Multiple Birth and Cut Algorithm for Multiple Object Detection

Ahmed Gamal-Eldin, Xavier Descombes, Guillaume Charpiat, Josiane Zerubia
INRIA Sophia-Antipolis Méditerannée
2004 route des Lucioles, BP 93
06902 Sophia-Antipolis
Cedex, France
{ahmed.gamal eldin, xavier.descombes, guillaume.charpiat, 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 energy minimization of an object based model, the marked point process. We compare the MBC to the MBD showing their respective advantages and drawbacks, where the most important advantage of the MBC is the reduction of number of parameters. We demonstrate that by proposing good candidates throughout the selection phase in the birth step, the speed of convergence is increased. In this selection phase, the best candidates are chosen from object sets by a belief propagation algorithm. We validate our algorithm on the flamingo counting problem in a colony and demonstrate that our algorithm outperforms the MBD algorithm.*

1. Introduction

Automatic detection and counting of multiple objects is a problem of major importance, that finds applications in different domains such as evaluation of populations 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 images. This framework was first introduced in [1]. The MPP modeling is based on defining a configuration space composed of object sets, to which a Gibbs energy function is attached. The detection is then obtained by minimizing this energy.

Initially, samplers of similar models like Markov random fields (MRF), either stochastic or deterministic, were based only on standard moves within the framework of Metropolis Hasting dynamics, where only one pixel changes at a time. During the last decade, multiple move methods emerged and most of them are based on graph cut techniques [2].

Point process samplers have also evolved from simple perturbations (standard moves) as in birth and death algorithm, where at each iteration, one object is either added to or removed from the current configuration [3], [4]. Such algorithms are extremely slow in image processing. Therefore, the Reverse Jump Markov Chain Monte Carlo algorithm [5] has been widely used for MPP in image processing [6], [7], [8] due to its flexibility, especially when using updating schemes such as Metropolis Hasting [5]. On the birth step, we now add moves such as split, translate, rotate, etc. The main limitation is that it still treats one or two objects at a time and has a rejection rate. Later, the Multiple Birth and Death algorithm (MBD) was proposed allowing multiple perturbations [9].

The main contribution of this paper is the development of a new multiple perturbation optimization technique, named Multiple Birth and Cut (MBC) [10]. It combines ideas from MBD and the popular graph cut algorithm. The MBC's algorithm major advantage is the severely reduced number of parameters as this algorithm does not involve the simulated annealing scheme. We propose an iterative algorithm to explore the configuration space. We choose between current objects and newly proposed ones using binary graph cuts. In the second part of this work, we propose some modifications to this first algorithm in order to increase its speed of convergence. The algorithm starts by proposing a dense configuration of objects from which the best candidates are selected using the belief propagation algorithm. Next, this candidate configuration is combined with the current configuration using the binary graph cut.

This paper is organized as follows: we start by briefly introducing the MPP model in section II. In section III, we describe optimization algorithms, multiple birth and death algorithm for MPP model optimization, then the Graph-Cut algorithm. We then describe the new proposed algorithms, followed by energy comparison of the different algorithms in section IV. In section V we show detection results, a comparison between the proposed algorithms and MBD. Conclusions are drawn in section VI.

2. Marked Point Process

Marked point process framework is suitable for defining 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 the definition of a prior on the object distribution in the scene. This framework can be interpreted as a generalization of the MRF theory, where the number of random variables is unknown. Moreover, an object is associated to each variable, on which geometric constraints can be modeled.

2.1 Point Process
Definition. Point processes are mathematical models for irregular or random point patterns. Let X denote a point process living on $K = [0, I_{max}]$ $[0, J_{max}]$. X is a measurable mapping from a probability space ($\Upsilon, \mathcal{A}, \mathcal{P}$) to the set of unordered configurations of points in K. K is a closed, connected subset of $\mathbb{R}^2$. This mapping defines a point process. Intuitively, a point process is a random variable whose realizations are random point configurations.

2.1.1 Poisson Point Process.
Poisson point process is the simplest process, from which more complex models can be built. It often serves as a reference model for complete spatial randomness. Let's consider a Poisson point process defined on K and specified by the intensity function $\rho : K \rightarrow [0, \infty)$. We only consider simple point process[1] $x = \{x_1, ..., x_n\}$ is defined in the space defined by the locally finite point configurations N_{lf}, which implies that $n(x_B) \leq \infty$ where n() is a counting function and $x_B = x \cap B$, where B is a bounded closed set. The intensity measure λ of the Poisson process is given by:

$$\lambda(B) = \int_B \rho(\xi) d(\xi), \quad B \subseteq K.$$

for $\xi \in S$. The Poisson probability measure on all $B \subset N_{lf}$ with intensity measure $\lambda(.)$ on K, where $\lambda(K) < \infty$, is given by:

$$\pi(B) = e^{-\lambda(K)} \left(1_{\emptyset \in B} + \sum_{n=1}^{\infty} \frac{\pi_n(B)}{n!} \right)$$

where

$$\pi_n(B) = \int_B \cdots \int_B 1\left[\{x_1, ..., x_n\} \in B \right] \left(\prod_{i=1}^{n} \rho(x_i) \right) dx_1 ... dx_n$$

The probability of having exactly n points is given by [4]:

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

where n is the number of points in the configuration.

