HAL
archives-ouvertes.fr

Multiple Birth and Cut Algorithm for Point Process Optimization

Ahmed Gamal Eldin, X. Descombes, Josiane Zerubia

▶ To cite this version:

Ahmed Gamal Eldin, X. Descombes, Josiane Zerubia. Multiple Birth and Cut Algorithm for Point Process Optimization. Signal-Image Technology and Internet-Based Systems (SITIS 2010), Dec 2010, Kuala Lumpur, Malaysia. inria-00516305v3

HAL Id: inria-00516305
https://hal.inria.fr/inria-00516305v3

Submitted on 15 Oct 2010

Multiple Birth and Cut Algorithm for Point Process Optimization

Ahmed Gamal-Eldin, Xavier Descombes and Josiane Zerubia
INRIA Sophia-Antipolis Méditerannée
2004 route des Lucioles, BP 93
06902 Sophia-Antipolis, Cedex, France
{ahmed.gamal_eldin, xavier.descombes, josiane.zerubia}@inria.fr

Abstract—In this paper, we describe a new optimization method which we call Multiple Birth and Cut (MBC). It combines the recently developed Multiple Birth and Death (MBD) algorithm and the Graph-Cut algorithm. MBD and MBC optimization methods are applied to the energy minimization of an object based model, the marked point process. We compare the MBC to the MBD showing the advantages and disadvantages, where the most important advantage is the reduction of the number of parameters. We validated our algorithm on the counting problem of flamingos in colony, where our algorithm outperforms the performance of the MBD algorithm.

I. INTRODUCTION

Automatic detection and extraction of multiple objects is of major importance, finding applications in different domains such as evaluation of population of trees, animals, cells, cartography, urban planning and military intelligence. A recent, object based method, embedded in a marked point process (MPP) framework, proved to be efficient for solving many challenging problems dealing with high resolution orthoimages. The framework was first introduced in [1]. The MPP modeling is based on defining a configuration space of the objects of interest, to which a Gibbs energy function is attached. Minimizing this energy function corresponds to the correct detection of the objects of interest.

A naive way of sampling this density function is the birth and death algorithm. At each iteration, there is one possible move, one object is either added to or removed from the current configuration [2] [3]. This algorithm is extreme slow in image processing. Then came the Reverse Jump Markov Chain Monte Carlo method that has been widely used for MPP in image processing [4] [5] [6] because of its flexibility, specially, when using updating schemes such as Metropolis Hasting [7]. To the birth step, moves are now added such as split, translate, rotate, etc. The main limitation is that it treats one object at a time and has a rejection rate. A faster algorithm so called Multiple Birth and Death (MBD) was proposed in [8]. Its advantages are as follows: making possible multiple perturbations in parallel, very good speed of convergence and simplicity of implementation.

Our main contribution in this paper is the proposition of a new optimization technique, the MBC. It combines ideas from MBD and the popular graph cut algorithm. The MBC algorithm major advantage is the very reduced number of parameters furthermore there is no simulated annealing scheme.

This paper is organized as follows: we start by briefly introducing the MPP model in section II. In section III, we describe the multiple birth and death algorithm for MPP model optimization, then the Graph-Cut algorithm. We then propose the new algorithm. In section IV we show detection results and a comparison between the new algorithm and MBD, and we conclude in section V.

II. MARKED POINT PROCESS

Marked point process framework is adapted to define some probabilistic models on configuration spaces consisting of an unknown number of parametric objects. Adding the Markov property allows the introduction of local interactions and defining a prior on the object distribution in the scene. This framework can be interpreted as a generalization of the Markov Random Field (MRF) theory, where the number of random variables is unknown. Moreover, an object is associated to each variable, on which some geometric constraints can be modeled.

A. Point Process

Definition. Point processes are mathematical models for irregular or random point patterns. First, let us consider a point process X living in $K = [0, I_{max}] \times [0, J_{max}]$. X is a measurable mapping from a probability space $(\Omega, \mathcal{A}, \mathcal{P})$ to the set of configuration of points of K. K is a closed, connected subset of $\mathbb{R}^2$, this mapping defines a point process. Basically, a point process is a random variable whose realizations are random point configurations.

Poisson Point Process. The most random (w.r.t. entropy) point process is the Poisson point process. Let $\lambda(.)$ be a positive measure on K and consider a Poisson point process X with intensity measure $\lambda(.)$ on K. The density of the process with mean $\lambda(K)$ is [3]:

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

where n is the number of points in the configuration.

Marked Point Process. Point processes were introduced in image processing because they easily allow to model scenes made of objects. Objects can have simple or complex shapes: simple shapes like lines for roads detection [4], rectangles for buildings [9], ellipses for trees [5] and flamingos [10];

complex shape, using active contours for complex forms, like tree crowns [11]. The geometrical parameters of any of these shapes represent the mark m_i associated to each point x_i. In the flamingo case for instance, the object is an ellipse. Let M be the mark space, $M = [a_{min}, a_{max}] \times [b_{min}, b_{max}] \times [0, \pi[$, where a and b are the major and the minor axis respectively, for which we select a minimum and a maximum value. Therefore, an object is defined as $\omega_i = (x_i, m_i) \in K \times M$, where x_i represents the object location and m_i its mark.

