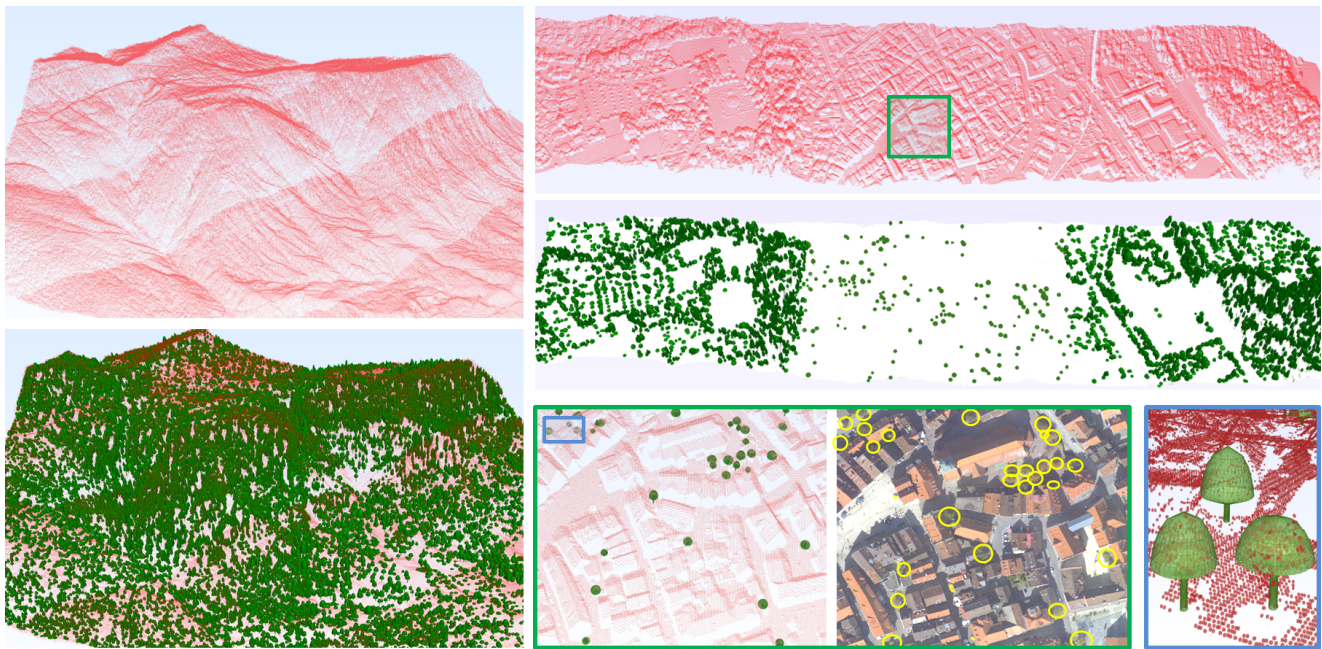
Our algorithm has been tested with  $\dim K = 3$  on an original model for extracting predefined parametric 3D-templates from unstructured point clouds containing a lot of outliers and noise. This model has been used to detect trees from Laser scans of urban environments composed of many other different 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 detailed in Appendix C (Fig. 19 and 20). 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 configuration space  $\mathcal{C}$  is of higher dimension since the objects are parametrically more complex. This allows our algorithm to exploit more deeply its potential. The rotation kernel is not used here since the objects are invariant by rotation. However, a switching kernel is used in order to exchange the type of object (*i.e.* conoidal, elliptical and semi-elliptical). The probability  $\Pr(t)$  of choosing the

perturbation type  $t$  is set to 0.4 if  $t = \text{'birth and death'}$ , 0.2 if  $t = \text{'translation'}$ , 0.1 if  $t = \text{'switching'}$ , and 0.3 if  $t = \text{'scale'}$  in our experiments.

