

A novel algorithm of occlusions and perspective effects using a 3D object process

Ahmed Gamal Eldin, Xavier Descombes, Josiane Zerubia

► **To cite this version:**

Ahmed Gamal Eldin, Xavier Descombes, Josiane Zerubia. A novel algorithm of occlusions and perspective effects using a 3D object process. ICASSP 2011 (International Conference on Acoustics, Speech and Signal Processing), May 2011, Prague, Czech Republic. 2011. <inria-00592449>

HAL Id: inria-00592449

<https://hal.inria.fr/inria-00592449>

Submitted on 12 May 2011

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

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

A NOVEL ALGORITHM FOR OCCLUSIONS AND PERSPECTIVE EFFECTS USING A 3D OBJECT PROCESS

Ahmed Gamal-Eldin, Xavier Descombes and Josiane Zerubia

INRIA Sophia-Antipolis Méditerranée
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 introduce a novel probabilistic approach to handle occlusions and perspective effects. The proposed method is an object based method embedded in a marked point process framework. We apply it for the size estimation of a penguin colony, where we model a penguin colony as an unknown number of 3D objects. The main idea of the proposed approach is to sample some candidate configurations consisting of 3D objects lying in the real plane. A Gibbs energy is defined on the configuration space, which takes into account both prior and data information. These configurations are projected onto the image plane. The configurations are modified until convergence using the multiple birth and death optimization algorithm and by measuring the similarity between the projected image of the configuration and the real image. During optimization, the proposed configuration is modeled by a mixed graph which represents all dependencies between the objects, including interaction between neighbor objects and parent-child dependency for occluded objects. We tested our model on synthetic image, and real images.

Index Terms— 3D point process, occlusion, mixed graph, object extraction, penguins colony

1. INTRODUCTION

Multiple object detection from a still image is a very challenging problem, finding applications in different domains. Our work focus is on large and dense scenes, with main application being the evaluation of a population of penguins, for our ecologist collaborators. Predicting the impact of future climate changes on populations and biodiversity is a central issue in the context of global climate warming. Seabirds are sensitive indicators of changes in marine ecosystems and might integrate and/or amplify the effects of climate forcing on lower levels in food chains. It is crucial to understand how and to what extent organisms are able to cope with climatic variations, especially in polar environments where the effect of climate change is the strongest. We consequently need long term monitoring system allowing to estimate each year the

number of breeding penguins in the colonies and their breeding success. The development of automatic counting system by photographic analysis of penguin colonies could be a powerful and cheap tool to monitor demography of penguins in remote areas and to study the impact of climate changes.

2. MARKED POINT PROCESS MODEL

Marked point process (MPP) 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; it was first introduced in [1]. See [2] for details.

Configuration Space: Lets consider a space $K \times \mathbb{R}^+$, where K is a closed, connected subset of \mathbb{R}^2 , referred to as the real plane and \mathbb{R}^+ refers to the z-axis (objects height). We consider a perspective transformation, supposed as known, \mathcal{P} , such that $\mathcal{I} = \mathcal{P}(K \times \mathbb{R}^+)$. The data consists of $I = (i_s)_{s \in S}$, where S is a discretization of the image plane \mathcal{I} and $i_s \in \Lambda$ is the color of pixel s . We consider configurations of an unknown number of objects in the real plane. For our application, consisting in counting a population of penguins, we consider two classes of objects $l \in \{0, 1\}$ corresponding respectively to adults and chicks. 3D color models of each class are designed using a CAD software¹. We consider two positions for the penguins $p \in \{0, 1\}$ corresponding to standing and lying (on the ground) animals. Finally, a penguin is also characterized by its orientation ϕ with respect to the z-axis (in cylindrical coordinates). 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, $M = \{0, 1\} \times \{0, 1\} \times [0, 2\pi[$. We consider a marked point process with points lying in K and marks in M . The configuration space is then defined as $\Omega = \bigcup_{n=0}^{\infty} \Omega_n$, $\Omega_n = \{\{\omega_1, \dots, \omega_n\} \subset K \times M\}$, where Ω_n is the subset of con-