2.1.2 Marked Point Process.

Point processes were introduced in image processing since they easily allow modeling scenes made of multiple objects. Objects can have simple or complex shapes: simple shapes like lines for road detection [6], rectangles for buildings [11], ellipses for trees [7] and flamingos [12]; complex shapes, using active contours for complex forms, like tree crowns [13]. In this paper for instance, each flamingo is modeled by an ellipse. Let M be the mark space, $M = [a_{min}, a_{max}] \times [b_{min}, b_{max}]$ $[0, \pi[$, where a and b are the length of the major and the minor axes respectively, for which we define a minimum and a maximum value[2], and $\theta \in [0, \pi[$ is the orientation of the ellipse. The geometry of the shape is represented by the mark m_i associated to each point x_i. Therefore, an object is defined as $\omega_i = (x_i, m_i) \in K \times M$.

$$\Omega = \bigcup_{n=0}^{\infty} \Omega_n, \quad \Omega_n = \left\{\left\{w_1, \dots w_n\right\}, w_i \subset K \times M\right\}, \tag{1}$$

where Ω_n is the subset of configurations containing exactly n objects, and $\omega = \{\omega_i, i = 1,\dots, n\}$. We define a reference measure as the product of the Poisson measure $v(x)$ on Υ and the Lebesgue measures on the mark space:

$$d\pi_\Upsilon(w) = dv(x) \prod_{i=1}^{n}\left(d\mu(m_i)\right)$$

The MPP is then defined by a density with respect to this measure:

$$d\pi(w) = f(w) d\pi_\Upsilon(w) \tag{2}$$

2.1.3 Markov Point Process.

Among MPP, Markov (or Gibbs) point process are of particular interest for applications in object detection. It allows modeling interactions between the objects. The density of the process is then written as the sum of potentials over interacting object (cliques):

$$f(w) = \frac{1}{Z}\exp\left[-U(w)\right] \tag{3}$$

where [14]:

$$U(w) = \left(V_0 + \sum_{w_i \in w} V_1(w_i) + \sum_{\{w_i, w_j\} \in w} V_1(w_i, w_j) + \dots\right) \tag{4}$$

Z is the partition function (normalizing constant), and V_k the potentials of order k. Minimizing the energy $U(w)$ corresponds to the target configuration. This energy takes into account the interactions between geometric objects U_p (prior energy) and a data energy U_d to fit the configuration onto the image:

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

where γ_p is the weight assigned to the prior term which can be estimated as in [15].

2.2 Prior

The possibility to introduce prior information is a major advantage of the MPP framework. This regularizes the configuration to match the real objects taking into consideration the image defects, due to, e.g., image resolution or noise. Since our objects (flamingos) in reality should not overlap, we penalize overlapping. Let $\mathcal{A}(\omega_i, \omega_j) \in [0, 1]$ represent the overlapping coefficient between two objects, defined as the normalized area of intersection, as shown in figure 1 and proposed by [12]:

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

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

---

[1]Simple point process means two points of a given configuration must be different.

[2]The min and max values of the major and minor axes are defined based on the image resolution, and knowing that the maximum dimension of a flamingo (from a top-view) is 80cm.

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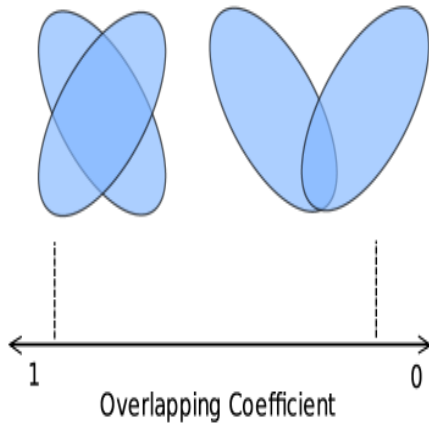

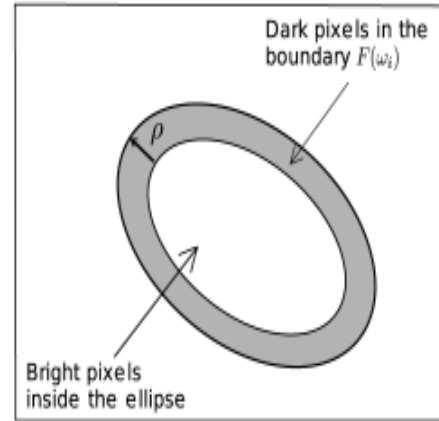Figure 1. The overlapping coefficient between two objects

Figure 2. Ellipse modeling a flamingo and the background around it to measure the relevance of the proposed object

which means that we do not allow a configuration with an overlapping coefficient greater than 10%. The total prior energy of the configuration is then given by:

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

where ~ is a symmetric reflexive relation used to determine the neighborhood of an object, and defined by the intersection of ellipses.

2.3 Data term

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

$$u_d(\omega) = \sum_{\omega_i \in \omega_j} u_d(\omega_i) \qquad (7)$$

The term $u_d(\omega_i)$ is the output of a local filter, evaluating from the 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 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 object $\omega_i = (x_i, m_i)$, with marks $m_i = (a, b, \theta)$, we define the boundary $\mathcal{F}(\omega_i)$ as the subset of K, between the ellipse 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 evaluate the contrast between the ellipse interior and the background. To evaluate the distance $d_B(\omega_i, \mathcal{F}(\omega_i))$, we assume that the interior of the ellipse and its background have a Gaussian distribution with parameters (μ_1, σ_1) and (μ_2, σ_2) respectively, which are estimated from the image. We compute a modified Bhattacharya distance between them as follows [12]:

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

The data energy $u_d(\omega_i)$ associated to object ω_i is then 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 positive values to small distances (weakly contrasted object) and

negative values (well contrasted) otherwise [12]:

$$Q_{d}(d_B) = \begin{cases} \left(1 - \dfrac{d_B}{d_0}\right) & \text{if } d_B < d_0 \\ exp\left(-\dfrac{d_B - d_0}{D}\right) - 1 & \text{if } d_B \geq d_0 \end{cases}$$

3. Optimization Alogrithms

Once the density is defined, the next step is to optimize the energy in order to obtain the corresponding realization. No direct simulation is possible due to the unknown normalizing constant.

3.1 Multiple Birth and Death

The Multiple Birth and Death (MBD) algorithm has been recently proposed making possible multiple perturbations in parallel [9]. The main idea is that, at each iteration n of the algorithm, given the current configuration $\omega_{[n]}$[3], we add a new random configuration ω' (multiple objects) and we treat the new configuration $\omega = \omega_{[n]} \cup \omega$' by removing non-fitting objects with an associated probability which guarantees 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 [9] 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.