Fig. 16 shows results obtained from laser scans of large urban and natural environments. 30 (respectively 5.4) thousand trees are extracted in 96 (resp. 53) minutes on the  $3.7\text{km}^2$  mountain area (resp.  $1\text{km}^2$  urban area) from 13.8 (resp. 2.3) million input points. The computation times can appear high, but finding non-trivial 3D-objects in such large scenes by point processes is a challenge which, to our knowledge, has not been achieved until now due to the extreme complexity of the state space. Note also that the performances could be improved by reducing the space  $\mathcal{C}$  with a 3D-point process on manifolds, *i.e.* where the z-coordinate of points is determined by an estimated ground surface.

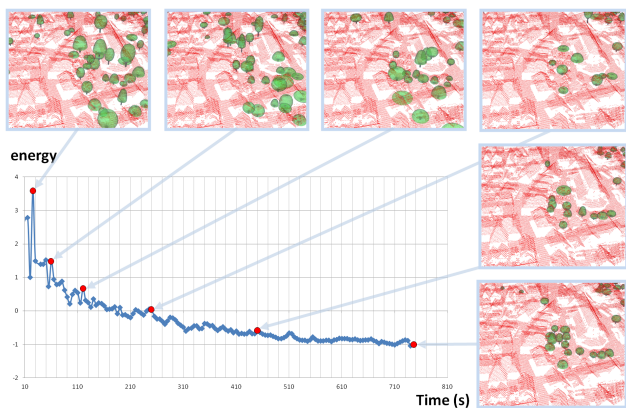
Evaluating the detection quality with accuracy for this application is a difficult task since no ground truth exists. As illustrated on the cropped part in Fig. 16, we have manually indexed the trees on different zones from aerial images acquired with the laser scans. The objects are globally well located and fitted to the input points with few omissions, even when trees are surrounded by other types of urban entities such as buildings. The non-overlapping constraint of the energy allows us to obtain satisfactory results for areas with high tree concentration. Errors may occur in distinguishing the tree type in spite of the tree competition term of the energy. Fig.





**Fig. 16** Tree recognition from point clouds by a 3D-point process specified by 3D-parametric models of trees. Our algorithm detects trees and recognizes their shapes in large-scale (left, input scan: 13.8M points) natural and (top right, input scan: 2.3M points) urban environments, in spite of other types of urban entities, e.g. buildings, car and fences, contained in input point clouds (red dot scans). An aerial image is joined to (bottom right) the cropped part to provide a more intuitive representation of the scene and the tree location. 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.

17 shows the evolution of object configurations during the sampling procedure.



**Fig. 17** Evolution of the object configurations during the sampling. At high temperature, objects of low quality are frequently accepted, leading to non relevant object configurations (top left close-ups). When the temperature decreases, the process becomes progressively selective (top right close-ups). At low temperature, the current object configuration evolves through some local adjustments: the process is stabilizing close to the global minimum (bottom right close-ups).