We consider a marked point process with points lying in K and with marks in M, the configuration space is then defined as:

$$\Omega = \bigcup_{n=0}^{\infty} \Omega_n, \ \ \Omega_n = \{\{\omega_1, \ldots, \omega_n\} \subset K \times M\}, \quad (1)$$

where Ω_n is the subset of configurations containing exactly n objects. n is a random variable which is unknown, it is the number of objects to be detected.

Let $\omega = \{\omega_i, i = 1, \ldots, n\}$, we define a reference measure which is the product of Poisson measure $\nu(\omega)$ and the Lebesgue measures on the mark space by:

$$d\pi_r(\omega) = d\nu(x) \prod_{i=1}^{n} (d\mu(m_i)),$$

The process is defined on the configuration space Ω as follows:

$$d\pi(\omega) = f(\omega) d\pi_r(\omega) \quad (2)$$

where $f(.)$ represents the prior and the data term.

Markov Point Process. The most interesting model for image processing applications is a Markov or Gibbs process. It allows modeling interactions between the objects. The density of the process is defined by an energy form of the sum of potential over interacting object (cliques):

$$f(x) = \frac{1}{Z} exp[-U(x)] \quad (3)$$

where [12]

$$U(x) = \left(V_0 + \sum_{x_i \in \mathbf{x}} V_1(x) + \sum_{\{x_i, x_j\} \in x} V_2(x_i, x_j) + \ldots\right) \quad (4)$$

and Z is the partition function (normalizing constant). Minimization of this energy corresponds to the correct configuration detection. This energy takes into account the interactions between geometric objects U_p (prior energy) and a data energy U_d to fit the configuration to the image:

$$U(x) = U_d(x) + \gamma_p U_p(x)$$

where γ_p is the weight assigned to the prior term; it can be estimated as in [13].
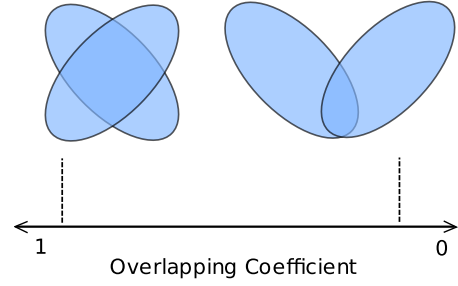

Fig. 1: The overlapping coefficient between two objects

B. Prior

The possibility to introduce prior information is a major advantage of the MPP. This regularizes the configuration to match the real objects taking into considerations imperfection due to *e.g.* image resolution. Since our objects in reality should almost not overlap, we penalize this situation. Let $\mathcal{A}(\omega_i, \omega_j) \in [0, 1]$ represents the overlapping coefficient between two objects, defined as the normalized area of intersection, as shown in figure 1 and proposed by [10]:

$$\mathcal{A}(\omega_i, \omega_j) = \frac{A(\omega_i \cap \omega_j)}{\min(A(\omega_i), A(\omega_j))}, \quad (5)$$

where $A(\omega_i)$ is the area of object ω_i. Let us consider a clique $\{\omega_i, \omega_j\}$, the prior energy of this local configuration is given by:

$$u_p(\omega) = \begin{cases} 0 & \text{if } \mathcal{A}(\omega_i, \omega_j) < 0.1 \\ \infty & \text{if } \mathcal{A}(\omega_i, \omega_j) \geq 0.1, \end{cases} \quad (6)$$

which means that we completely forbid a configuration with overlapping coefficient greater than 10%. The total prior energy of the configuration is then given by:

$$U_p(\omega_i, \omega_j) = \sum_{\omega_i \sim \omega_j} u_p(\omega_i, \omega_j),$$

where $\sim$ is a symmetric relation $\sim_r \in K \times M$.

C. Data term

Given the independence of the data term of each object, data term energy of a configuration ω is given by:

$$U_d(\omega) = \sum_{\omega_i \in \omega} u_d(\omega_i) \quad (7)$$

The term $u_d(\omega_i)$ is the output of a local filter, evaluating from a data point of view, the relevance of object ω_i. The object contains information on both its location and its shape. The data term can thus be interpreted as an adaptive local filter by selecting or favoring a specific shape and object depending locally on the data. For the selected flamingo example, as presented in figure 2, each flamingo can be modeled as a bright ellipse surrounded by a darker background. For an
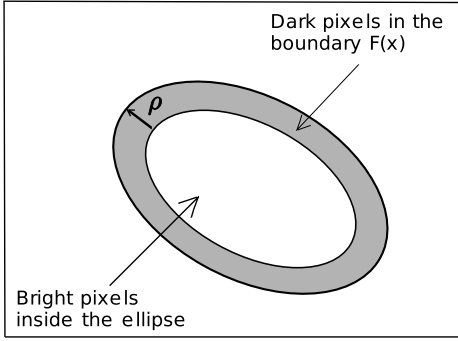
Fig. 2: Modeling flamingo by an ellipse. Defining a background around it to measure the relevance of the proposed object

