

A FAST MULTIPLE BIRTH AND CUT ALGORITHM USING BELIEF PROPAGATION

Ahmed Gamal-Eldin, Xavier Descombes, Guillaume Charpiat and Josiane Zerubia.

INRIA Sophia-Antipolis Méditerranée
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 present a faster version of the newly proposed Multiple Birth and Cut (MBC) algorithm. MBC is an optimization method applied to the energy minimization of an object based model, defined by a marked point process. We show that, by proposing good candidates in the birth step of this algorithm, the speed of convergence is increased. The algorithm starts by generating a dense configuration in a special organization, the best candidates are selected using the belief propagation algorithm. Next, this candidate configuration is combined with the current configuration using binary graph cuts as presented in the original version of the MBC algorithm. We tested the performance of our algorithm on the particular problem of counting flamingos in a colony, and show that it is much faster with the modified birth step.

Index Terms— point process, multiple birth and cut, belief propagation, graph cut, multiple object detection

1. INTRODUCTION

Multiple objects detection from a still image is a very challenging problem, finding applications in different domains such as counting crowd, evaluation of tree populations, roads extraction, in urban planning and military intelligence. A recent object based method embedded in a Marked Point Process (MPP) framework has proved to be efficient for solving many such challenging problems, especially on high resolution images. The MPP modeling is based on a configuration space of the objects of interest, on which a Gibbs energy function is defined [1]. The solution is then obtained by minimizing this energy.

Initially, Markov random field samplers, either stochastic or deterministic, were based only on *standard moves* within the framework of Metropolis Hasting, where only one pixel changes at a time. During the last decade, *multiple moves* methods emerged and most of them are based on graph cut techniques [2]. MPP samplers have also evolved from *simple perturbations* (standard moves) with methods like Reversible Jump Markov Chain Monte Carlo method [3] to *multiple perturbations* using the Multiple Birth and Death (MBD) algo-

rithm [4]. Very recently, an optimization method based on graph cuts, named Multiple Birth and Cut was introduced in [5]. The authors proposed an iterative algorithm to explore the configuration space, where they used a special graph to choose between current objects and newly proposed ones using binary graph cuts. Although the MBC method presented in [5] is slower than the MBD algorithm, it has very promising advantages such as getting rid of the simulated annealing framework and related parameters.

Our main contribution herein is a modification related to the birth step in the MBC algorithm, which consists of proposing *relevant* objects leading to a faster convergence.

2. MARKED POINT PROCESS

2.1. Point Process

Definition. A point process is a random variable whose realizations are random configurations of points. Let us consider a point process X taking values in $K = [0, I_{max}] \times [0, J_{max}]$, a closed, connected subset of \mathbb{R}^2 . X is a measurable mapping from a probability space $(\Omega, \mathcal{A}, \mathcal{P})$ to the set of configurations of points of K . This mapping defines a point process.

Marked Point Process. Point processes were introduced in image processing because they allow to easily model scenes made of objects. Objects can have simple or complex shapes. In this paper for instance, each flamingo (object) is modeled by 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 define¹ a minimum and a maximum value. The geometrical parameters of the shape represent the mark m_i associated to each point x_i . Therefore, an object is defined as $\omega_i = (x_i, m_i) \in K \times M$, where x_i represents the object location and m_i its mark.

We consider a marked point process with points in K and

¹Knowing the average object dimensions, its parameters min and max values are defined based on the image resolution. The maximum dimension of a flamingo in reality (from a top view) is 80cm.

with marks in M , the configuration space is then defined as:

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

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 $\nu(\omega)$ and the Lebesgue measures on the mark space:

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

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

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

Markov Point Process. We consider a Gibbs/Markov process defined by local interactions between objects. The density of the process is given by:

$$f(\omega) = \frac{1}{Z} \exp[-U(\omega)] = \frac{1}{Z} \exp[-(U_d(\omega) + \gamma_p U_p(\omega))], \quad (3)$$

where U_p is the prior energy which takes into account the interactions between geometric objects, U_d is the data energy to fit the configuration to the image, Z is the partition function and γ_p is the weight of the prior term. The searched solution corresponds to the minimum of the energy $U(\omega)$.