Here we only consider the discrete case of the MBD algorithm, summarized in 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, α_δ and α_β are coefficients to decrease the intensity of the process and the temperature respectively. Then the iterations start in step 3 till step 6, the algorithm keeps iterating until convergence. At iteration n, a configuration ω is transformed into ω'' $= \omega^1 \cup \omega^2$, 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:

$$q_\delta(v) = \begin{cases} \Delta v\delta & \text{if } \omega \leftarrow \omega \cup \omega_i \\ 1 - \Delta v\delta & \text{if } \omega \leftarrow \omega \text{ (no birth in } \Delta v) \end{cases}$$

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 $\omega \cup \omega_i$ is given by:

$$p_\delta(\omega_i) = \begin{cases} \dfrac{\delta_{\alpha\beta}(\omega_i)}{1 + \delta_\alpha(\omega_i)} & \text{if } \omega \leftarrow \omega \setminus \omega_i \\ \dfrac{1}{1 + \delta_{\alpha\beta}(\omega_i)} & \text{if } \omega \leftarrow \omega \text{ (}\omega_i \text{ surives)} \end{cases}$$

where $_{\alpha\beta}(\omega_i) = \exp(-\beta [U(\omega \setminus \{\omega_i\}) - U(\omega)])$, resulting in ω'' $= \omega^1 \cup \omega^2$, where $\omega^1 \subseteq \omega$ and $\omega^2 \subseteq \omega$'. This death probability is calculated for every $\omega_i \in \omega$, and the object ω_i is killed (removed) with probability $p_\delta(\omega_i)$.

To speed up the process, we propose two ideas:

• We utilize a birth map which means we consider an inhomogeneous birth rate to favor birth at locations where the data tends to define an object. Thus, we decrease the required number of iterations, while staying uniform over the mark space M.

• The death test is applied on the objects sorted by their data term, starting by objects with bad data term. The order of the death test does not affect the final result, but it has a great impact on the speed of convergence of the algorithm.

3.2 Graph Cut

In the last few years, a new approach of energy minimization based on graph cuts has emerged in computer vision. Graph cuts

---

[3]The subscript [i] indicates the iteration number of the algorithm.

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 is equivalent to minimizing the energy. The minimum cut is efficiently calculated using the max flow algorithm.

The use of graph cuts in computer vision was first introduced in [16]. The authors demonstrated how a Maximum a Posteriori (MAP) estimate of a binary MRF can be exactly calculated using the maximum flow algorithm. Then, it has been extended to MRF with pairwise interactions with multiple labels [17], [18].

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 [19] and using regional cues based on Gaussian mixture models [20], video segmentation [21] taking advantage of the redundancy between video frames (dynamic graph cuts), image restoration [22], stereo vision [23], [24].

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 $a_\beta(\omega_i) = e^{-\beta(U(\omega \backslash \omega_i)-U(\omega))}$
7: **until** Convergence, if not converged, set $\omega_{[n+1]} = \omega$, $\delta_{[n+1]} = \delta_{[n]} \times \alpha_\delta$, $\beta_{[n+1]} = \beta_{[n]} \times \alpha_\beta$, $n \leftarrow n+1$ and go to "Birth"

Here we describe a simple example of graph cut algorithm for solving an image processing problem. Many computer vision problems can be formulated as energy minimization problems. Energy minimization to solve the pixel labeling problem (segmentation) can be represented as follows: given an input set of pixels $\mathcal{P} = \{p_1,..., p_n\}$ and a set of labels $\mathcal{L} = \{l_1,..., l_m\}$, the goal is to find a labeling $f: \mathcal{P} \rightarrow \mathcal{L}$ which minimizes some energy function. We are interested in binary labeling, where $\mathcal{L} = \{0, 1\}$. A standard form of the energy function is [25]:

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

where N represents pixel neighborhoods. $D_p(f_p)$ is a function based on the observed data, it gives the cost of assigning the label fp to pixel p. $V_{p,q}(f_p, f_q)$ is 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.

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 $\rightarrow \mathbb{R}^+ \cup \{0\}$. This graph has two special vertices, the source 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 $S \in$ 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 which finds the problem's solution in polynomial time with small constant [26]. For an **S-T** cut, and a labeling f which maps from V to {0, 1} for a binary partition, $f(v) = 0$ means that $v \in S$ and $f(v) = 1$ means that $v \in T$.

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

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

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 [16], it is possible to compute the global minimum of submodular binary energies. This has been generalized to multi-labels under condition on V (.,.) [18], [27].

3.3 Multiple Birth and Cut
The main contribution of this paper is the introduction of a new optimization algorithm that we call the Multiple Birth and Cut (MBC) algorithm to minimize the energy function of a point process. 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 proper convergence.

In [28], the authors presented an interesting graph model for a mosaic problem. The main goal of their work was to simulate classic mosaics from digital images. For good visual appearance, mosaics should satisfy constraints such as nonoverlapping. They generate a set of candidate layers containing tiles respecting the constraints and they "stitch" them in an iterative way. In the "stitching process", the selection between tiles of the current layer and a new candidate layer is solved by a graph cut algorithm. We generalize this idea to the optimization of the energy function associated to a MPP.

The MBC algorithm is described in the sequel, using figure 3 and summarized in algorithm 2.

Initialization: In step (1) of the algorithm we initialize our unique variable R, which represents the number of objects to be proposed at each iteration. This parameter R can easily be set, we set it to one fifth of the expected population size. Different initializations only affect the speed of convergence but not the detection results. In step (2), we generate a candidate configuration ω' which we set to an initial configuration $\omega_{[0]}$. The set of non-overlapping ellipses ω' is generated as follows: each proposed object !i (with position and mark) is rejected if it intersects at least one of the existing ellipses in the current ω', otherwise it is kept [29]. $\omega_{[0]} = \{a, b, c\}$ is represented in figure 3(a) in green. Now the algorithm starts iterating between the *Birth* and the *Cut* steps until convergence.

Config \ fs	0	1
$\omega_i \in \omega_{[n]}$	$u_d(\omega_i)$	$1 - u_d(\omega_i)$
$\omega_i \in \omega'$	$1 - u_d(\omega_i)$	$u_d(\omega_i)$

Table 1. Data Term

$(f$s$, f$r$)$	Vsr $(f$s$, f$r$)$
(0,0)	0
(0,1)	∞
(1,0)	0
(1,1)	0

Table 2. Prior Term

Birth: In the birth step we propose a new configuration ω' , e.g., $\omega' = \{d, e, f, g\}$ of "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 from $\omega = \omega_{[n]} \cup \omega'$ 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]. Edge weights are assigned as shown in tables I and II. Between each object and the source (t-links), 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 (n-links), we assign the prior term: ∞ if they are neighbor, otherwise it is zero.