object $\omega_i = (x_i, m_i)$, with marks $m_i = (a, b, \theta)$, we define the boundary $\mathcal{F}(\omega_i)$ as a subset of K, between ω_i border and a concentric one ω_i', with marks $m_i' = (a + \rho, b + \rho, \theta)$. This boundary represents the background and we want to evaluate the contrast between the ellipse interior and the background. To evaluate the distance $d_B(\omega_i, \mathcal{F}(\omega_i))$, let the interior of the ellipse and its background have a Gaussian distribution with parameters (μ_1, σ_1) and (μ_2, σ_2) respectively. We compute a modified Bhattacharya distance between them as follows [10]:

$$d_B(\omega_i, \mathcal{F}(\omega_i)) = \frac{(\mu_1 - \mu_2)^2}{4\sqrt{\sigma_1^2 + \sigma_2^2}} - \frac{1}{2} log \frac{2\sigma_1 \sigma_2}{\sigma_1^2 + \sigma_2^2}.$$

The data energy $u_d(\omega_i)$ associated to object ω_i is given by:

$$u_d(\omega_i) = \mathcal{Q}_d(d_B(\omega_i), \mathcal{F}(\omega_i)),$$

where $\mathcal{Q}_d(d_B) \in [-1, 1]$ is a quality function which gives large values to small distances (weakly contrasted object) and small values (well contrasted) otherwise [10].

$$\mathcal{Q}_d(d_B) = \begin{cases} (1 - \frac{d_B}{d_0}) & \text{if } d_B < d_0 \\ exp(-\frac{d_B - d_0}{D}) - 1 & \text{if } d_B \geq d_0, \end{cases}$$

where D is a scale parameter calibrated to 100 and d_0 is estimated either for the whole image or for each region, as detailed in [10].

III. OPTIMIZATION ALGORITHMS

Once the density is defined, the issue now is to sample it to obtain the corresponding realization. No direct simulation is possible due to the normalizing constant. Reversible Jump Markov Chain Monte Carlo method has being widely used for MPP in image processing [4] [5] because of its flexibility and speed of convergence compared to previous sampler (birth and death [2]). The major drawback of this algorithm is being based on proposing *local perturbations* (involving one or two objects[1]) in the scene, which can be rejected [7], which limits its convergence speed.

[1]For merging two objects into one.

A. Multiple Birth and Death

The Multiple Birth and Death (MBD) has been recently proposed making possible *multiple perturbations in parallel* [8]. The main idea is that at each iteration n of the algorithm, given the current configuration $\omega_{(n)}$, we propose the addition of a new configuration ω' (multiple objects) and we treat the new configuration $\omega = \omega_{(n)} \cup \omega'$ by removing non-fitting objects with associated continuous probability assuming the convergence to the right distribution. This method performs the sampling of the process by considering a Markov chain consisting of a discrete time multiple birth-and-death process describing all possible transitions from the configuration $\omega_{(n)}$ to the elements of Ω. The authors in [8] demonstrated that this Markov chain can be considered as an approximation of a continuous-time reversible process and converge to it, which, in a simulated annealing scheme, guarantees weak convergence to the measure concentrated on the global minimum of the energy function.

Algorithm 1 Multiple Birth and Death

1: $n \leftarrow 0$, $\omega_{(0)} \leftarrow \emptyset$
2: $\delta = \delta_{(0)}$, $\beta = \beta_{(0)}$
3: **repeat**
4:     Birth: generate ω', a realization of a Poisson process of intensity δ
5:     $\omega \leftarrow \omega_{(n)} \cup \omega'$
6:     Death: For each $\omega_i \in \omega$, calculate the death probability $d(\omega_i) = \frac{\delta a_\beta(\omega_i)}{1 + \delta a_\beta(\omega_i)}$, where $a_\beta(\omega_i) = e^{-\beta(U(\omega \setminus \omega_i) - U(\omega))}$
7: **until** Convergence, if not converged, set $\omega_{(n+1)} = \omega$, $n \leftarrow n + 1$, $\delta_{(n+1)} = \delta_{(n)} \times \alpha_\delta$, $\beta_{(n+1)} = \beta_{(n)} \times \alpha_\beta$, and go to "Birth"

We only consider here the discrete case of the MBD algorithm, summarized in the above algorithm 1. Let δ be the intensity of the process; first we initialize the algorithm (step 1 and 2), by setting the starting values for δ and β (inverse temperature) used for the simulated annealing scheme. The iteration starts in step 3 till step 6, the algorithm keeps iterating until convergence. At iteration n, a configuration ω is transformed into $\omega'' = \omega_1 \cup \omega_2$, where $\omega_1 \subseteq \omega$ and ω_2 is a configuration such that $\omega_1 \cap \omega_2 = \emptyset$.

