Depth Map Calculation for a Variable Number of Moving Objects Using Markov Sequential Object Processes

M.N.M. van Lieshout

Abstract—We advocate the use of Markov sequential object processes for tracking a variable number of moving objects through video frames with a view towards depth calculation. A regression model based on a sequential object process quantifies goodness of fit; regularization terms are incorporated to control within and between frame object interactions. We construct a Markov chain Monte Carlo method for finding the optimal tracks and associated depths and illustrate the approach on a synthetic data set as well as a sports sequence.

Index Terms—Depth calculation, Markov chain Monte Carlo, Markov sequential object process, object tracking, regularization, stochastic geometry.

1 Introduction

1.1 Motivation

TRACKING moving objects is the dual task of detecting the objects in a video sequence and following their movements through the sequence. Thus, a tracking algorithm must decide whether there are any objects of a specified kind in each of the image frames; if so, it determines the number of objects, their locations, shapes, sizes, and spatial relationships within each frame, as well as their movements across frames. In other words, for each object, a record has to be kept of the frame in which it is first seen in the observation window, its trajectory over time, and possibly the frame in which the object is last observed.

Object tracking is important, as motion is a prime source of semantic information. Applications include monitoring and surveillance, robotics, and biomedical image analysis. Our motivation comes from the need to transform plentiful 2D content to a format suitable for display on a 3D-TV in light of the dearth of "true" 3D content. To do so, our goal is to infer a depth map. Indeed, when two objects pass each other, their relative distance from the camera becomes apparent and can be propagated over frames. It is interesting to note that the occlusion that is often claimed to hinder higher order vision tasks is actually a great help in this context.

Motion tracking is a complex task, and the classical approach is to tackle easier subproblems [5], [25], [26]. In an initialization phase, the number of objects to be followed is decided upon, and their positions and velocities are measured. To deal with measurement noise, a set of equations is derived for the movement of an object from one frame to the next. These, in turn, form the basis for a Kalman or data association filtering phase that outputs cleaned, more robust object coordinates and relates these to the measurements [3], [11]. Note though that the Kalman filter can be proved to be optimal only for the prediction of the unobserved state of a linear system under Gaussian noise, a condition that rarely holds for features extracted from video data. For this reason, particle filters [6] were proposed that use a Monte Carlo approximation to the posterior probability distribution. However, the approach suffers from initialization problems when dealing with a variable number of objects [9], [26], and is not able to capture interactions between the objects [12].

An alternative to Gaussian modeling is to use the Hough transform [8], which translates complex feature recognition problems into easier to handle local peak detection problems. In tracking, the equations for the movement of an object from one frame to the next can be expressed in terms of translation and rotation parameters, evidence for which in turn is accumulated in Hough space [10]. The Hough transform is robust against noise and occlusion; its main disadvantage is the need for storage, although this may be alleviated somewhat by restriction of the parameter space.

1.2 Background and Related Work

The goal of this paper is to present a coherent theoretical framework for deriving partial depth order relations between a variable number of moving objects. In particular, we advocate the use of (sequential) spatial object processes, building on successful work on the application of stochastic geometric ideas to computer vision problems.

The idea to use Markov object processes as priors in vision can be traced back to the early 1990s [1], [2], [19], [21]. By their very nature, such models—in combination with a term for assessing the fit between the hypothesized scene and the data image(s)—allow for an unknown, variable number of objects, and may exhibit complicated interactions between objects. Moreover, there is no need for linearity or Gaussianity assumptions, and the posterior distribution quantifies the uncertainty about the validity of the hypothesis. The ideas have been taken up and applied to a variety of pattern recognition problems in fields such as confocal microscopy [23] or remote sensing [13], [24], to name but a few. Recently, [12] proposed an object process prior with a view towards the movements of a colony of ants. A sequential, data driven Metropolis-Hastings algorithm was designed, and shown to be effective in dealing with interactions between the ants, but less so in case of occlusion.