•Optimizing: To this graph, we apply the graph cut algorithm, to assign labels f0; 1g. The *key element* to satisfy the *submodularity* condition (equation 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'. Our energy contains indeed supermodular terms, which are made submodular by inverting label interpretations. Label '1' for $\omega_i \in \omega_{[n]}$ stands for 'keep' this object, label '0' stands for 'kill' (remove) this object whereas for $\omega_i \in \omega'$ label '1' for stands for 'kill' this object and label '0' stands for to 'keep' this object.

Based on this labeling interpretation, and on the defined interaction cost from table II, the regularity condition (9) is satisfied since the *l.h.s.* will always be less than the *r.h.s.* which is equal to infinity.

Convergence test: The convergence for this type of models with a highly non-convex function is harder to verify than for convex models with gradient descent algorithms. Usually, we consider that the algorithm has converged if the energy has not decreased for ten successive iterations. The number of objects also can be used in a similar way: if it stays constant for n successive iterations, then the algorithm has converged. An alternative is to use a fixed number of iterations.
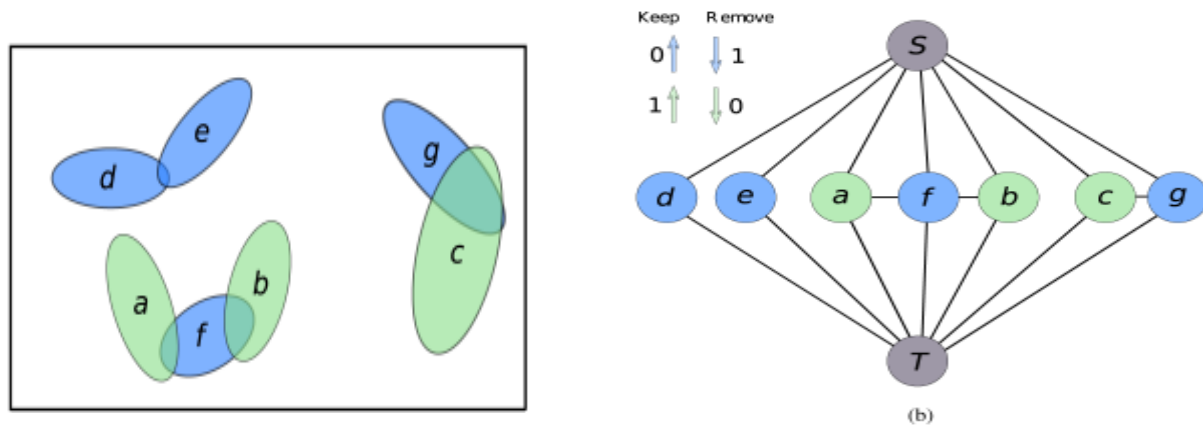

Figure 3. (a) Image containing a current configuration $\omega_{[n]}$ in green and a candidate configuration ω' in blue. (b) The special graph constructed for $\omega_{[n]} \cup \omega'$

Algorithm 2 Multiple 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: $\omega_{[n+1]} \leftarrow Cut(\omega_{[n]} \cup \omega')$ (optimize with graph cuts)

7: **until** converged

---

[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<<N.

Speed of Convergence:
We present the energy evolution during the optimization of both algorithms on a sample from a whole colony of flamingos. In figure 4(a), we present the photo of the whole colony with the selected rectangular sample, and in figure 4(b), we present the detection result on this sample.

Curves shown in figure 5 present the energy evolution of the process with respect to time of both MBC and MBD (two runs) algorithms on the selected sample.

This graph presents the energy evolution of two runs of the MBD algorithm with two sets of parameters. The energy of the first run is presented by the blue curve, with the following parameters: $\Delta\delta = 0.9985$, $\Delta\beta = 0.9975$ and for 4000 iterations, while the second run, presented by the green curve has the following parameters: $\Delta\delta = 0.9995$, $\Delta\beta = 0.9985$, with 10000 iterations. Both MBD runs have the same initial intensity of the process $\delta = 2000$ and temperature $T = \frac{1}{\beta} = 1/50$. The MBC energy evolution curve is presented by the red curve, and its unique parameter is $R = 1000$.

Both MBD runs follow a simulated annealing scheme. They start by random configurations, then the algorithm keeps organizing the configurations until it arrives at a saddle point, where it performs small local refinement and the energy decreases very fast.

We see from the energy evolution in figure 5 that setting the optimal set of parameters for the MBD algorithm for convergence is not trivial. The energy evolution curves show that the second set of parameters gives better results by reaching a lower minimum of energy, lower that the MBC algorithm and even faster. The advantage of the MBC algorithm is that it reaches a similar value of the energy without any parameter tuning, but in a longer time.

Based on the energy evolution curve presented in figure 5, we notice that the MBD algorithm is currently faster than our algorithm, for three main reasons:

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 small, we construct a new graph at each iteration.

3) The birth map is not integrated.

3.4 Optimized MBC using Belief Propagation
From the reasons mentioned above about the speed limitation of the MBC algorithm, we can summarize reason 1 and 2 by saying that the main limitation comes from the quality of the proposed configuration at each iteration. Since the proposed configuration respects the non-overlapping constraint from the beginning, it can not take a real advantage of using the birth map and consequently requires a large number of iterations.

We propose to insert a selection phase in the birth step, which allows adding more relevant objects in the birth step, thus reducing the number of iterations (time to converge). In the sequel we explain the modified MBC algorithm, which is summarized in algorithm 3.