¹Software such as Blender and 3D Max

figurations containing exactly n objects. The process is defined on the space Ω by $d\pi(\omega) = h(\omega)d\nu(x) \prod_n (d\mu(m_i))$, where $\omega = \{(x_i, m_i), i = 1, \dots, n\}$, $x = \{x_i, i = 1, \dots, n\}$, $m = \{m_i, i = 1, \dots, n\}$, $\nu(\cdot)$ is the measure of the Poisson process of intensity $\lambda(u)$, $u \in K$, $\mu(\cdot)$ is a measure on M and $h(\cdot)$ is a density. We consider a Gibbs density:

$$h(\omega) = \frac{1}{Z} \exp\{-U(\omega)\}, \quad (1)$$

where Z is the partition function (normalizing constant), and $U(\cdot)$ an energy. This energy takes into account the interactions between geometric objects (prior energy) and a data energy to fit the configuration to the image (data energy): $U(x) = U_d(x) + \gamma_p U_p(x)$.

Prior Energy: For each 3D object ω_i , we consider his occupancy area in K by an ellipse $C(\omega_i)$. Both standing and lying objects are well approximated by this ellipse. The overlapping between two objects is then approximated by the intersection of the two corresponding ellipses, which reduces the prior energy term to the exact model as [3].

3. OCCLUSION AND PERSPECTIVE

We consider an object based data energy, which means that the global data term is a sum of local terms applied on each object in the configuration. Unlike existing approaches using marked point processes for object detection, we cannot consider the data term associated to an object independently of the remaining objects in the configuration. Indeed, the data differs from the real plane where the objects are considered. It is the data of 3D objects living on this plan and we only have a perspective view of the scene, which introduces occlusions and therefore dependencies between the projected objects. Computing the data term requires to project the scene on the image plane. This can be obtained rapidly using the GPU and OpenGL environment.

Lets consider the configuration ω presented in figure 1, from 1.(a) and (b), we notice the dependencies between objects, in the prior and data term. *e.g.*, object I and J interact, a prior energy exist on this clique, and this is presented in the graph in figure 1.(c) with a undirected edge. Object dependencies are represented by a mixed graph G , where $G = (V, E, P)$. G being the graph composed of V vertices representing the objects, E the edges connecting those objects and P to indicate the direction if the edge is directed. We consider a data term associated with the configuration ω given by:

$$U_d(\omega) = \sum_{\omega_i \in \omega} u(\omega_i | c(\omega_i)), \quad (2)$$

where $c(\omega_i)$ denote children of ω_i , which depend on the camera position with respect to the scene. A child is an object which inherits his parent (some) pixels if the parent was removed from the configuration.

Given the proposed configuration ω (figure 1), *e.g.* lets calculate the death probability object A . From figure 1.(b), we see that object A partially hides object E . In the mean time, object B also partially occlude object E . Object A and object B are both parents of object E , this is represented by directed arrows in the mixed graph in figure 1.(c). The delta energy associated with the removal of object A is given by:

$$\begin{aligned} \Delta E_A &= E(\omega \setminus A) - E(\omega) \\ &= E'_{B|\partial(B)} + E'_E - \left(E_{A|\partial(A)} + E_{B|\partial(B)} + E_E \right), \end{aligned}$$

where $E'_{B|\partial(B)}$ is the *new energy* of the B after the removal of A , since object B gain some pixels so its data energy changes, and ∂ indicate the neighbors of an object. From the above equations we notice the new dependence between object A and B , since they share a common child. This is presented in the mixed graph with a dotted line. We call this extra dependence a *moralization* of the graph. This parent-child relation induces a new dependencies in the MPP model. This dependency is *markovian*. *e.g.* the existence of object F , made object B independent of object H ; in the case of absence of F , B would have been parent of H .