Being mostly concerned with objects that do not overlap or have a similar appearance, the abovementioned papers do not take into account the relative depths of objects in the scene. The work [17] concerned with recognition of piles of mushrooms in a single image frame did, but at high computational cost. Recently, [14], [15] introduced so-called *Markov sequential object processes*, that seem to be well suited to deal with depth ordering and occlusion because—in contrast to classical Markov object processes—they explicitly model the permutation order of the objects involved. This paper is a first study into the use of such models for depth estimation by tracking moving objects. Its plan is as follows: First, in Section 2, we fix notation. In Section 3, we propose a regression model based on a Markov sequential object process to assess the likelihood of hypothesized scenes. Section 4 introduces further regularization terms to control within and between frame object interactions. Section 5 is devoted to the construction of a Markov chain Monte Carlo method for finding the optimal tracks. Section 6 first investigates the ability of the method to deal with objects that enter or leave the scene, pass each other, or are completely occluded, as well as with varying contrast on a toy example. We then present an example concerning a sports sequence, and the paper closes with a summary and discussion of future work.

2 Preliminaries and Notation

2.1 Setup

The data consist of a sequence of image frames $\mathbf{y} = (\mathbf{y}^i; i = 1, \ldots, I)$, where $\mathbf{y}^i = (y_t^i;\ t \in T)$, $i = 1, \ldots, I$. The "*image space*" T is an arbitrary finite set of pixels, and $I \in \mathbb{N}$ is the number of frames. The observed value y_t^i at pixel $t \in T$ in frame i ranges over a set V that is arbitrary, typically $\{0, 1, \ldots, 255\}^d$ with $d = 1$ for gray-level images and $d = 3$ for colored ones. A sequence of $I = 3$ gray-level images consisting of 200×200 pixels is presented in Fig. 1.

• *The author is with the Centrum voor Wiskunde en Informatica, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands, and Eindhoven University of Technology, PO Box 4079, 1009 AB Amsterdam, The Netherlands. E-mail: colette@cwi.nl.*

Manuscript received 5 Mar. 2007; revised 8 Nov. 2007; accepted 28 Jan. 2008; published online 13 Feb. 2008.
Recommended for acceptance by P. Perez.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-03-0141.
Digital Object Identifier no. 10.1109/TPAMI.2008.45.

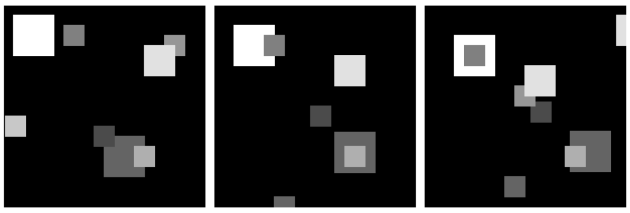


Fig. 1. Video sequence of moving squares.

Every image frame depicts a scene that contains objects of a certain type that we are interested in. Here, we suppose the "*object space*" $D \times L$ of possible objects to be a Cartesian product of location and object parameters. The set D is a compact subset of $\mathbb{R}^2$ equipped with the volume measure μ and is used to specify the location of the object; L is an arbitrary complete separable metric space (equipped with a probability distribution μ_L) that is designed to capture the geometry and appearance of an object, for example its shape, size, orientation, color, or texture. For the sequence of Fig. 1, the objects are squares, whose location may be parametrized by the top left corner. The object description is complete upon specifying its side length and color, so $L = [s_{\min}, s_{\max}] \times V$ for $0 < s_{\min} < s_{\max} < \infty$ and $V = \{0, \ldots, 255\}$, the set of gray-levels.

We assume that each object $x \in D \times L$ determines a region $R(x) \subseteq T$ in image space that is "occupied" by the object, and refer to it as the "*template*" of x in T. In the squares example, an object $x = (t_1, t_2, s, v)$ has as template $R(x)$ a square with top left pixel $(\lfloor t_1 \rfloor, \lfloor t_2 \rfloor])$ and bottom right pixel $(\lceil t_1 + s \rceil, \lceil t_2 + s \rceil)$ (alternatively, round to the nearest integer), both clipped to T if necessary, painted with gray-level v.