## 8 Experiments with Markov Random Fields

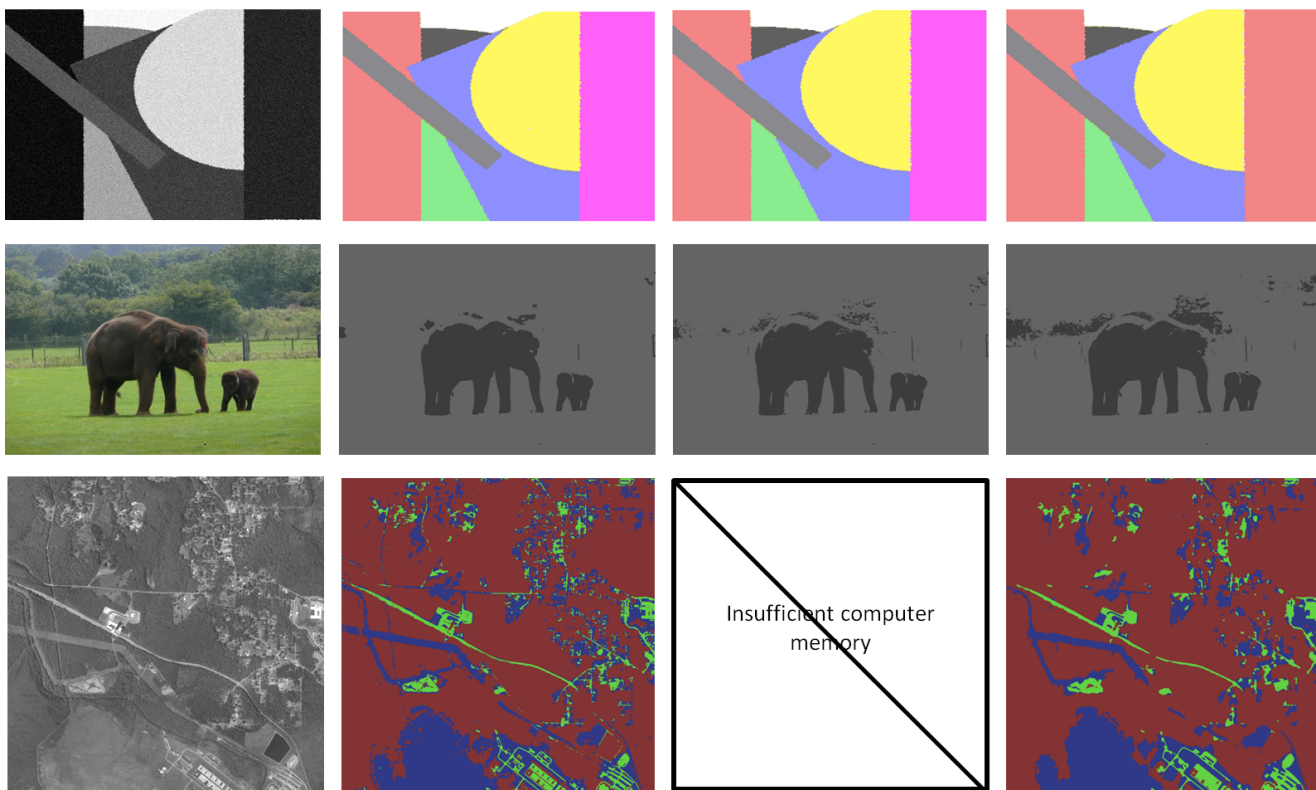
As Markov point processes can be seen as a generalization of MRFs, we evaluated the potential of our sampler for optimizing energies for MRF-based labeling problems. Indeed, traditional MRFs represent simplified point processes having the two following characteristics:

- (i) the dimension of the configuration space is fixed (graph structure is not dynamic anymore, but static),
- (ii) the parametric objects become labels, *i.e.* a finite set of integers.

Under these conditions, the standard energy form of Markov point process (Eq. 5) can be reformulated as

$$U(l) = \sum_{i \in \mathcal{V}} D_i(l_i) + \sum_{(i,j) \in \mathcal{E}} V(l_i, l_j) \quad (19)$$

where  $\mathcal{V}$  is the set of vertices (pixels in case of images),  $\mathcal{E}$  is the set of edges (*i.e.* pairs of adjacent vertices), and  $l \in [1, N]^{\text{card}(\mathcal{V})}$  is a configuration of labels over  $\mathcal{V}$  with  $N$ , the number of labels of the problem. This formulation is actually the standard MRF-energy for labeling problems, as described by Szeliski et al. (2008). Note that there is no constraint imposing on the form of the pairwise interaction term  $V$  when using our sampler.



	our sampler		$\alpha$ -expansion		$\alpha$ - $\beta$ swap		BP		ICM	
	energy	time(s)	energy	time(s)	energy	time(s)	energy	time(s)	energy	time(s)
<b>Simulated</b> (260k pix.) 8 labels, $\epsilon = 1$ pix.	1.3192 ( $\times 10^3$ )	<b>0.92</b>	1.3187 ( $\times 10^3$ )	6.759	1.3186 ( $\times 10^3$ )	7.71	<b>1.3185</b> ( $\times 10^3$ )	8.25	4.1186 ( $\times 10^3$ )	4.7
<b>Elephants</b> (10M pix.) 2 labels, $\epsilon = 1$ pix.	40.74 ( $\times 10^3$ )	<b>40.01</b>	40.64 ( $\times 10^3$ )	45.22	40.64 ( $\times 10^3$ )	73.84	<b>40.63</b> ( $\times 10^3$ )	209.7	40.77 ( $\times 10^3$ )	55.5
<b>Aerial</b> (72M pix.) 3 labels, $\epsilon = 21$ pix.	<b>3.751</b> ( $\times 10^6$ )	<b>2047.6</b>	-	-	-	-	-	-	18.69 ( $\times 10^6$ )	132600