Algorithm 3 Multiple Birth and Cut

1: $n \leftarrow 0$, $R \leftarrow$ const

2: generate ω', $\omega_{[0]} \leftarrow \omega'$

3: **repeat**

4:   Birth: generate Γ

5:   $\omega' \leftarrow$ Select from (Γ)

6:   Cut: $\omega_{[n+1]} \leftarrow \text{Cut}(\omega_{[n]} \cup \omega')$

7: $n \leftarrow n + 1$
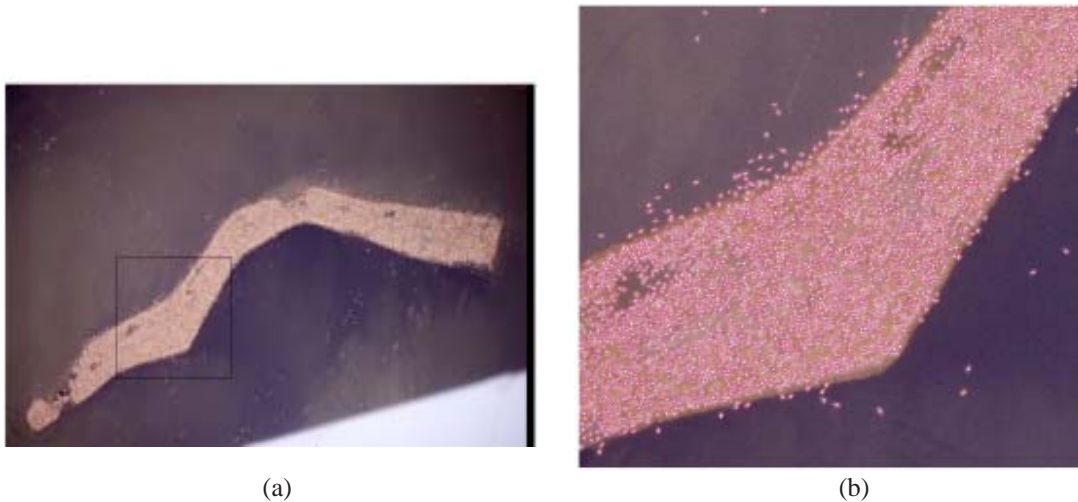
8: **until** converged

Figure 4. (a) An aerial image of a full colony ©Tour du Valat, with a highlighted rectangle. (b) Detection result using MBC algorithm in the highlighted rectangle, where each flamingo is surrounding by an ellipse ©Ariana/INRIA

Selection Phase: In the birth step, the new algorithm generates a dense configuration . This configuration has a special organization, where $\Gamma = \{X_1, X_2, ..., X_n\}$ and $Xi = \{\omega_i^1, \omega_i^2, ..., \omega_i^l\}$. Each X_i encodes l candidates from which only one will be kept, see figure 6(b). The aim of this organization is, instead of proposing a single object ω_i to detect a given object o_j, to propose many objects at a similar location represented by X_i at each iteration and then select the most relevant object in X_i during the selection phase. The generation of elements takes advantage of the *birth map* to speed up the process.

Now rises the question of how to select the best candidate inside each X_i. If all the X_i were independent, then the selection of every $\omega_i^j \in X_i$ could simply be calculated based on the data term $u_d(\omega_i^j)$. However, if we consider a dense configuration of objects during the birth step, the independence hypothesis does not hold.

We propose the optimal selection of ω' from an almost very dense configuration . The idea is to generate such that the interaction graph between sets X_i remains a tree (with no loops). The global optimum $!_0$ can then be inferred rapidly on this tree using belief propagation [30].

Belief propagation is a particular case of dynamic programming, more precisely, it is a variation of Dynamic Time Warping suitable to trees instead of chains, often formulated with message passing. The core of the algorithm relies on the tree structure of the interactions between variables, i.e. if ω_1 is a leaf, it interacts with only one variable, ω_2:

$$\inf_{\omega_1, \omega_2, ..., \omega_n} \left[\sum_i u_d(\omega_i) + \sum_{i \sim j} u_p(\omega_i, \omega_j) \right] = \inf_{\omega_2, \omega_3, ..., \omega_n} \left[\sum_{i>1} u_d(\omega_i) + \sum_{i \sim j > 1} u_p(\omega_i, \omega_j) \right]$$

where $v_d = u_d$ except for $v_d(\omega_2) = u_d(\omega_2) + \inf_{\omega_1}\{u_d(\omega_1) + u_p(\omega_1, \omega_2)\}$. This optimization over ω_1 given ω_2 is easy to perform and rewrites the problem into a similar one but with one fewer variable. Repeating this trick n times solves the problem, with linear complexity in the number of variables.

Once a configuration Γ is generated, we apply belief propagation to select the best candidate inside each X_i, which gives the global optimum ω' of this configuration Γ.

While generating Γ, the algorithm keeps track of the created neighborhood to verify that it always represents a tree.

The generation and selection phase schedules are presented on figure 6. On figure 6(a), we present the current configuration $\omega_{[n]} = \{a, b, c\}$. In figure 6(b), the algorithm generates a dense configuration $= \{X_1, X_2, X_3, X_4\}$. We apply the belief

propagation on to select only one (the best one) from each X_i candidates ($\{\omega_i^0, \omega_i^1,...,\omega_i^j\}$) as on figure 6(c) by $\omega' = \{d, e, f, g\}$. On figure 6(d), we present the combination of the current configuration $\omega_{[n]}$ and the newly proposed and selected ω' by $\omega = \omega_{[n]} \cup \omega'$ on which the graph is constructed for the Cut step. In figure 6(e) we present the tree structure (forest) for a much larger configuration, showing each X_i as a node, and the existing connections between them representing the neighborhood of each object (no loops).