For the mean time, we are testing a naive data term based on an interactive segmentation method, while the aim is a fully automated system. The segmentation gives an image with three labels, adults, chicks and background. This reduces the similarity measure of an object to a simple count of the number of correct labeled pixels inside the proposed object. For a given object ω_i , let \mathbf{o}_i be the set of projected pixels and let \mathbf{y}_i be there corresponding set of pixels in the input image. The goal is to measure the distance $d(\mathbf{o}_i, \mathbf{y}_i)$.

$$u_d(\omega_i) = \sum_{s \in \mathbf{o}_i} \alpha_i \left(\frac{-\lambda \#_{good} + \#_{wrong}}{\lambda \#_{good} + \#_{wrong}} \right)^n, \quad (3)$$

where $\#_{good}$ is the number of pixels whose class (label) matches, $\#_{wrong}$ is the number of pixels who do not match. α_i model the *effect of partial occlusion* of ω_i , by giving a lower weight to occluded objects based on the ratio of its number of visible pixels to its total number of pixels.

Perspective effect: For the synthetic images, the camera parameters are known. First collection of real photos taken by the ecologists did not have the camera parameters. Recently we received a new collection of images based on a defined protocol between us and the ecologist, where we can recover the camera parameters. The change in scale is naturally obtained (handled) by projecting the 3D scene, so far objects become small with the perspective projection.

4. OPTIMIZATION

Once the density defined, the issue now is to sample it to obtain corresponding realization. No direct simulation is possible due to the normalization constant. Multiple Birth and

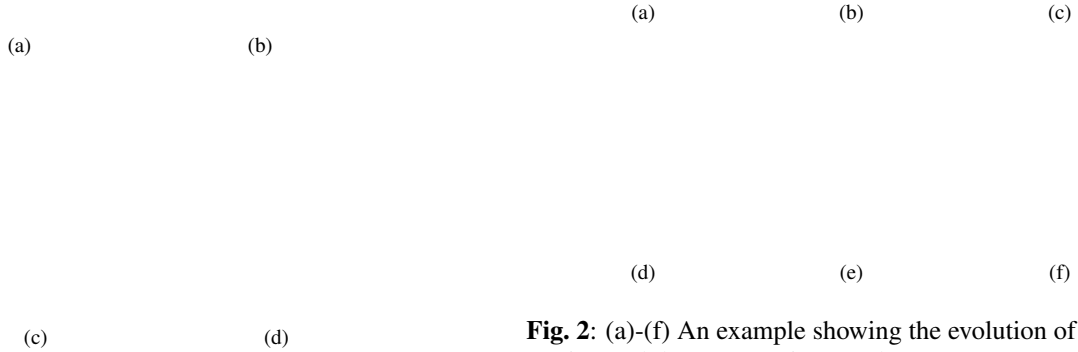


Fig. 1: (a) Top view of a configuration ω and camera position. (b) 3D representation of a configuration ω showing occlusions between parents and children. (c) A mixed graph representing all the dependencies. (d) The undirected graph.

Death (MBD) was recently introduced for MPP optimization with a major advantage, making possible multiple perturbations for the current configuration in parallel. The main idea is that at each iteration n of the algorithm, given the current configuration ω_n , we propose the addition of a new configuration ω' (multiple objects) and now we treat the new configuration $\omega = \omega_n + \omega'$ by removing non-fitting objects with certain probability. The algorithm is summarized in algorithm 1, for details, refer to [4].