2.2. Prior

We define a non-overlapping interaction to avoid multiple detection of the same object. 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 proposed in [6]:

$$\mathcal{A}(\omega_i, \omega_j) = \frac{A(\omega_i \cap \omega_j)}{\min(A(\omega_i), A(\omega_j))}, \quad (4)$$

where $A(\omega_i)$ is the area of object ω_i . We forbid any configuration with overlapping coefficient greater than 10%. The total prior energy of the configuration is then given by [5]:

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

where $u_p(\omega_i, \omega_j) = \infty$ if $\mathcal{A}(\omega_i, \omega_j) > 0.1$ and 0 otherwise.

2.3. Data term

Each flamingo can be modeled as a bright ellipse surrounded by a darker background. We want to evaluate the contrast between the ellipse interior and the background, to evaluate the fitness of a proposed object. For an object $\omega_i = (x_i, m_i)$, with marks $m_i = (a, b, \theta)$, we define $\mathcal{F}(\omega_i)$ as the subset of K , between the border of ω_i and a concentric ellipse ω'_i , with

marks $m'_i = (a + \rho, b + \rho, \theta)$, which represents the background. To evaluate the contrast, assuming Gaussian distributions, we compute statistics on the image inside respectively the ellipse ω_i and $\mathcal{F}(\omega_i)$. We estimate the means and standard deviations of the object (μ_1, σ_1) and the corresponding background (μ_2, σ_2) , we then compute the corresponding Bhattacharya distance d_B . The total data term energy is given by:

$$U_d(\omega) = \sum_{\omega_i \in \omega} u_d(\omega_i), \quad (5)$$

where $u_d(\omega_i) = \mathcal{Q}_d(d_B(\omega_i), \mathcal{F}(\omega_i))$, where $\mathcal{Q}_d(d_B) \in [0, 1]$ is a quality function which gives positive values to small distances and negative values when $d_B(\omega_i)$ is greater than a given threshold [6].

3. MULTIPLE BIRTH AND CUT

To minimize the energy $U(\omega)$, an algorithm based on a graph cut (MBC) has been recently proposed [5]. It consists of iterating a *birth* step, where new objects are added to the current configuration and a *cut* step, keeping the best subset of objects among the current and new together. The speed limitation of this algorithm [5] mainly 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 benefit from the *birth map*² and consequently requires a large number of iterations.

In this paper, we propose to insert a *selection phase* in the birth step, which allows generating and inserting many more objects in the birth step, thus reducing the number of iterations. The algorithm is summarized in algorithm 1.

Algorithm 1 Multiple Birth and Cut

- 1: $n \leftarrow 0, R \leftarrow \text{const}$
 - 2: generate $\omega', \omega_{[0]} \leftarrow \omega'$
 - 3: **repeat**
 - 4: Birth: generate Γ
 - 5: $\omega' \leftarrow \text{Select_from}(\Gamma)$
 - 6: Cut: $\omega_{[n+1]} \leftarrow \text{Cut}(\omega_{[n]} \cup \omega')$
 - 7: $n \leftarrow n + 1$
 - 8: **until** converged
-

3.1. Selection Phase

In the birth step, the algorithm generates a dense configuration Γ . This configuration has a special organization, where $\Gamma = \{X_0, X_1, \dots, X_n\}$ and $X_i = \{\omega_i^0, \omega_i^1, \dots, \omega_i^l\}$. Each X_i encodes l candidates from which only one should be kept, see figure 1(b). The aim of this organization is, instead of proposing a single object ω_i to detect the real object \mathbf{o}_j , we

²To speed up the process, we consider an inhomogeneous birth rate to favor birth of several objects in the neighborhood of well chosen positions

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 Γ elements take advantage of the *birth map* to speed up the process, while staying uniform over the mark space M .

Now raises 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 $\omega_j^i \in X_i$ could simply be calculated based on the data term $u_d(\omega_j^i)$. However, if we consider a dense configuration of objects during the birth step, the independence hypothesis is broken.

In this paper we propose an optimal selection of ω' from an almost very dense configuration Γ . The idea is to generate Γ such that the interaction graph between variables ω_i remains a tree (with no loop). The global optimum ω' can be inferred rapidly on this tree using belief propagation [7].