The transition associated with the birth of an object in a small volume $\Delta v \subset K$ is given by:

$$q_\delta(v) = \begin{cases} \lambda \Delta v \delta & \text{if } \omega \leftarrow \omega \setminus \omega_i \\ 1 - \lambda \Delta v \delta & \text{if } \omega \leftarrow \omega \ (\omega_i \ is \ kept), \end{cases}$$

This transition is simulated by generating ω', a realization of a Poisson process of intensity δ. The death transition probability of an object ω_i from the configuration $\omega \cup \omega'$ is given by:

$$p_\delta(\omega_i) = \begin{cases} \frac{\delta e^{\beta E(\omega_i, \omega \setminus \omega_i)}}{1 + \delta e^{\beta E(\omega_i, \omega \setminus \omega_i)}} = \frac{\delta a_\beta(\omega_i)}{1 + \delta a(\omega_i} & \text{if } \omega \leftarrow \omega \setminus \omega_i \\ \frac{1}{1 + \delta a_\beta(\omega_i)} & \text{if } \omega \leftarrow \omega \ (\text{object } \omega_i \text{ is killed}), \end{cases}$$

resulting in $\omega'' = \omega_1 \cup \omega_2$, where $\omega_1 \subseteq \omega$ and $\omega_2 \subseteq \omega'$

B. Graph Cuts

In the last few years, a new approach of energy minimization based on graph cuts has emerged in computer vision. Graph cuts efficiently solved the optimization problem of certain energy families by finding either a global or a local minimum with a very high speed of convergence. This technique is based on a special graph construction from the energy function to be minimized. Finding the minimum cut of this graph also minimizes the energy. The minimum cut is efficiently calculated using the max flow algorithm.

The use of graph cuts in computer vision was first introduced [14]. They demonstrated how the Maximum a Posteriori (MAP) estimate of a binary MRF can be exactly calculated using the maximum flow algorithm. Then, it was extended to MRF with multiple labels [15] [16].

This method has been extensively used to compute the MAP solution for a large number of applications for discrete pixel labeling. It has been applied to image segmentation using geometric cues [17] and using regional cues based on Gaussian mixture models [18], video segmentation [19] taking advantage of the redundancy between video frames (dynamic graph cuts), image restoration [20], stereo vision [21] [22], . . .

1) Introduction to Graph Cuts: Many computer vision problems can be formulated as energy minimization problems. Energy minimization to solve the pixels labeling problem (segmentation) can be represented as follows: given an input set of pixels $\mathcal{P} = \{p_1, \ldots, p_n\}$ and a set of labels $\mathcal{L} = \{l_1, \ldots, l_m\}$, the goal is to find a labeling $f : \mathcal{P} \to \mathcal{L}$ which minimizes some energy function. We are interested by binary labeling, where $\mathcal{L} = \{0, 1\}$. A standard form of the energy function is [23]:

$$E(f) = \sum_{p \in \mathcal{P}} D_p(f_p) + \sum_{p,q \in \mathcal{N}} V_{p,q}(f_p, f_q), \quad (8)$$

where $\mathcal{N}$ represents the pixel neighborhood. $D_p(f_p)$ is a function based on the observed data, it gives the cost of assigning the label f_p to pixel p. $V_{p,q}(f_p, f_q)$ gives the cost of assigning labels (f_p, f_q) to pixels (p, q), where (p, q) are neighbors. $D_p(f_p)$ is always referred to as the data term and $V_{p,q}(f_p, f_q)$ as the smoothness or prior term.

2) Graph Cuts: Let $G = (V, E, C)$ be a directed graph which consists of a finite set V of vertices, a set $E \subset V^2$ of edges and a cost function $C : E \to \mathbb{R}^+ \cup \{0\}$. This graph has two special vertices, the source $\mathcal{S}$ and the sink $\mathcal{T}$, also called terminal nodes. An S-T cut is a partition (S,T) of the vertices ($S \cup T = V$ and $S \cap T = \emptyset$), such that $\mathcal{S} \in S$ and $\mathcal{T} \in T$. The cost of the S-T cut $C(S, T)$ is the sum of all costs of all edges that go from S to T:

$$C(S, T) = \sum_{u \in S, v \in T:(u,v) \in E} C(u, v).$$

A minimal cut of the graph G, is a cut whose cost is minimal. In general it is an NP hard problem. This problem is equivalent to the maximum flow from source to sink. Many algorithms have been proposed to solve this problem based on Ford and Fulkerson theorem or based on the Push-Relabel algorithm. In this paper we use the Ford and Fulkerson theorem, it finds the

TABLE I: Data Term

f_s	$D_s(f_s)$	Configuration
$f_s = 0$	$u_d(\omega_i)$	$\omega_i \in \omega_{(n)}$
$f_s = 1$	$1 - u_d(\omega_i)$	$\omega_i \in \omega_{(n)}$
$f_s = 1$	$1 - u_d(\omega_i)$	$\omega_i \in \omega'$
$f_s = 0$	$u_d(\omega_i)$	$\omega_i \in \omega'$

TABLE II: Prior Term

(f_s, f_r)	$V_{sr}(f_s, f_r)$
(0,0)	0
(0,1)	∞
(1,0)	0
(1,1)	0

problem solution in polynomial time with small constants [24]. For a S-T cut, and a labeling f which maps from V to $\{0, 1\}$ for a binary partitioning, $f(v) = 0$ means that $v \in S$ and $f(v) = 1$ means that $v \in T$.

3) Energy minimization using Graph Cuts: In [25], the authors explained which class of energy function can be minimized by graph cuts. One important result from this paper is the regularity condition which must be satisfied, it is a necessary and sufficient condition. This condition represents the homogeneity of the labeling. For a two neighbor pixel configuration (i, j) for which we assign labels $\{0, 1\}$, the condition is [23]:

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0), \quad (9)$$

which states that the energy required (cost) for assigning the same label to neighbor pixels should be less than or equal to the energy for assigning them different labels.