On figure 7(a) we present a sample from another flamingo colony and on 7(b) the detection result using optimized-MBC algorithm, showing the quality of the detection.
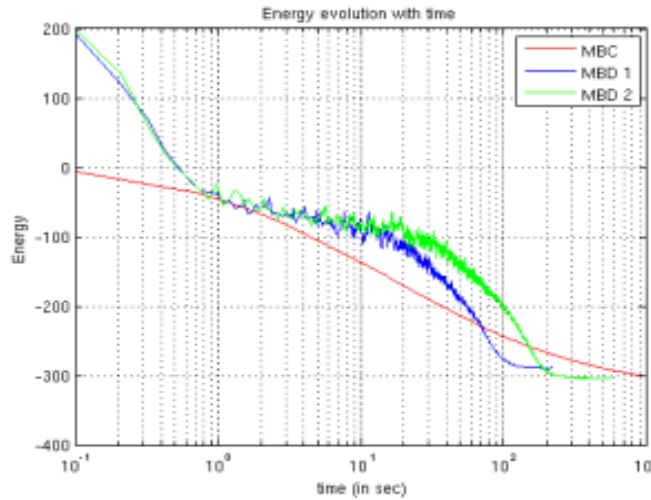
Figure 5. Energy evolution of the configuration during the optimization with respect to time
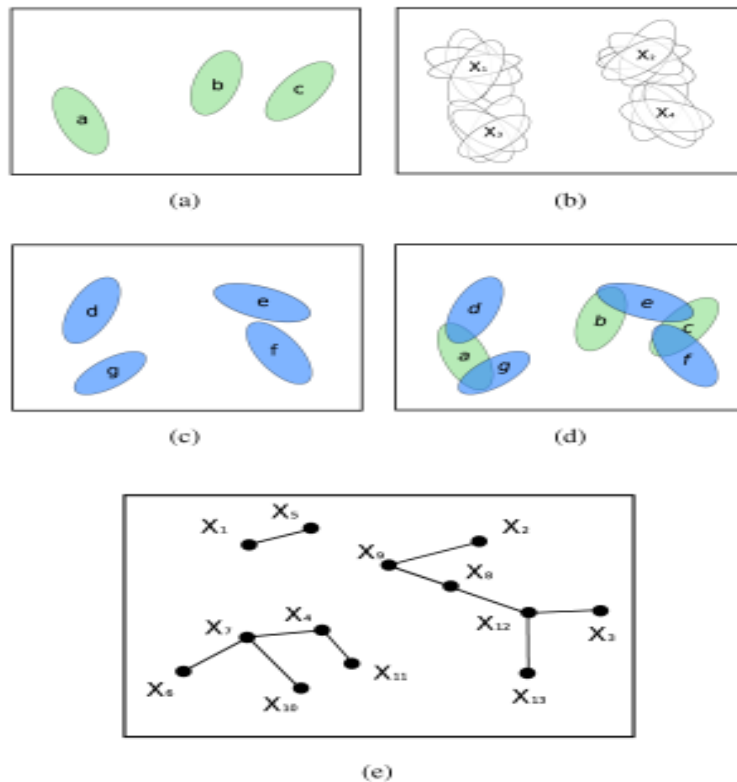

Figure 6. (a) Current configuration $\omega_{[n]}$ in green. (b) Proposed dense configuration . (c) Selected $\omega_{[n]}$ from the candidates of Γ. (d) The configuration $\omega = \omega_{[n]} \cup \omega'$ on which the graph is constructed for the Cut step. (e) A forest of trees of a large configuration from which we select on object per node using belief propagation
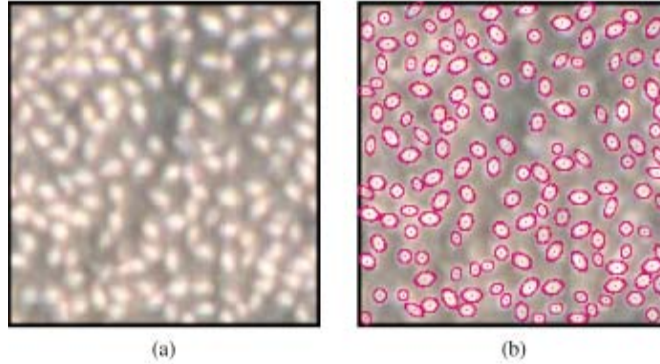
Figure 7. (a) A sample from a flamingo colony ©Tour du Valat. (b) The detection
result, each flamingo is surrounded by a pink ellipse ©Ariana/INRIA

4. Energy Comparison

In this section, we verify that after this modification of the data term (10), we still minimize the same energy using graph cut at each iteration.

For the graph cut algorithm, edge weights have to be nonnegative, so we normalize the data term $Q_d(d_B) \in [0, 1]$. For each ω_i, the data term becomes:

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

Let $U^{GC}(\omega_i)$ be the energy given by the graph cut, with $\omega = \omega_{[n]} \cup \omega'$ g where $\omega_{[n]} = \{\omega_1,..., \omega_p\}$ and $\omega' = \{\omega_{p+1},..., \omega_q\}$. 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:

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

and the prior term is given by:

$$U_d^{GC}(\omega) = \sum_{i=1}^{p}\sum_{j=p+1}^{q} u_p(\omega_i,\omega_j)\delta_{f(\omega_i)=0}\,\delta_{f(\omega_i)=1}$$

$u_p(\omega_i,\omega_j)$ 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:

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

$$= \sum_M u_d(\omega_i) + \sum_{M \cup D} \left(\frac{1-u_d(\omega_i)}{2}\right)$$

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

$$= U_d(\omega) + K(\omega)$$

where after optimization M is the set of objects that we keep and D is the set of objects that we kill. Thus minimizing $U_d^{GC}(\omega)$ is equivalent to minimizing $U_d(\omega)$ plus a constant $K(\omega)$, function of the configuration ω but not of M. It becomes:

$$\operatorname*{argmin}_{\omega} U_{GC}(\omega) = \operatorname*{argmin}_{H} U(\omega)$$

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

1) Convergence: The algorithm keeps iterating until convergence. Convergence can be evaluated by monitoring the number of objects or the energy of the configuration: 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 ! 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 lower-bounded. Therefore, we have a suf- ficient condition for our algorithm to converge at least to a local minimum.

5. Results

In this section we present results of flamingo detection from aerial images comparing MBC, Optimized-MBC (using belief propagation) and MBD algorithm. First we present results on four different colonies. In table III, data is composed of two to three samples from each of the four colonies. We show the percentage of correct detection of flamingos, negative false and positive false. These results are validated by ecologists5. Results in table III show that the newly proposed algorithms outperform the MBD algorithm for the detection. For the detection rate, MBC outperforms MBD, and Optimized-MBC outperforms both of them. Both MBC algorithms have lower negative and positive rates for the majority of the samples.