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 and thus interacts with only one variable, ω_2 :

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

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 the belief propagation algorithm to select the best candidate inside each X_i , which gives the global optimum ω' from 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 progress are presented in figure 1. In figure 1(a), we present the current configuration $\omega_{[n]} = \{a, b, c\}$. In figure 1(b), the algorithm generates a dense configuration $\Gamma = \{X_1, X_2, X_3, X_4\}$, and from each X_i candidates, only one is kept as in figure 1(c), $\omega' = \{d, e, f, g\}$. In figure 1(d), we present $\omega = \omega_{[n]} \cup \omega'$ on which the graph is constructed for the Cut step.

3.2. Cut Step

Given the output of the previous iteration $\omega_{[n]}$ and the newly proposed ω' , we want to find the optimal configuration from $\omega_{[n]} \cup \omega'$ that minimizes the total energy. We briefly explain this cut step based on graph cuts, for details please refer to [5].

In contrast to most graph cut problems where the graph nodes represent pixels, the nodes of our graph represent ob-

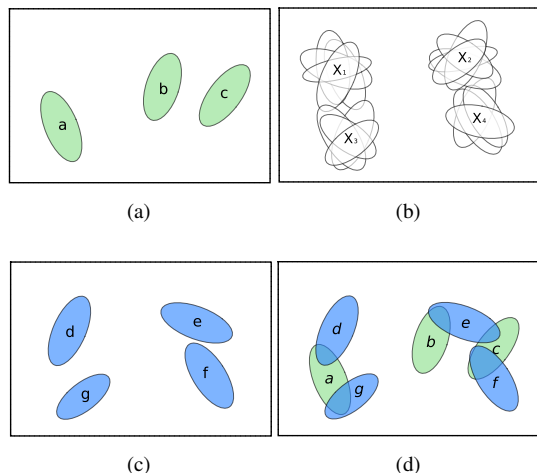


Fig. 1: (a) Current configuration $\omega_{[n]}$ in green. (b) Proposed dense configuration Γ . (c) Selected $\omega'_{[n]}$ from the candidates of Γ . (d) The configuration $\omega = \omega_{[n]} \cup \omega'$ on which the graph is constructed for the Cut step.

jects. Each node is connected to the terminals S and T , and the connecting edges are referred to as *t-links*. Interacting nodes, are connected with edges which are referred to as *n-links*. To the *t-links* and the *n-links* weights are assigned the data term and the prior term respectively.

A binary graph cut is applied to this graph, which assigns a label '0' or '1' to each node (object). The key for using this labeling to keep the good objects and remove the non-fitting ones is that the labeling 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.

For the *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 opposite for $\omega_i \in \omega'$, with $1 - u_d(\omega_i)$ to the source and $u_d(\omega_i)$ to the sink. For the *n-links*, all costs are zeros except for the labeling (0, 1) which corresponds to the case of two overlapping objects, in which case the cost is *infinity*.

4. RESULTS

In this section we present some results of flamingo detection from aerial images comparing our new algorithm to the standard MBC algorithm [5]. First we present the detection result on a sample from a colony image. In figure 2(a) we present the sample, and in 2.(b) the detection result, showing the quality of the detection. Secondly, we present the energy evolution during the optimization of both the basic MBC and the newly proposed version, and we also present object detection rates. We compared both algorithms on three samples of around 260, 1900, and 3250 flamingos. Figure 3.(a,c,e)

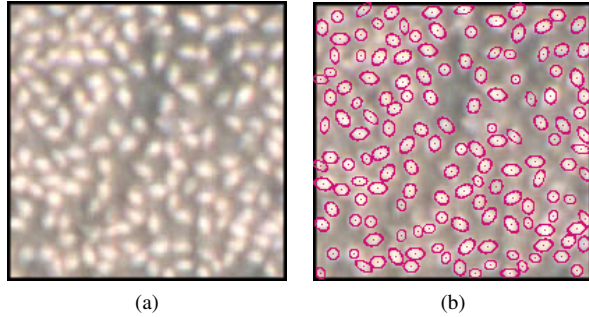


Fig. 2: (a) A sample from a flamingo colony. (b) The detection result, each flamingo is surrounded by a pink ellipse.

shows the energy evolution with respect to time of both algorithms for the first, second and third samples respectively. We conclude that for a very small number of objects, there is no advantage using the modified version of the algorithm, while it becomes very interesting when the number of objects increases. The conclusion is similar for the rate of object detection, as presented in figure 3(b,d,f), our algorithm has a higher detection rate for relatively large number of objects.