As stated in [14], it is possible to compute the global minimum of the energy for the binary case. This has been generalized to multi-labels under condition on $V(.,.)$ (distance) [16] [26].

C. Multiple Birth and Cut

The main contribution of this paper is the introduction of a new optimization algorithm, the MBC to minimize the MPP energy given by (3). Although the MBD algorithm has been proved to converge to a global minimum and has a good convergence speed, it still has parameters to be tuned which are the intensity of the field, how it decreases, the temperature of the simulated annealing and its scheduling. Wrong selection of those parameters will prevent from proper convergence.

In [27], the authors presented an interesting graph modeling for a mosaic problem. The main goal of their work was to simulate classic mosaics from digital images. For good visual appearance, mosaic should satisfy some constraints such as non-overlapping. They generate a set of candidate layers containing tiles respecting their constraints and they stitch them in an iterative way. In the stitching process, the selection between tiles of current layer and new candidate layer was solved by a graph cut algorithm.

We generalize this idea to the optimization of MPP. The MBC algorithm is described using figure 3 and summarized in algorithm 2.

Initialization: In step (1) of the algorithm we initialize our unique variable R, the number of objects to be added in each proposed configuration[2]. In step (2), we generate a candidate configuration ω' which we set to $\omega_{(0)}$ as an initial configuration. ω' is a sample of non-overlapping ellipses[3] [28]. $\omega_{(0)}$ is represented in figure 3.(a) in green, $\omega_{(0)} = \{a, b, c\}$. Now the algorithm iterations start.

Birth: In the birth step we propose a new configuration ω', *e.g.* $\omega' = \{d, e, f, g\}$ "non-overlapping" ellipses, which are shown in figure 3.(a) in blue. Note that objects $\{d, e\}$ have an overlapping of less than our defined threshold (10%), so they are considered as non-overlapping, as stated in (6).

Cut:

- Graph construction: In the cut step, a graph is constructed for $\omega = \omega_{(n)} \cup \omega'$ as shown in figure 3.(b), *each node represents an object* (ω_i), contrary to most graph cut problems where each node represents one pixel[4]. Edges weights are assigned as show in table I and II. Between each object and the source, for $\omega_i \in \omega_{(n)}$ the weight to the source is the data term $u_d(\omega_i)$ and $1 - u_d(\omega_i)$ to the sink, while it is the inverse for $\omega_i \in \omega'$, it is $1 - u_d(\omega_i)$ to the source and $u_d(\omega_i)$ to the sink. For the edges between objects, we assign it the prior term, ∞ if they are connected, otherwise it is zero.

- Optimizing: To this graph, we apply the graph cut algorithm, to assign labels $\{0, 1\}$. The *key element* to satisfy the *regularity* condition stated in (9), is that the labeling (generated by the graph cut optimization) is differently interpreted for the current configuration $\omega_{(n)}$ and the newly proposed one ω'. Label '1' for $\omega_i \in \omega_{(n)}$ means 'keep' this object, label '0' means 'kill' (remove) it while label '1' for $\omega_i \in \omega'$ means 'kill' this object and label '0' means to 'keep' this object.

Algorithm 2 Multitple Birth and Cut

1: $n \leftarrow 0$, $R \leftarrow const$
2: generate ω' , $\omega_{(0)} \leftarrow \omega'$
3: **repeat**
4:     Birth: generate ω'
5:     $\omega \leftarrow \omega_{(n)} \cup \omega'$
6:     Cut: optimize with graph cuts
7: **until** converged

---

1) Energy comparison: For the graph cut algorithm, edge weights have to be non negative, so we normalize the data term $\mathcal{Q}_d(d_B) \in [0, 1]$. For each ω_i, its data term becomes:

$$u_d^{GC}(\omega_i) = \frac{1 + u_d(\omega_i)}{2} \qquad (10)$$

[2]This parameter can easily be set, we used to set it to one fifth of the expected population size. Different initializations just affect the speed of convergence but not the detection results.

[3]We generate a random object (point and mark), it is rejected if it intersects at least one of the existing ellipses, otherwise it is kept.

[4]In a standard graph cut binary image restoration problem, for an image of size N^2, the required graph is of size N^2 (number of nodes). For a MPP problem, for an image of size N^2, the size of the graph is M (number of objects), where $M \ll N$.
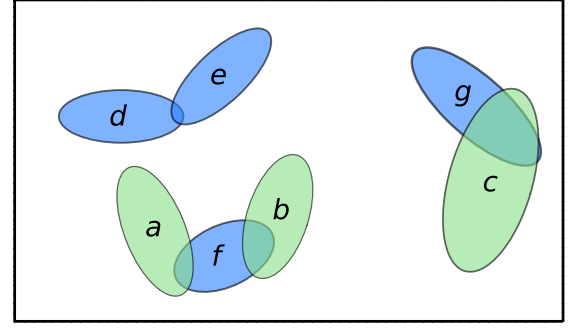
(a)
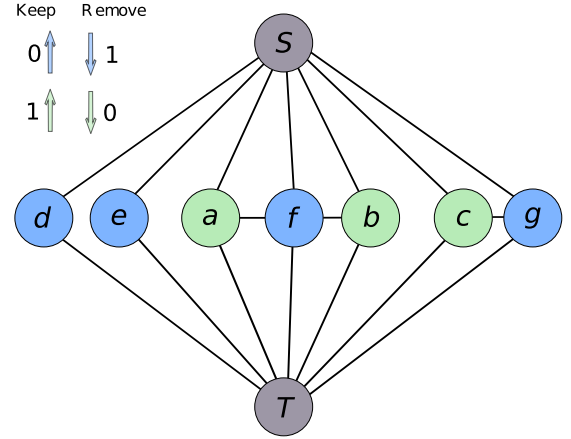
(b)