An "*object configuration*" is simply a finite vector of objects $\vec{\mathbf{x}} = (x_1, \cdots, x_n)$ where $x_j \in D \times L$, $j = 1, \cdots, n$, $n \geq 0$. The objects may be in any spatial relation to each other; the number of objects is variable and may be zero. An object configuration $\vec{\mathbf{x}}$ is mapped to a "*signal*" image

$$\theta_t(\vec{\mathbf{x}}) := \begin{cases} \theta(x_j) & \text{if } t \in R(x_j) \setminus \cup_{k<j} R(x_k) \\ \theta_0 & \text{if } t \in T \setminus \cup R(x_j), \end{cases} \tag{1}$$

$t \in T$, using the templates $R(x_j)$ and parameters $\theta(x_j)$ in some parameter space Θ compatible with V. The signal can be thought of as the ideal (noise-free) image. Here, θ_0 is the "background" signal; the conditions simply make sure that among the objects occupying a given pixel, the one with the smallest index determines the signal. Thus, the model explicitly accounts for occlusion, in contrast to unordered object processes [2], [12] and in a simpler way than in [17]. In our example, for $x = (t_1, t_2, s, v)$, $\theta(x) = v$ is the gray-level, but it should be noted that more complex features such as texture and blur may easily be accommodated.

2.2 Object Tracking as a Statistical Inference Problem

In order to quantify how well a sequence of object configurations $\mathbf{x} = (\vec{\mathbf{x}}^1, \ldots, \vec{\mathbf{x}}^I)$ describes a given video sequence $\mathbf{y}$, we shall formulate a probability density of the form

$$f(\mathbf{x}) \propto \exp[-U(\mathbf{x})] \tag{2}$$

whose "*energy function*"—also known as "*Hamiltonian*"—is the sum of two terms: a **regression** term to describe the fit to the data, and a **regularization** term for spatial and temporal coherence. For clarity's sake, we repress the dependence of U on data association and other terms, which shall be discussed in later sections. Having formulated a suitable probability density f, the goal is to find the mode of $f(\mathbf{x})$ by means of a Monte Carlo approach.

3 The Regression Model

Suppose Θ and V are compatible in the sense that an L_p distance can be defined between the data and signal images. Then, upon observation of the video sequence $\mathbf{y}$, write

$$U(\mathbf{x}) = \sum_{i=1}^{I} \lambda_i L_p(\mathbf{y}^i, \theta(\vec{\mathbf{x}}^i))^p \tag{3}$$

for $p \in \{1, 2\}$ and $\lambda_i > 0$. This energy function describes the "forward problem" of image formation and measures the goodness of fit between the hypothesized object configurations and the actual data sequence. In probabilistic terms, (3) amounts to assuming independent Gaussian noise at each pixel for $p = 2$, and independent Laplacian (double exponential) noise for $p = 1$ [2]. Clearly, where appropriate, other types of noise could be used instead.

Given observation of $\mathbf{y}$, we seek to minimize the energy function. Since it is a sum of individual pixel error terms, optimization of (3) over object configurations is equivalent to least squares $(p = 2)$, respectively, least absolute deviation $(p = 1)$ regression. Note however that a minimum is not guaranteed to exist, nor, if it does exist, to be unique. Indeed, spurious objects "behind" the signal of those closer to the camera (having a lower index) do not affect the energy function and may cause oversegmentation.

Except in simple cases, it is not possible to solve the regression explicitly and we resort to Monte Carlo methods. The idea of this approach is to design a Markov chain that at each transition step proposes a small change (for example, to add or delete a single object) to the current sequence of object configurations, accepts the change with a probability that depends on the improvement in energy it causes, and has the probability distribution defined by the energy function U as its limit distribution.

From a computational point of view, it is highly desirable that the transitions are easy to implement. Consider the potential energy

$$\lambda_i \sum_{t \in R(\xi) \setminus \cup_k R(x_k^i)} \left[|y_t^i - \theta(\xi)|^p - |y_t^i - \theta_0|^p \right] \tag{4}$$

required for adding object ξ to $\vec{\mathbf{x}}^i$ in frame i to obtain the vector $(\vec{\mathbf{x}}^i, \xi)$. Equation (4) depends only on $R(\xi)$ and those templates $R(x_k^i)$ that overlap $R(\xi)$, so that the computation involves *local* knowledge only. Note that, if ξ were added at position k, (4) would be replaced by

$$\sum_{t \in R(\xi) \setminus \cup_{l<k} R(x_l^i)} \left[|y_t^i - \theta(\xi)|^p - |y_t^i - \theta_t(\vec{\mathbf{x}}^i)|^p \right].$$