5. CONCLUSION

So far, optimizing the MPP was only possible within a simulated annealing scheme, until, recently, an efficient optimization algorithm for MPP based on graph cut was proposed [5]. In this paper we presented a new version of this optimization algorithm, using belief propagation to optimize the proposed configuration inside each iteration to obtain a relevant proposed configuration. The results show that our algorithm is substantially faster than the basic MBC algorithm. Flamingo colonies consist in general of more than one thousand objects, which makes our algorithm much more interesting for a real application. We have demonstrated how our algorithm in the MPP framework can be used to efficiently solve the flamingo counting as one of many possible applications.

Acknowledgment

The authors would like to thank Antoine Arnaud and Arnaud Béchet from Tour du Valat for providing the flamingo images.

6. REFERENCES

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

[2] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” *IEEE Trans. PAMI*, vol. 26, no. 9, pp. 1124–1137, September 2004.

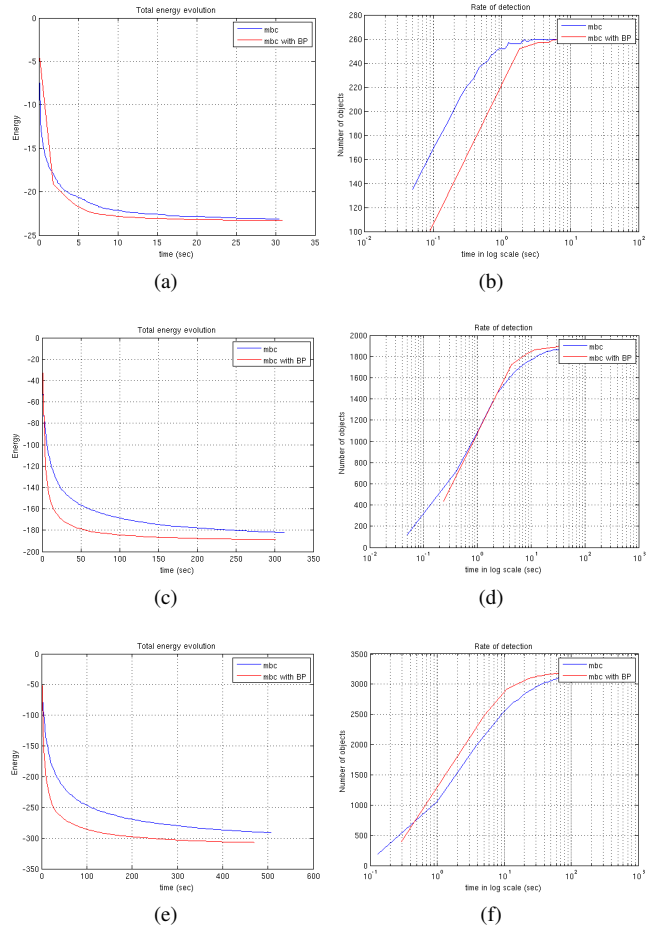


Fig. 3: (a, c, e) Shows the energy evolution for the three samples of around 260, 1900, and 3250 flamingos. (b, d, f) Shows the flamingo detection rate for the same samples.

[3] P. J. Green, “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination,” *Biometrika*, vol. 82, pp. 711–732, 1995.

[4] X. Descombes, R. Minlos, and E. Zhizhina, “Object extraction using a stochastic birth-and-death dynamics in continuum,” *JMIV*, vol. 33, pp. 347–359, 2009.

[5] A. Gamal Eldin, X. Descombes, and J. Zerubia, “Multiple birth and cut algorithm for point process optimization,” in *Proc. IEEE SITIS*, Kuala Lumpur, Malaysia, December 2010.

[6] S. Descamps, X. Descombes, A. Béchet, and J. Zerubia, “Automatic flamingo detection using a multiple birth and death process,” in *Proc. IEEE ICASSP*, Las Vegas, USA, March 2008.

[7] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, 1988.