**Fig. 18** Image segmentation. (left) Input images (from top to bottom: Simulated, Elephants and Aerial). (second column) Results obtained by our sampler, by (third column)  $\alpha$ -expansion, and by (last column) ICM. The goal here is not to evaluate the segmentation model which is obviously not optimal, but to compare the results from various optimization techniques with different input image sizes and neighboring distances  $\epsilon$ . (Table) Performances of various optimization algorithms in terms of reached energy and computation time. From a small image, the gain of performance of our sampler is relatively minor: the reached energy is similar to graph-based methods while running times are slightly improved. From a big image and a large neighborhood distance, e.g. Aerial, our algorithm becomes more interesting as the conventional graph-based approaches encounter memory problems whereas iterative methods, e.g. ICM, are extremely slow. Note that the energy value from the ICM algorithm can be poor as this optimization method gets stuck in local minimums.

This point constitutes an important advantage compared to graph-cut based methods.

In order to compare the potential of our sampler with the other optimization algorithms, we assume that no data knowledge can be extracted from the input images. A regular partitioning scheme (Fig. 4) is then used, and the labels at each pixel are perturbed randomly. The width of a cell is given by the cell independence

condition (Eq. 15). As  $\delta_{max} = 0$  (objects are fixed spatially), the width of cells must be superior or equal to the limit distance  $\epsilon$  of the neighborhood relationship. For instance in case of an image labeling problem with a 4- or 8-connexity neighborhood, the condition simply implies that each cell can be associated to one single pixel.

The performances of our sampler have been tested from

a basic model for image segmentation. The unitary data term  $D_i(l_i)$  measures the quality of label  $l_i$  at pixel  $i$  through Gaussian distributions. More precisely, the radiometry distribution of each class is modeled by a Gaussian law whose mean and standard deviation are model parameters to be estimated or fixed by a user. The potential  $V$  corresponds to the conventional Potts model (Li, 2001) so that the labeling is smoothed in a local neighborhood of length  $\epsilon$ .

Fig. 18 shows the performances obtained from various images and provides comparisons with the standard optimization techniques<sup>1</sup>, *i.e.* max-product Belief Propagation (BP) (Weiss and Freeman, 2001), Graph-Cut based algorithms (Boykov et al., 2001) and Iterated Conditional Modes (ICM) (Besag, 1986). Our sampler competes well with these algorithms. From a small input image, the reached energy is usually slightly higher than by using  $\alpha$ -expansion,  $\alpha$ - $\beta$  swap or BP, but the computation time is lower. The gain of time becomes especially attractive when the image size and the neighborhood length increase. In particular, the results obtained from *Aerial*, show that our sampler proposes an interesting alternative to the standard graph-based optimization techniques which encounter memory problems from such very big images, here  $8,500 \times 8,500$  pixels with neighborhood radius  $\epsilon = 21$  pixels.

Our sampler benefits from three advantages compared to graph-cut methods: (i) it can be performed from very big images without memory problems, (ii) the potential term  $V$  can have any form, and (iii) data-driven proposition kernels can be introduced in the sampler. For instance, one can use an edge descriptor from the input image to create a space-partitioning tree in which the lowest levels focus on discontinuities whereas the highest levels target the homogeneous zones of the image. Note also that the neighborhood distance  $\epsilon$  can be considered as variable on the image domain. In particular, one can spatially adapt the subdivision stopping criterion of cells in function of local maximal neighborhood distances.