As $R(\xi) \setminus \cup_{l<k} R(x_l^i) = R(\xi) \setminus \bigcup_{l<k : R(x_l^i) \cap R(\xi) \neq \emptyset} R(x_l^i)$, the potential energy above does not depend on those $R(x_l^i)$ with $l < k$ that do not overlap $R(\xi)$. Its second term involves only $R(x_l^i)$ with $l \geq k$, and then only those that overlap $R(\xi)$. Thus, the role of objects with $l < k$ and those with $l \geq k$ is different: The first determine the set of pixels over which to take the sum, the second ones' signal value is used. In mathematical terms, the above considerations amount to saying that each single frame energy function defines a "*Markov sequential object process*" [15] with respect to the "*overlapping objects relation*" $\xi \sim \eta \Leftrightarrow R(\xi) \cap R(\eta) \neq \emptyset$, $\xi, \eta \in D \times L$.

4 Regularization

Since the energy function (3) depends on $\mathbf{x}$ through the signal images only, it does not allow for data association between objects in adjacent frames and is prone to oversegmentation as one may add any number of objects hidden behind those in an optimal configuration without affecting the optimality. To overcome these shortcomings, we add terms to the energy function that prevent overlap (V_2), favor temporal cohesion between objects in subsequent frames, and include matchings to keep track of an object as it moves across the image frames (V_3).

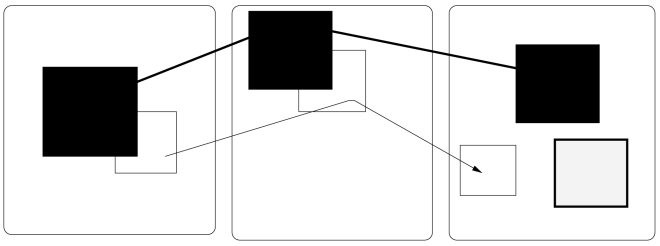


Fig. 2. Data association by matching.

4.1 Within Frame Interaction

As in object recognition, in order to avoid overfitting, a natural condition for V_2 is to impose Markovianity with respect to the overlapping objects relation. In this paper, we use the "*Strauss*" potential given by

$$V_2(\vec{\mathbf{x}}) = \beta n(\vec{\mathbf{x}}) + \gamma n_o(\vec{\mathbf{x}}),$$

where $n(\vec{\mathbf{x}})$ is the length of the object sequence $\vec{\mathbf{x}}$, and $n_o(\vec{\mathbf{x}})$ is the number of pairs $\{\xi, \eta\}$ in $\vec{\mathbf{x}}$ for which $R(\xi) \cap R(\eta) \neq \emptyset$. The parameter β is a real, γ is positive. Clearly, if we add a new object ξ to the sequence, the required potential energy

$$V_2((\vec{\mathbf{x}}, \xi)) - V_2(\vec{\mathbf{x}}) = \beta + \gamma \#\{x_j \in \vec{\mathbf{x}} : R(\xi) \cap R(x_j) \neq \emptyset\}$$

depends only on those existing objects whose template overlaps $R(\xi)$. Therefore, such updates are local operations.

More generally, one might use a pairwise interaction model [20]

$$V_2(\vec{\mathbf{x}}) = \sum_k \beta(x_k) + \sum_{k<l} \varphi(x_k, x_l)$$

for some symmetric function $\varphi(\cdot, \cdot) \geq 0$, and intensity function $\beta(\cdot)$ that could, e.g., penalize colors close to the background, or favor large objects over small ones.

4.2 Propagation over Frames

In the previous section, we defined inhibition between the objects in a single frame. Between frames, we would like to have attraction, that is, temporal cohesion. Let $S_{m,n}$ be the set

$$\{(M, N, \pi) : M \subseteq \{1, \ldots, m\}; N \subseteq \{1, \ldots, n\}; |M| = |N|\}$$

with m, $n \in \mathbb{N}_0$, and $\pi : M \to N$ a bijection. Given an $s \in S_{m,n}$, the components are denoted by $M(s)$, $N(s)$, and $\pi(s)$, respectively, [16]. Here, m is the number of objects in some image frame, n that in the next frame. The sets M and N contain the indices of the objects that are present in both frames in, respectively, the first and second frame. The bijection π provides the data association, that is, the object with index $i \in M$ in the first image frame is identified with that having index $\pi(i) \in N$ in the consecutive frame. We shall refer to s as a "*matching*." For a graphical representation, refer to Fig. 2, where the matchings $s \in S_{2,2}$ and $s' \in S_{2,3}$ are indicated by arrows, and $|M(s)| = |N(s)| = 2 = |M(s')| = |N(s')|$.