Fig. 3: (a) Image contaning a current configuration $\omega_{(n)}$ in green and a candidate configuration ω' in blue. (b) The special graph constructed for $\omega_{(n)} \cup \omega'$.

In the sequel, we verify that after this modification to the data term (10), we minimize the same energy using graph cut at each iteration. Let U^{CG} be the energy given by the graph cut, let $\omega = \{\omega_{(n)}, \omega'\}$ where $\omega_{(n)} = \{\omega_1, \dots, \omega_p\}$ and $\omega' = \{\omega_{p+1}, \dots, \omega_q\}$. The energy of the whole graph is the sum of the data term edges and prior term edges:

$$U^{GC}(\omega) = U_d^{GC}(\omega) + U_p^{GC}(\omega),$$

where the data term is given by:

$$U_d^{GC}(\omega) =$$
$$\sum_{i \in <1,p>} \left[\left(\frac{1 + u_d(\omega_i)}{2} \right) \delta_{f(\omega_i)=0} + \left(\frac{1 - u_d(\omega_i)}{2} \right) \delta_{f(\omega_i)=1} \right]$$
$$+ \sum_{i \in <p+1,q>} \left[\left(\frac{1 - u_d(\omega_i)}{2} \right) \delta_{f(\omega_i)=0} + \left(\frac{1 + u_d(\omega_i)}{2} \right) \delta_{f(\omega_i)=1} \right],$$

and the prior term is given by:

$$U_p^{GC}(\omega) = \sum_{\substack{i \in <1,p>, \\ j \in <p+1,q>}} u_p(\omega_i, \omega_j) \delta_{f(\omega_i)=0} \delta_{f(\omega_j)=1}.$$

$u_p(\omega_i, \omega_j)$ is defined as in table II, then $U_p^{GC}(\omega) = U_p(\omega)$. The graph cut energy for the data term is given by:

$$U_d^{GC}(\omega) = \sum_K \left(\frac{1 + u_d(\omega_i)}{2} \right) + \sum_D \left(\frac{1 - u_d(\omega_i)}{2} \right)$$

$$= \sum_K u_d(\omega_i) + \sum_{K+D} \left(\frac{1 - u_d(\omega_i)}{2} \right)$$

$$= U_d(\omega) + \sum_{K+D} \left(\frac{1 - u_d(\omega_i)}{2} \right)$$

$$= U_d(\omega) + \mathcal{K}(\omega)$$

where after optimization, K is the set of objects that we keep and D is the set of objects that we kill. So minimizing $U_d^{GC}(\omega)$ is equivalent to minimizing $U_d(\omega)$ plus a constant $\mathcal{K}(\omega)$, function of the configuration. It becomes:

$$\underset{\omega}{\arg\min}\, U_{GC}(\omega) = \underset{S}{\arg\min}\, U(\omega),$$

where $S = \{u \in \Omega | u \subset \omega\}$.

2) Convergence: The algorithm keeps iterating until convergence. Convergence can be evaluated either by monitoring the number of objects or the energy of the configuration and when it becomes stable, we consider that the algorithm has converged.

Using graph cut, we obtain the global minimum for a configuration $\omega = \omega_{(n)} \cup \omega'$ at each iteration. Let the energy of the configuration ω at the nth iteration be $U^{[n]}(\omega)$, $U^{[n]}(\omega) \leq U^{[n-1]}(\omega)$, it is monotonically decreasing. The non overlapping prior and the finite size of the image induce that the energy is bounded from below. Therefore, we have a sufficient condition for our algorithm to converge at least to a local minimum.

IV. RESULTS

In this section we present results of flamingo detection from aerial Color Infrared Images (CIR) comparing our new algorithm to the MBD algorithm. First we presents results on four different colonies. In table III, our data is composed of two or three samples from each of the four colonies photos. We show the percentages of correct detection of flamingos, negative false and positive false. These results are validated by ecologists[5]. Result in table III shows that the newly proposed algorithm outperforms the MBD algorithm for the detection. MBC has always a higher detection rate, lower negative and positive rates for the majority of the samples.

Secondly, we present the energy evolution during the optimization of both algorithms on a sample from the whole colony. In figure 4.(a), we present the photo of a whole colony, and a rectangular sample, and in figure 4.(b), we highlight the result for this sample. In the curve shown in figure 5, we present the evolution of the energy of the process with respect to time for both MBC and MBD algorithms on the same sample. MBD algorithm is currently faster than our algorithm, for three main reasons:

[5]Ecologists from Tour du Valat.