## 9 Conclusion

We propose a new algorithm to sample point processes whose strengths lean on the exploitation of Markovian properties to enable the sampling to be performed in parallel, and the integration of a data-driven mechanism allowing efficient distributions of the points in the scene. Our algorithm improves the performances of the

existing samplers in terms of computation times and stability, especially on large scenes where the gain is very important. It can be used without particular restrictions, contrary to most samplers, and even appears as an interesting alternative to the standard optimization techniques for MRF labeling problems. In particular, one can envisage using the model proposed in Section 7 to extract any type of parametric objects from large 3D-point clouds. This new algorithm has two limitations. First the performances are not significant when dealing with small input data, as a result of a partitioning of the space  $K$  restricted in terms of number of cells. Second, the performance of the algorithm depends on the potential of the graphics card of the computer.

In future works, it would be interesting to extend the algorithm to point processes in 4D for addressing spatio-temporal problems in which 3D-objects evolve during time. Also, one could improve the algorithm to optimize MRF-based energies and develop a more competitive solution by modeling a data-driven proposition kernel whose density map is of lower intensity in homogeneous regions.

## Appendix A: Population counting model

Let  $x$  denote a configuration of ellipses for which the center of mass of an ellipse is contained in the compact set  $K$  supporting the input image (see Fig. 19). The energy follows the form specified by Eq. 5. The unitary data term  $D(x_i)$  and the potential  $V(x_i, x_j)$  are given by:

$$D(x_i) = \begin{cases} 1 - \frac{d(x_i)}{d_0} & \text{if } d(x_i) < d_0 \\ \exp\left(\frac{d_0 - d(x_i)}{d_0}\right) - 1 & \text{otherwise} \end{cases} \quad (20)$$

$$V(x_i, x_j) = \beta \frac{A(x_i \cap x_j)}{\min(A(x_i), A(x_j))} \quad (21)$$

where

- $d(x_i)$  represents the Bhattacharyya distance between the radiometry inside and outside the object  $x_i$ :

$$d(x_i) = \frac{(m_{in} - m_{out})^2}{4(\sigma_{in}^2 + \sigma_{out}^2)} - \frac{1}{2} \ln\left(\frac{2\sigma_{in}\sigma_{out}}{\sigma_{in}^2 + \sigma_{out}^2}\right) \quad (22)$$

where  $m_{in}$  and  $\sigma_{in}$  (respectively  $m_{out}$  and  $\sigma_{out}$ ) are the intensity mean and standard deviation in  $S_{in}$  (respectively in  $S_{out}$ ).

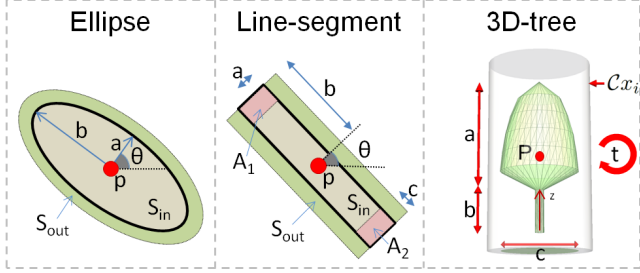
- $d_0$  is a coefficient fixing the sensitivity of the object fitting. The higher the value of  $d_0$ , the more selective the object fitting. In particular,  $d_0$  has to be high when the input images are corrupted by a significant amount of noise.

<sup>1</sup> GCO C++ library (<http://vision.csd.uwo.ca/code/>).



- $A(x_i)$  is the area of object  $x_i$ .
- $\beta$  is a coefficient weighting the non-overlapping constraint with respect to the data term.

Note that a basic mathematical dilatation is used in practice to roughly extract the class of interest from the image of birds for creating a space-partitioning tree.



**Fig. 19** Objects and their parameters for the various presented models. (left) Ellipses and (middle) line-segments are defined by a 2D-point  $p \in K$  (center of mass of the object) and some marks. These additional parameters are the semi-major axis  $b$ , the semi-minor axis  $a$ , and the angle  $\theta$  for an ellipse, and the semi-length  $b$ , the semi-width  $a$ , the orientation  $\theta$ , and the anchor length  $c$  for a line-segment. The inside (respectively bordering) volume of the object is denoted by  $S_{in}$  (respectively  $S_{out}$ ). The anchors are denoted by  $A_1$  and  $A_2$ . (right) 3D-trees are defined by a 3D-point  $p \in K$  (center of mass of the object), a type  $t \in \{\text{conoidal, ellipsoidal, semi-ellipsoidal}\}$  illustrated on Fig. 20, and 3 additional parameters which are the canopy height  $a$ , the trunk height  $b$  and the canopy diameter  $c$ . The cylindrical volume  $\mathcal{C}x_i$  represents the attraction space of object  $x_i$  in which the input points are used to measure the quality of this object.