For two configurations $\vec{\mathbf{x}}^i$, $\vec{\mathbf{x}}^{i+1}$ in consecutive frames, and $s^{i,i+1} \in S_{n(\vec{\mathbf{x}}^i), n(\vec{\mathbf{x}}^{i+1})}$, we define $V_3(\vec{\mathbf{x}}^i, \vec{\mathbf{x}}^{i+1}, s^{i,i+1})$ to be

$$\begin{aligned}
&\sum_{l \in M(s^{i,i+1})} \tau(\vec{\mathbf{x}}^i_l, \vec{\mathbf{x}}^{i+1}_{\pi(s^{i,i+1})(l)}) + \\
&\sum_{l \notin M(s^{i,i+1})} \lambda(\vec{\mathbf{x}}^i_l) + \sum_{l \notin N(s^{i,i+1})} \lambda(\vec{\mathbf{x}}^{i+1}_l) + \\
&\sum_{x_l \sim x_k \in \vec{\mathbf{x}}^i; l<k \in M(s^{i,i+1})} \rho \mathbf{1}\Big\{\pi(s^{i,i+1})(l) > \pi(s^{i,i+1})(k)\Big\} + \\
&\sum_{x_l \sim x_k \in \vec{\mathbf{x}}^{i+1}; l<k \in N(s^{i,i+1})} \rho \mathbf{1}\Big\{\pi^{-1}(s^{i,i+1})(l) > \pi^{-1}(s^{i,i+1})(k)\Big\}.
\end{aligned} \tag{5}$$

The positive valued function $\lambda(\cdot)$ penalizes unmatched objects, whereas the symmetric positive valued function $\tau(\cdot, \cdot)$ quantifies the dissimilarity between its arguments. The parameter $\rho \geq 0$ is intended to propagate relative depth information gathered when objects overlap over time.

In summary, we obtain a total energy function $U(\mathbf{x}; \mathbf{s}; \mathbf{y})$ as the sum

$$\lambda_1 \sum_{i=1}^{I} V_1(\mathbf{y}^i | \vec{\mathbf{x}}^i) + \lambda_2 \sum_{i=1}^{I} V_2(\vec{\mathbf{x}}^i) + \lambda_3 \sum_{i=1}^{I-1} V_3(\vec{\mathbf{x}}^i, \vec{\mathbf{x}}^{i+1}, s^{i,i+1}) \tag{6}$$

of the ingredients discussed separately above, where $\lambda_i > 0$ and $V_1(\mathbf{y}^i | \vec{\mathbf{x}}^i) = L_p(\mathbf{y}^i, \theta(\vec{\mathbf{x}}^i))^p$, cf. (3).

5 Metropolis-Hastings Sampler

Our goal is to find the optimal configuration sequence with matchings, in the sense of minimizing the energy function (6). We shall do this by simulated annealing within the Metropolis-Hastings framework [4], [7], [15], i.e., we multiply the energy function by a series of non-negative constants increasing to infinity (usually referred to as "*inverse temperature*"). The probability distributions so defined are close to a uniform one for small constants, whereas for large inverse temperatures the distribution becomes peaked around the energy minimizers.

Some care has to be taken in designing the Metropolis-Hastings chain. One must make sure that any vector of object configurations and any matching between frames can be reached in a finite number of steps from any other such vector with matchings, that there are no cycles, that changes in dimension due to addition or deletion of objects are properly handled, and that the set of update proposals is rich enough to allow for an efficient exploration of the state space. Inspired by [16], we propose the following types of update proposals:

- addition of a singly matched object;
- addition of a doubly matched object;
- addition of an unmatched object;
- deletion of a singly matched object;
- deletion of a doubly matched object;
- deletion of an unmatched object;
- modification of the permutation order;
- addition of a match;
- deletion of a match;
- modification of an object (its location, color, size, etc.).