TABLE III: Comparison between MBC and MBD

Image	Qualifiers	MBC	MBD
Fang02 sample 1	Good detection	0.93	0.87
	Neg. false	0.07	0.13
	Pos. false	0.16	0.09
Fang02 sample 2	Good detection	0.98	0.96
	Neg. false	0.02	0.04
	Pos. false	0	2
Fang05 sample 1	Good detection	0.86	0.82
	Neg. false	0.14	0.18
	Pos. false	0.1	0.07
Fang05 sample 2	Good detection	0.97	0.9
	Neg. false	0.03	0.1
	Pos. false	0.08	0.14
Fang05 sample 3	Good detection	0.94	0.9
	Neg. false	0.1	0.1
	Pos. false	0.06	0.14
Tuz04 sample 1	Good detection	1	0.99
	Neg. false	0.0	0.01
	Pos. false	0.04	0.01
Tuz04 sample 2	Good detection	0.98	0.98
	Neg. false	0	0
	Pos. false	0.04	0.04
Tuz04 sample 3	Good detection	1	1
	Neg. false	0	0
	Pos. false	0.02	0
Tuz06 sample 1	Good detection	1	1
	Neg. false	0	0
	Pos. false	0.01	0
Tuz06 sample 2	Good detection	0.98	0.95
	Neg. false	0	0.04
	Pos. false	0.09	0.06
Tuz06 sample 3	Good detection	0.99	0.95
	Neg. false	0.01	0.04
	Pos. false	0.12	0.08

1) During the algorithm iterations, each proposed configuration ω' in the MBD algorithm can be very dense but in our algorithm, it has to respect the non-overlapping constraint.
2) Although the size of the used graph is not large, we construct a new graph at each iteration.
3) The birth map used in MBD is not yet integrated in MBC.

We used the graph-cut code developed by Olga Veksler [23] [29] [30].

V. CONCLUSIONS AND FUTURE WORK

We have presented an efficient optimization algorithm to minimize a highly non-convex energy function which was previously solved within a simulated annealing scheme. We avoid the difficult task of setting the temperature and cooling parameters of the simulated annealing. We showed the quality of the detection on many test samples of four different datasets. The MBC algorithm reaches a lower energy level than the MBD but in longer time. We can reach the same level using MBD, but it requires many trials to set the perfect parameters values.
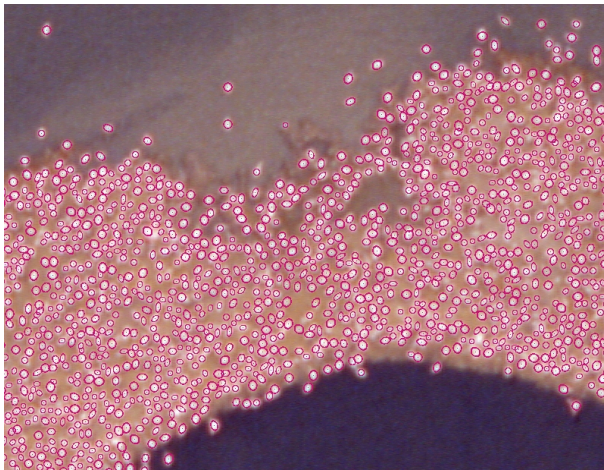
We are currently studying in details the minimum of energy we obtain with the proposed algorithm, to see if calculating
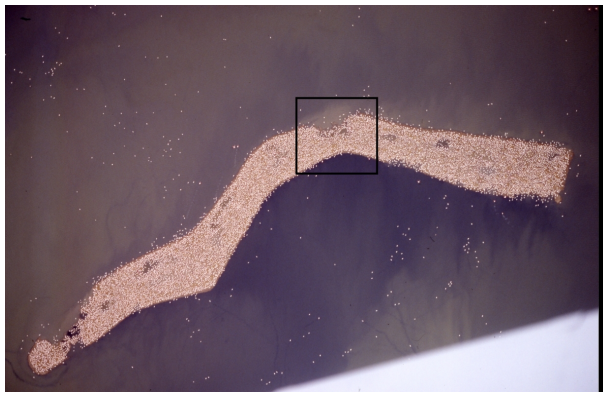
(a)

(b)

Fig. 4: (a) An aerial image of a full colony, with a highlited rectangle ©Tour du Valat. (b) Showing the detection result in the highlighted rectangle by surrounding each flamingo with an ellipse ©Ariana/INRIA.

Fig. 5: Energy evolution of the configuration during the optimization with respect to time.

the global minimum for successive subsets of the configuration space give the global minimum or not. We are also investigating the possibility of treating those subsets independently for later parallelization of the algorithm.

ACKNOWLEDGMENT

REFERENCES

[1] A. J. Baddeley and M. N. M. V. Lieshout, *Stochastic geometry models in high-level vision*. Journal of Applied Statistics, 1993, vol. 20.

[2] C. J. Geyer and J. Moller, "Simulation and likelihood inference for spatial point processes," *Scandinavian Journal of Statistics*, vol. 21, pp. 359–373, 1994.

[3] M. N. M. V. Lieshout, *Markov point processes and their applications*, I. C. Press, Ed. World Scientific Publishing Company, 2000.