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 λ
 - 5: $\omega \leftarrow \omega_{(n)} \cup \omega'$
 - 6: Death: For each $\omega_i \in \omega$, calculate the death probability $d(\omega_i) = \frac{\delta_{\alpha\beta}(\omega_i)}{1 + \delta_{\alpha\beta}(\omega_i)}$
 - 7: **until** Convergence, if not converged, set $\omega_{(n+1)} = \omega, n \rightarrow n + 1, \delta_{(n+1)} = \delta_{(n)} \times \alpha_\delta, \beta_{(n+1)} = \beta_{(n)} \times \alpha_\beta$, and go to "Birth"
-

Graph Algorithm For Death: Here we present our algorithm (algo) based on the undirected graph to accomplish the death step taking into consideration all dependencies. Our algo is described algorithm 2, with a sketched example in figure 2 for the configuration ω of figure 1. The algo starts by constructing an undirected graph G representing all the dependencies in ω . Next, the algo generate a queue Q with a

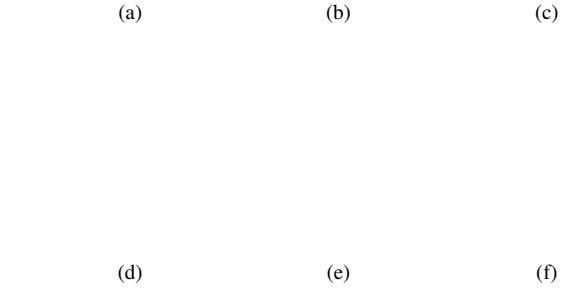


Fig. 2: (a)-(f) An example showing the evolution of a configuration and the representing graph.

topologically sorted version of G , based on a *method*. The selected *method*, define how the graph will be traversed, this affect the speed of convergence. From the many possible way to sort this graph, we propose to start first by the objects which are closer to the camera. While this method does not give the minimum number of graph partitions, but the reason for this is, given a proposed configuration ω , object that are closer to the camera, partially or fully occlude (hides) object behind them (children), this can *mislead* the evaluation (death) of their children if the algo test the children first. In step 3 and 4, we create two auxiliary queues, Q' , to hold tested objects, and their nodes color become black, and Q'' to hold blocked nodes and their color become gray. Step 5 to 24, we loop until testing every object $(1, \dots, \mathcal{N})$ in Q (in the graph G). In step 6, we start looping on element of Q , if they are not blocked, then we add them to Q' , and block the adjacent nodes. As shown in figure 2.(a), $Q = \{B, C, E, F, A, D, G, H, I, J\}$, B is first added to Q' , its color become black, and block their adjacent nodes $\{A, E, F\}$, then, we also add $\{C, G, I\}$ to Q' , their color also become black, and we also block their adjacent nodes, all blocked nodes color become gray. Now all elements of Q are either blocked or in the testing queue Q' . The delta energy associated with the proposed removal of node $\in Q'$ is calculated, then the death probability is calculated (as in algo 1 step 6), and if an element is killed, it is remove from Q' . Now the graph has to be updated after the death (removal) of some nodes, the same for Q , and Q'' has to be reset to release blocked nodes. And the algo continue as in figure2.(c), the algo test $\{E, F, J\}$, then E and J are killed. Now A and B are independent, they were only dependent by moralization when they had a common child E . Finally A and G are tested, and both of them are killed. Now a full death step of algorithm 1 is accomplished.

Algorithm 2 Death using Graph

```
1:  $G \leftarrow \text{ConstructGraph}(\omega)$ 
2:  $\mathcal{Q} \leftarrow \text{TopologicalSort}(G, \text{method})$ 
3:  $\mathcal{Q}' \leftarrow \emptyset$  (queue for tested nodes)
4:  $\mathcal{Q}'' \leftarrow \emptyset$  (queue for blocked nodes)
5: while  $\text{Sizeof}(\mathcal{Q}') \neq \mathcal{N}$  do
6:   while  $\mathcal{Q} \neq \emptyset$  do
7:      $u \leftarrow \text{Dequeue}(\mathcal{Q})$ 
8:     if  $u \in \mathcal{Q}''$  then
9:       continue
10:    end if
11:     $\mathcal{Q}' \leftarrow u$  (black color)
12:     $\text{Breadth\_Block}(G, u, \mathcal{Q}'')$  (gray color)
13:  end while
14:   $\text{CalcDeltaEnergy}(G \setminus \mathcal{Q}', G)$ 
15:  for all each  $u \in \mathcal{Q}'$ 
16:    calc death prob (algo. 1) do
17:    if  $u$  killed then
18:       $\mathcal{Q}' \leftarrow \mathcal{Q}' \setminus u$ 
19:    end if
20:  end for
21:  Set all nodes  $\in \mathcal{Q}'$  to blue
22:   $G \leftarrow \text{UpdateGraph}(\mathcal{Q} \cup \mathcal{Q}')$ 
23:   $\mathcal{Q} \leftarrow \text{TopologicalSort}(G, \text{method})$ ,  $\mathcal{Q}'' \leftarrow \emptyset$ 
24: end while
```