As an example, suppose we propose to add an unmatched object. First, a frame is selected uniformly from the set $\{1, \ldots, I\}$. Then, if frame $i \in \{1, \ldots, I\}$ is chosen with current configuration $\vec{\mathbf{x}}^i$, generate a new object according to $(\mu \times \mu_L)(D \times L)^{-1} d(\mu \times \mu_L)(\xi)$ and insert it into the object sequence $\vec{\mathbf{x}}^i$ at a uniformly chosen position j to obtain $c_j(\vec{\mathbf{x}}^i, \xi)$. The indices for the matchings involving frame i are adjusted and denoted by $c_j(s^{i-1,i}, \xi)$ and $c_j(s^{i,i+1}, \xi)$. Finally, accept the move with probability

$$\frac{(\mu \times \mu_l)(D \times L)\ \exp[-U(\mathbf{x}'; \mathbf{s}'; \mathbf{y}) + U(\mathbf{x}; \mathbf{s}; \mathbf{y})]}{n(\vec{\mathbf{x}}^i) + 1 - |N(s^{i-1,i}) \cup M(s^{i,i+1})|}$$

truncated at 1. Here, the new sequence $\mathbf{x}'$ differs from $\mathbf{x}$ only in frame i with $\vec{\mathbf{x}}^{i\prime} = c_j(\vec{\mathbf{x}}^i, \xi)$, and $\mathbf{s}'$ is identical to $\mathbf{s}$ except for the matchings involving frame i which are replaced by $c_j(s^{i-1,i}, \xi)$ and $c_j(s^{i,i+1}, \xi)$. For the first and last frame, one of the matchings does not exist and is not adjusted.

The other updates are described in full detail in the Appendix, which can be found at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.45/, where a proof of the validity of the method is given as well. From a computational point of view, the Markov properties discussed in Sections 3 and 4 imply a local computational effort.

TABLE 1
Result after Simulated Annealing (See Text) for Fig. 1

Frame 1				
top left		side	color	match
89.18	119.10	20.19	75	7
129.10	138.91	19.98	175	6
1.31	108.52	20.31	200	
98.86	129.13	39.97	100	8
138.90	38.94	29.92	225	1
59.40	18.89	20.38	128	2
9.38	8.75	39.94	255	4
158.63	29.04	19.75	150	5

Frame 2				
top left		side	color	match
118.78	48.57	29.91	225	1
49.18	29.04	20.39	128	4
59.33	188.90	20.06	100	9
19.30	19.48	39.67	255	8
124.57	56.69	19.72	150	2
128.59	138.96	19.91	175	5
94.96	99.25	20.30	75	7
119.09	124.67	40.05	100	6

Frame 3			
top left		side	color
98.85	58.90	29.87	225
89.02	78.95	19.70	150
189.76	8.83	29.74	225
39.24	39.21	20.50	128
139.14	139.38	19.94	175
143.81	124.24	40.08	100
104.64	94.72	20.27	75
28.67	28.65	39.65	255
78.86	169.46	20.00	100

TABLE 2
Pairwise Probabilities p_{ij}^k of Object i Having a Lower Sequence Index than Object j after Annealing (See Text) for Frames $k = 1, 2, 3$ (From Top to Bottom) and Data as Given in Fig. 1

-	0.50	0.63	1.00	0.45	0.45	0.80	0.80
0.50	-	0.62	1.00	0.45	0.45	0.80	0.80
0.37	0.38	-	0.75	0.33	0.34	0.67	0.66
0.00	0.00	0.25	-	0.10	0.10	0.40	0.40
0.55	0.55	0.67	0.90	-	0.50	0.83	1.00
0.55	0.55	0.66	0.90	0.50	-	1.00	0.83
0.20	0.20	0.33	0.60	0.17	0.00	-	0.50
0.20	0.20	0.34	0.60	0.00	0.17	0.50	-