## Appendix B: Line-network extraction model

A line-segment is defined by five parameters, including the 2D point corresponding to the center of mass of the object (Fig. 19). Similarly to the population counting model detailed in Appendix A, the fitting quality with respect to the data is based on the Bhattacharyya distance: the unitary data term  $D(x_i)$  of the energy is given by Eq. 20. The potential  $V(x_i, x_j)$  penalizes strong object overlaps (see Eq. 21), but also takes into account a connection interaction in order to favor the linking of the line-segments. The potential term is thus given by:

$$V(x_i, x_j) = \beta_1 \frac{A(x_i \cap x_j)}{\min(A(x_i), A(x_j))} + \mathbf{1}_{x_i \sim_{nc} x_j} \times \beta_2 f(x_i, x_j) \quad (23)$$

where

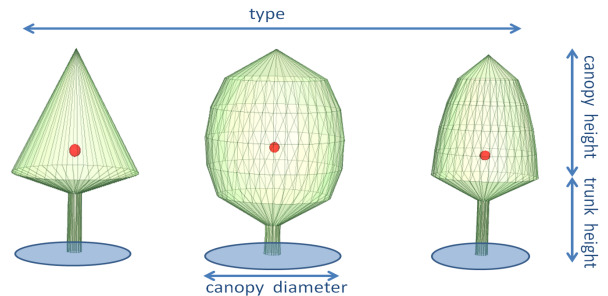
- $\beta_1$  and  $\beta_2$  are two coefficients weighing respectively the non-overlapping and connection constraints with respect to the data term.

- $\sim_{nc}$  is the non-connection relationship between two objects.  $x_i \sim_{nc} x_j$  if the anchor areas of  $x_i$  and  $x_j$  (see Fig. 19) do not overlap.
- $\mathbf{1}_{condition}$  is the indicative function returning one when *condition* is valid, and zero otherwise.
- $f(x_i, x_j)$  is a symmetric function weighting the penalization of two non-connected objects  $x_i$  and  $x_j$  with respect to their average fitting quality. The function  $f$  is introduced to slightly relax the connection constraint when the two objects are of very good quality.

As for the bird counting problem, a basic mathematical dilatation has been used to roughly extract the class of interest from the aerial image shown on Fig. 14. Indeed the pixels corresponding to the class road in this image are relatively bright compared to the background. The segmented result is obviously not optimal, but sufficient to create an efficient space-partitioning tree.

## Appendix C: Tree recognition model formulation

Let  $x$  represent a configuration of 3D-models of trees from a template library described in Fig. 20. The center of mass  $p$  of a tree is contained in the compact set  $K$  supporting the 3D bounding box of the input point cloud (Fig. 19). We denote by  $\partial x_i$  the surface of the object  $x_i$ , and by  $\mathcal{C}x_i$  the cylindrical volume having a vertical axis passing through the center of mass of  $x_i$ , in which the input points are considered to measure the quality of  $x_i$ . The unitary data term  $D(x_i)$  and the



**Fig. 20** Library of tree models - the objects are specified by a 3D point (center of mass illustrated by a red dot) and additional parameters (blue arrows) including the canopy type whose shape can be conoidal (*e.g.* pine or fir), ellipsoidal (*e.g.* poplar or tilia) or semi-ellipsoidal (*e.g.* oak or maple).