5. RESULTS

In this section, we present results on two synthetic images and primary results on a real image. Configuration shown in figure 3.(a) consists of 33 adult penguins and 44 chicks, and in figure 3.(b) we present the results, with 35 detected adult and 50 chick. Configuration shown in figure 3.(c) consists of 32 adult penguins and 35 chicks, and in figure 3.(d) we present the results, with 31 detected adult and 57 chick. By looking at these results, we can see that we can detect most of the partially occluded objects. This detection is correct when the occlusion is between an adult and a chick, and an over-detection inside the same type. This limitation comes from the currently naive used data term, where just one label for a dense homogeneous group, with no edge (borders) informations, it becomes very hard to make exact detection. In the real image, we have a small part of a colony image. For this image we approximate the camera parameters manually. Our model was able to detect partially the configuration, still suffering from the imperfection of the manual approximation of the camera parameters. We also get a *full 3D reconstruction* of the scene as a byproduct of the proposed algorithm.

6. CONCLUSION

In this paper we presented an effective algorithm for a very challenging problem, multiple object extraction under *occlusion* and *perspective* effects. We presented how parent-child dependency can be treated with the help of a mixed graph,

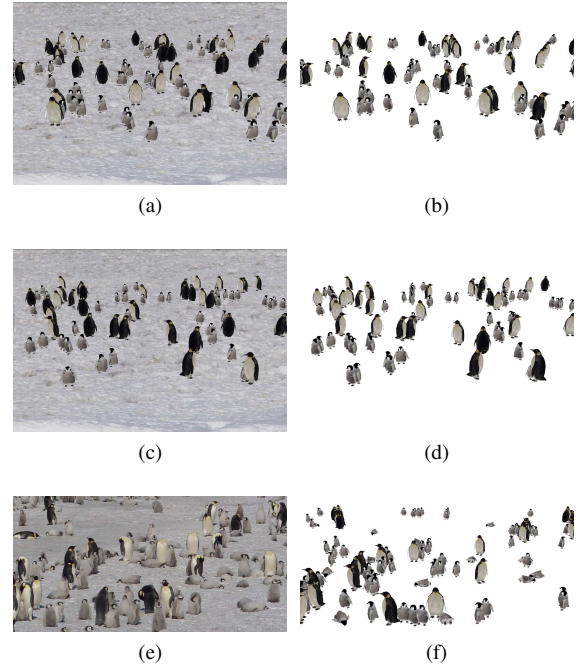


Fig. 3: (a) and (c) Shows inputs for synthetic images, (f) for real image, and (b), (d) and (e) are the results

extending the current MPP model. We also proposed the usage of a fine quality 3D geometrical model instead of usual approximation by rectangle for any type of object making advantage of the GPU, which allowed the detection to be based on accurate geometrical model for objects. The algorithm has proved to be efficient on semi-synthetic images. We are currently only getting preliminary results on real images because we only use one feature the color space reduction, also due to the manual camera parameters approximations. We are currently working on a supervised method for the data term.

7. 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] M N M van Lieshout, *Markov point processes and their applications*, Imperial College Press, 2000.
- [3] 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, mars 2008.
- [4] 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.