[4] R. Stoica, X. Descombes, and J. Zerubia, "A Gibbs point process for road extraction in remotely sensed images," *International Journal of Computer Vision (IJCV)*, vol. 57, no. 2, pp. 121–136, 2004.
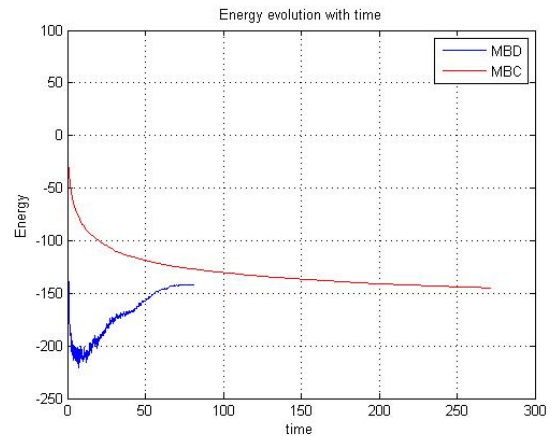
[5] G. Perrin, X. Descombes, and J. Zerubia, "2D and 3D vegetation resource parameters assessment using marked point processes," in *Proc. International Conference on Pattern Recognition (ICPR)*, Hong-Kong, August 2006.

[6] W. Ge and R. Collins, "Marked point processes for crowd counting," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, Miami, USA, July 2009.

[7] C. J. Geyer, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, vol. 82, pp. 711–732, 1995.

[8] X. Descombes, R. Minlos, and E. Zhizhina, "Object extraction using a stochastic birth-and-death dynamics in continuum," *Journal of Mathematical Imaging and Vision*, vol. 33, pp. 347–359, 2009.

[9] M. Ortner, X. Descombes, and J. Zerubia, "Building outline extraction from digital elevation models using marked point processes," *International Journal of Computer Vision (IJCV)*, vol. 72, no. 2, pp. 107–132, April 2007.

[10] S. Descamps, X. Descombes, A. Béchet, and J. Zerubia, "Automatic flamingo detection using a multiple birth and death process," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, USA, March 2008.

[11] M. S. Kulikova, I. H. Jermyn, X. Descombes, E. Zhizhina, and J. Zerubia, "Extraction of arbitrarily shaped objects using stochastic multiple birth-and-death dynamics and active contours," in *Proc. IS&T/SPIE Electronic Imaging*, San Jose, USA, January 2010.

[12] A. Baddeley and R. Turner, "Modelling spatial point patterns in r," in *Case Studies in Spatial Point Pattern Modelling. Lecture Notes in Statistics 185, 2374*. Springer, 2006.

[13] F. Chatelain, X. Descombes, and J. Zerubia, "Parameter estimation for marked point processes. application to object extraction from remote sensing images." in *Proc. Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, Bonn, Germany, August 2009.

[14] D. M. Greig, B. T. Porteous, and A. H. Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 51, no. 2, pp. 271–279, 1989. [Online]. Available: http://www.jstor.org/stable/2345609

[15] O. V. Yuri Boykov and R. Zabih, "Markov random fields with efficient approximations," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998, pp. 648–655.

[16] H. Ishikawa, "Exact optimization for Markov random fields with convex priors," *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 25, pp. 1333–1336, 2003.

[17] Y. Boykov, "Computing geodesics and minimal surfaces via graph cuts," in *Proc. International Conference on Computer Vision (ICCV)*, 2003, pp. 26–33.

[18] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive gmmrf model," in *Proc. European Conference on Computer Vision (ECCV)*, Prague, May 2004.

[19] P. Kohli and P. H. S. Torr, "Dynamic graph cuts for efficient inference in Markov random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 12, pp. 2079–208, 2007.

[20] H. Ishikawa and D. Geiger, "Mapping image restoration to a graph problem," in *Proc. of IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Antalya, Turkey, June 1999.

[21] S. Roy and I. J. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem," *IEEE International Conference on Computer Vision (ICCV)*, vol. 0, p. 492, 1998.

[22] J. Kim, V. Kolmogorov, and R. Zabih, "Visual correspondence using energy minimization and mutual information," in *Proc. International Conference on Computer Vision (ICCV)*, ser. 2, October 2003, pp. 1033–1040.

[23] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 26, no. 2, pp. 147–159, February 2004.

[24] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *J. ACM*, vol. 35, no. 4, pp. 921–940, 1988.

[25] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 26, no. 2, pp. 65–81, 2004.

[26] H. Ishikawa and D. Geiger, "Occlusions, discontinuities, and epipolar lines in stereo," in *Proc. European Conference on Computer Vision (ECCV)*, Freiburg, Germany, June 1998, pp. 232–248.

[27] O. V. Yu Liu and O. Juan, "Simulating classic mosaics with graph cuts," in *Proc. Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2007, pp. 55–70.

[28] D. Stoyan and H. Stoyan, *Fractals, Random Shapes and Point Fields: Methods of Geometrical Statistics*. J. Wiley & Son, 1995.

[29] O. V. Y. Boykov and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 11, pp. 1222–1239, November 2002.

[30] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 26, no. 9, pp. 1124–1137, September 2004.