Secondly, we present the energy evolution during the optimization using MBD, the basic MBC and the optimized MBC algorithms while presenting at the same time the object detection rates. We compare the three algorithms on three samples of different size, the approximate number of flamingos in those samples are 250, 1900, and 3200 (computed from evaluation). Figures 8(a,c,e) show the energy evolution with respect to time of the three algorithms for the first, second and third samples respectively. We conclude that MBD can reach a lower minimum of the energy faster than MBC, but optimized MBC reaches lower minimum, whatever the size of the colony. For the detection rate, as presented in figure 8(b,d,f), MBD has the lowest detection rate because of the difficulty of parameter tuning; MBC has the highest detection rate for small configuration size, while for average size, it becomes similar to the optimized MBC, and for large colonies, optimized MBC has the highest detection rate; knowing that both MBC algorithms give very small negative false rates.
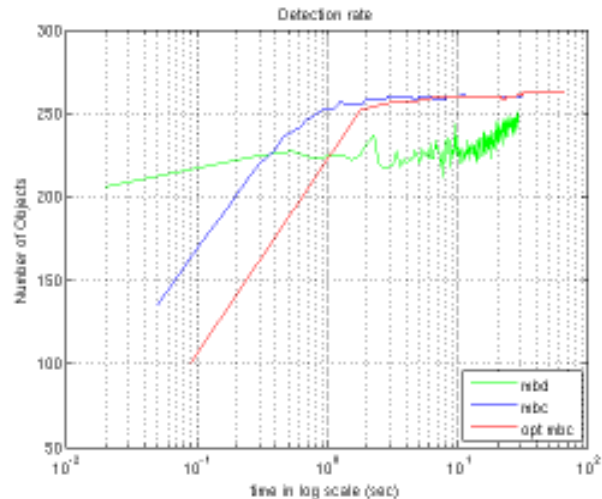
We used the graph-cut code developed by Olga Veksler [25], [31], [2].
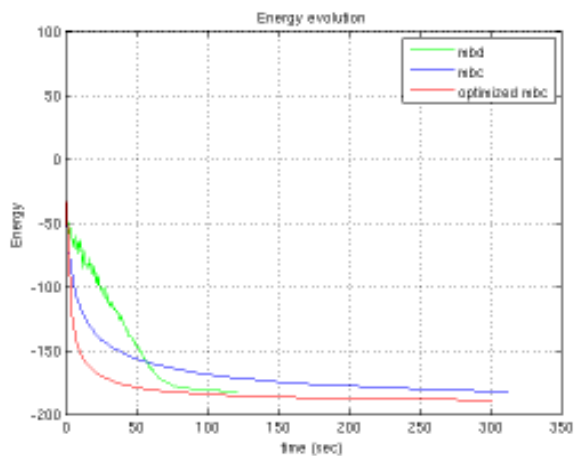
6. Conclusions And Future Work

In this paper we presented an efficient optimization algorithm to minimize a highly non-convex energy function which was previously optimized within a simulated annealing scheme.
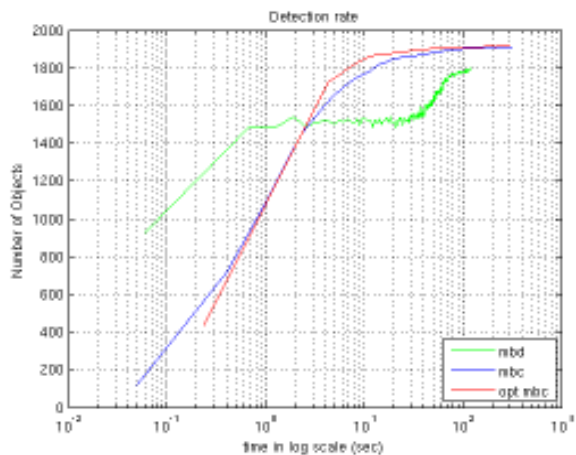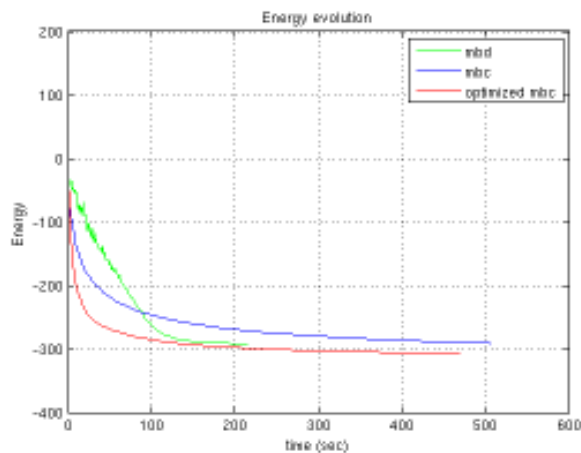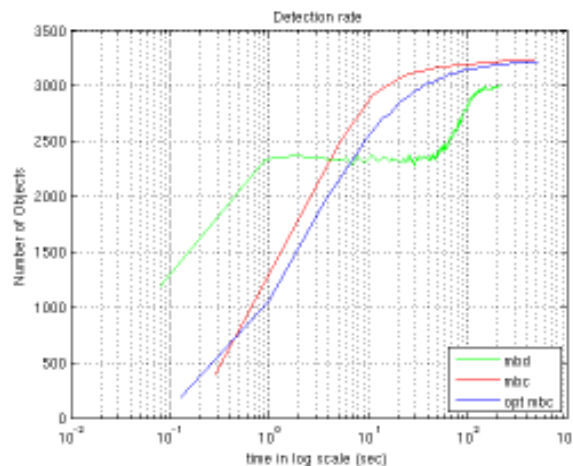

(a)