pairwise potential  $V(x_i, x_j)$  are given by:

$$D(x_i) = \frac{1}{|\mathcal{C}x_i|} \prod_{p_c \in \mathcal{C}x_i} \gamma(d(p_c, \partial x_i)) \quad (24)$$

$$V(x_i, x_j) = \beta_1 V_{overlap}(x_i, x_j) + \beta_2 V_{competition}(x_i, x_j) \quad (25)$$

where

- $|\mathcal{C}x_i|$  is a coefficient normalizing the unitary data term with respect to the number of input points contained in  $\mathcal{C}x_i$ .
- $d(p_c, \partial x_i)$  is a distance measuring the coherence of the point  $p_c$  with respect to the object surface  $\partial x_i$ .  $d$  is not the traditional orthogonal distance from point to surface because, as real trees do not describe ellipsoidal/conoidal shapes, input points are not homogeneously distributed on the object surface. Here,  $d$  is defined as the combination of the planimetric distance, *i.e.* the projection in the plane of equation  $z = 0$  of the Euclidean distance, and the altimetric variation such that points outside the object are more penalized than inside points. Note that  $d$  is invariant by rotation around the Z-axis.
- $\gamma(\cdot) \in [-1, 1]$  is a quality function which is strictly increasing.
- $V_{overlap}$  is the pairwise potential penalizing strong overlapping between two objects, and given by:

$$V_{overlap}(x_i, x_j) = \frac{A(x_i \cap x_j)}{\min(A(x_i), A(x_j))} \quad (26)$$

where  $A(x_i)$  is the area of the object  $x_i$  projected onto the plane of equation  $z = 0$ .

- $V_{competition}$  is the pairwise potential favoring a similar tree type  $t$  in a local neighborhood:

$$V_{competition}(x_i, x_j) = \mathbf{1}_{t_i \neq t_j} \quad (27)$$

where  $\mathbf{1}$  is the indicative function.

- $\beta_1$  and  $\beta_2$  are two coefficients weighting respectively the non-overlapping constraint and the competition term with respect to the data term.

In order to roughly extract the class of interest from the point clouds, the scatter descriptor proposed by Lafarge and Mallet (2012) is used to identify the points which potentially correspond to trees.

## Acknowledgments

This work was partially funded by the European Research Council (ERC Starting Grant ‘‘Robust Geometry Processing’’, Grant agreement 257474). The authors thank A. Lehmußola, V. Lempitsky, H. Bischof,

R. Ehrlich, the French Mapping Agency (IGN), the Tour du Valat, and the BRGM for providing the datasets, as well as the reviewers for their valuable comments.

## References

- Baddeley, A. J., Lieshout, M. V., 1993. Stochastic geometry models in high-level vision. *Journal of Applied Statistics* 20 (5-6).
- Benchmark, 2013. Datasets, results and evaluation tools available at <http://www-sop.inria.fr/members/Florent.Lafarge/benchmark/evaluation.html>.
- Besag, J. E., 1986. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society* 48 (3).
- Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. *PAMI* 23 (11).
- Byrd, J., Jarvis, S., Bhalerao, A., 2010. On the parallelisation of mcmc-based image processing. In: *IEEE International Symposium on Parallel and Distributed Processing*. Atlanta, U.S.
- Chai, D., Forstner, W., Lafarge, F., 2013. Recovering line-networks in images by junction-point processes. In: *CVPR*. Portland, US.
- Chai, D., Forstner, W., Yang, M. Y., 2012. Combine Markov random fields and marked point processes to extract building from remotely sensed images. In: *ISPRS congress*. Melbourne, Australia.
- Descombes, X. (Ed.), 2011. *Stochastic Geometry for Image Analysis*. Wiley-ISTE.
- Descombes, X., Minlos, R., Zhizhina, E., 2009. Object extraction using a stochastic birth-and-death dynamics in continuum. *Journal of Mathematical Imaging and Vision* 33 (3).
- Earl, D., Deem, M., 2005. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics* 23 (7).
- Ge, W., Collins, R., 2009. Marked point processes for crowd counting. In: *CVPR*. Miami, U.S.
- Gonzalez, J., Low, Y., Gretton, A., Guestrin, C., 2011. Parallel Gibbs sampling: From colored fields to thin junction trees. *Journal of Machine Learning Research*.
- Green, P., 1995. Reversible Jump Markov Chains Monte Carlo computation and Bayesian model determination. *Biometrika* 82 (4).
- Grenander, U., Miller, M., 1994. Representations of Knowledge in Complex Systems. *Journal of the Royal Statistical Society* 56 (4).