(b)
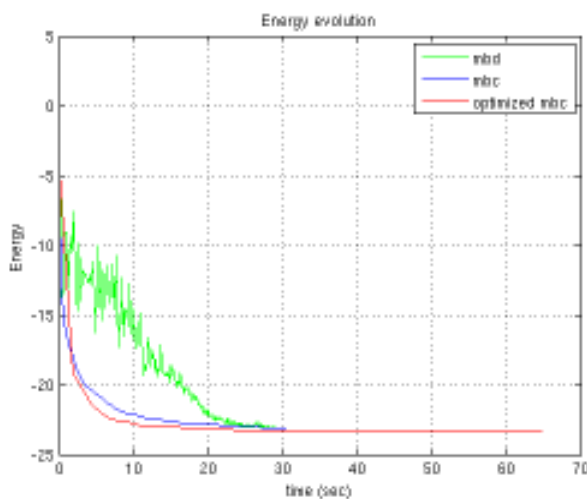
Figure 8. (a, c and e) show the energy evolution for the three samples of around 250, 1900 and 3200 flamingos. (b, d and f) show the flamingo detection rate for the same samples

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 data-sets. The basic MBC algorithm reaches a lower energy level than MBD but requires more computation time. We also presented an optimized version of the MBC algorithm, using belief propagation to optimize the newly proposed configuration at each iteration in order to obtain a relevant proposed configuration. The results show that the optimized MBC is substantially faster than the basic MBC algorithm. We demonstrated how our algorithm, defined in the MPP framework, can be used to efficiently solve the flamingo counting problem as one of many possible applications. More specifically, flamingo colonies consist of thousands of objectss which makes the use of our algorithm advantageous for the application.

We are currently studying the minimum energy obtained via our algorithm. We are investigating possible options for graph re-usage instead of constructing a graph at each iteration, and also the possibility to use parallelization techniques.

7. Acknowledgment

Image	Qualifiers	MBC	Opt. MBC	MBD
Fang02 sample 1	Good detection	93	90	87
	Neg. false	0.07	0.08	0.13
	Pos. false	0.16	0.12	0.09
Fang02 sample 2	Good detection	98	98	96
	Neg. false	0.02	0.02	0.04
	Pos. false	0	0.11	2
Fang05 sample 1	Good detection	86	85	82
	Neg. false	0.14	0.15	0.18
	Pos. false	0.1	0.2	0.07
Fang05 sample 2	Good detection	97	97	90
	Neg. false	0.03	0.03	0.1
	Pos. false	0.08	0.07	0.14
Fang05 sample 3	Good detection	94	95	90
	Neg. false	0.1	0.40	0.1
	Pos. false	0.06	0.13	0.14
Tuz04 sample 1	Good detection	100	100	99
	Neg. false	0.0	0.0	0.01
	Pos. false	0.0	4 0.01	0.01
Tuz04 sample 2	Good detection	98	98	98
	Neg. false	0	0	0
	Pos. false	0.04	0.04	0.04
Tuz04 sample 3	Good detection	100	100	100
	Neg. false	0	0	0
	Pos. false	0.02	0	0
Tuz06 sample 1	Good detection	100	100	100
	Neg. false	0	0	0
	Pos. false	0.01	0	0
Tuz06 sample 2	Good detection	98	100	95
	Neg. false	0	0	0.04
	Pos. false	0.09	0	0.06
Tuz06 sample 3	Good detection	99	99	95
	Neg. false	0.01	0.01	0.04
	Pos. false	0.12	0.12	0.08

Table 3. Comparison between MBC, optimized MBC and MBD

References

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

[2] Boykov,Y., Kolmogorov, V (2004). An experimental comparison of mincut/ max-flow algorithms for energy minimization in vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (PAMI), 26 (9) p. 1124–1137.

[3] Geyer,C. J., Moller, J (1994). Simulation and likelihood inference for spatial point processes, *Scandinavian Journal of Statistics*, 21, p. 359–373.

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

[5] Geyer, 2004 (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, Biometrika, 82, p. 711–732.

[6] Stoica, R., Descombes, X., Zerubia, J (2004). A Gibbs point process for road extraction in remotely sensed images, *International Journal of Computer Vision* (IJCV), 57 (2) 121–136.

[7] Perrin, G., Descombes, X.m Zerubia, J (1006). 2D and 3D vegetation resource parameters assessment using marked point processes, *In:* Proc. International Conference on Pattern Recognition (ICPR), Hong-Kong,
August.

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

[9] 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, p. 347–359.

[10] Gamal Eldin, A., Descombes, X ., Zerubia, J (2010).Multiple birth and cut algorithm for point process optimization," in Proc. IEEE SITIS, Kuala Lumpur, Malaysia, December 2010. [Online]. Available: http://hal.archives-ouvertes.fr/inria-00516305/fr/

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

[12] Descamps, S. . Descombes, X., B´echet, A., Zerubia, J (2008). 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.

[13] Kulikova, M. S., Jermyn, I. H., Descombes, X., Zhizhina, E., Zerubia, J (2010). 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.

[14] Baddeley, A., Turner, R (2006). Modelling spatial point patterns in R, *In:* Case Studies in Spatial Point Pattern Modelling. Lecture Notes in Statistics 185, 2374. Springer.

[15] Chatelain, F., Descombes, X., Zerubia, J. (2009). 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.

[16] Greig, D. M.,Porteous, B. T., Seheult, A. H (1989). Exact maximum a posteriori estimation for binary images, *Journal of the Royal Statistical Society. Series B (Methodological)*, 51 (2) 271–279. [Online]. Available: http://www.jstor.org/stable/2345609

[17] Boykov, Y., Veksler, O., Zabih, R.(1998). Markov random fields with efficient approximations, *In*: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 648–655.

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

[19] Boykov, Y. (2993). Computing geodesics and minimal surfaces via graph cuts, *In:* Proc. International Conference on Computer Vision (ICCV) p. 26–33.

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

[21] Kohli, P., Torr, P.H.S (2007). Dynamic graph cuts for efficient inference in Markov random fields," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 29 (12) 2079–208.

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

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

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

[25] Kolmogorov,V., Zabih, R(2004). What energy functions can be minimized via graph cuts?, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 26 (2) 147–159, February.

[26] Goldberg, A. V. , Tarjan, R. E (1988). A new approach to the maximum-flow problem, *J. ACM*, 35 (4) 921–940.

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

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

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

[30] Pearl, J (1998).Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers.

[31] Boykov, Y., Veksler, O., Zabih, R (2002). Fast approximate energy minimization via graph cuts, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23 (11) 1222–1239.