- Han, F., Tu, Z. W., Zhu, S., 2004. Range image segmentation by an effective jump-diffusion method. *PAMI* 26 (9).
- Harkness, M., Green, P., 2000. Parallel chains, delayed rejection and reversible jump mcmc for object recognition. In: *BMVC*. Bristol, U.K.
- Hastings, W., 1970. Monte Carlo sampling using Markov chains and their applications. *Biometrika* 57 (1).
- Lacoste, C., Descombe, X., Zerubia, J., 2005. Point processes for unsupervised line network extraction in remote sensing. *PAMI* 27 (10).
- Lafarge, F., Gimel'farb, G., Descombes, X., 2010. Geometric feature extraction by a multi-marked point process. *PAMI* 32 (9).
- Lafarge, F., Mallet, C., 2012. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. *IJCV* 99 (1).
- Lehmussola, A., Ruusuvauro, P., Selinummi, J., Huttenen, H., Yli-Harja, O., 2007. Computational framework for simulating fluorescence microscope images with cell populations. *IEEE Trans. on Medical Imaging* 26 (7).
- Lempitsky, V., Zisserman, A., 2010. Learning to count objects in images. In: *NIPS*. Vancouver, Canada.
- Li, S., 2001. *Markov Random Field Modeling in Image Analysis*. Springer.
- Lieshout, M. V., 2008. Depth map calculation for a variable number of moving objects using markov sequential object processes. *PAMI* 30 (7).
- Liu, J., 2001. *Monte Carlo Strategies in Scientific Computing*. Springer.
- Mallet, C., Lafarge, F., Roux, M., Soergel, U., Bretar, F., Heipke, C., 2010. A marked point process for modeling lidar waveforms. *IP* 19 (12).
- Nguyen, H.-G., Fablet, R., Bouchet, J., 2010. Spatial statistics of visual keypoints for texture recognition. In: *ECCV*. Heraklion, Greece.
- Ortner, M., Descombes, X., Zerubia, J., 2008. A marked point process of rectangles and segments for automatic analysis of digital elevation models. *PAMI* 30 (1).
- Rochery, M., Jermyn, I., Zerubia, J., 2006. Higher order active contours. *IJCV* 69 (1).
- Salamon, P., Sibani, P., Frost, R., 2002. *Facts, Conjectures, and Improvements for Simulated Annealing*. SIAM Monographs on Mathematical Modeling and Computation, Philadelphia, U.S.
- Srivastava, A., Grenander, U., Jensen, G., Miller, M., 2002. Jump-Diffusion Markov processes on orthogonal groups for object pose estimation. *Journal of Statistical Planning and Inference* 103 (1/2).
- Stoica, R. S., Martinez, V., Saar, E., 2007. A three dimensional object point process for detection of cosmic filaments. *Journal of the Royal Statistical Society* 56 (4).
- Sun, K., Sang, N., Zhang, T., 2007. Marked point process for vasculartree extraction on angiogram. In: *EMMCVPR*. Ezhou, China.
- Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C., 2008. Comparative study of energy minimization methods for markov random fields with smoothness-based priors. *PAMI* 30 (6).
- Tu, Z., Zhu, S., 2002. Image Segmentation by Data-Driven Markov Chain Monte Carlo. *PAMI* 24 (5).
- Utasi, A., Benedek, C., 2011. A 3-D marked point process model for multi-view people detection. In: *CVPR*. Colorado Springs, U.S.
- Verdie, Y., Lafarge, F., 2012. Efficient Monte Carlo sampler for detecting parametric objects in large scenes. In: *ECCV*. Firenze, Italy.
- Weiss, Y., Freeman, W., 2001. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Trans. on Information Theory* 47 (2).
- Zhu, S., Guo, C., Wang, Y., Xu, Z., 2005. What are Textons? *IJCV* 62 (1-2).