-	0.66	0.79	0.93	1.00	0.75	1.00	1.00
0.34	-	0.67	1.00	0.62	0.59	0.82	0.95
0.21	0.33	-	0.67	0.42	0.42	0.62	0.83
0.07	0.00	0.33	-	0.22	0.24	0.43	0.71
0.00	0.38	0.58	0.78	-	0.50	1.00	1.00
0.25	0.41	0.58	0.76	0.50	-	0.75	1.00
0.00	0.18	0.38	0.57	0.00	0.25	-	1.00
0.00	0.05	0.17	0.29	0.00	0.00	0.00	-

-	1.00	0.75	0.60	0.60	0.90	1.00	0.90	0.75
0.00	-	0.50	0.30	0.30	0.70	1.00	0.70	0.50
0.25	0.50	-	0.33	0.33	0.66	0.75	0.67	0.50
0.40	0.70	0.67	-	0.50	0.83	0.90	1.00	0.67
0.40	0.70	0.67	0.50	-	1.00	0.90	0.84	0.67
0.10	0.30	0.34	0.17	0.00	-	0.60	0.51	0.33
0.00	0.00	0.25	0.10	0.10	0.40	-	0.40	0.25
0.10	0.30	0.33	0.00	0.16	0.49	0.60	-	0.33
0.25	0.50	0.50	0.33	0.33	0.67	0.75	0.67	-

6 Examples

6.1 Synthetic Example

In order to assess the efficacy of the method proposed in this paper, first consider the synthetic data presented in Fig. 1. For this scene, the ground truth is known, allowing us to verify the result obtained by the method. The scene was chosen to include objects entering and leaving the image frame or passing each other, a square leaving one connected component to join another, complete occlusion, as well as varying contrast.

We model the squares as in Section 2 with $s_{\min} = 5.0$ and $s_{\max} = 45.0$. In the absence of noise, the robust Laplace criterion is used to evaluate the goodness of fit with dispersion parameter 0.05. A penalty of 10.0 is added for each square, a penalty of 30.0 for each pair of overlapping objects. Each missing match contributes a factor of $\lambda(\cdot) \equiv 40.0$ to V_3; the dissimilarity $\tau(x_1, x_2)$ is the sum of the absolute difference in side lengths, 10 times the absolute gray-level difference and $\|l_1 - l_2\|^2/150.0$, where l_j denotes the top left point of x_j; setting $\rho = 10.0$ completes the model.

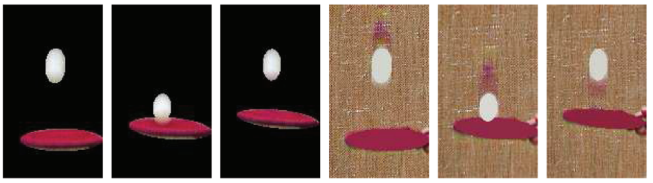
We ran the Metropolis-Hastings algorithm outlined in Section 5 with two additional object specific updates: splitting a current object in two, and merging two close objects together into a single new one. Specifically, we propose to merge two uniformly chosen squares in a uniformly chosen frame into their "convex hull," i.e., the smallest square that contains both, provided neither of the two objects is contained in the other and the side length of the new square falls within the model range. The new square's color, matches, and position in the sequence are those of either of the original squares with probability 1/2 each. The proposal is accepted according to the Hastings ratio. Reversely, to split a uniformly chosen object in a uniformly chosen frame, at one of its four uniformly selected corners a new object is placed with side length chosen uniformly conditioned on the event of not exceeding that of the object to be split. Then, an interval is selected on either of the opposite sides (probability 1/2 each) to define the second object by a uniformly chosen location and side length distributed as that of the first object. One of the new squares is colored, matched and placed as the original; the other is colored according to the reference measure, placed uniformly at random into the sequence, and receives matches uniformly chosen from amongst the available options. Again, the Hastings ratio determines the acceptance probability of the split move thus described.

After a heating and burn-in phase of in total 100,000 updates, annealing was carried out at temperatures $T_n = 1.0/(1.0 + 0.005n)$, $0 \le n < 1,500$. The signal of the near optimal sequence of configurations at temperature 0.118 is exactly that of the data and given in Fig. 1. Not visible in the figure, but present in the near optimal configuration is a square hidden behind the light one in the top right quadrant of the middle frame, see the listed sequence of object configurations with associated matchings in Table 1. The matches are correctly reproduced. The energy function takes value 723.8 which should be compared to 723.3 for the ground truth. The difference is due to slight variations on the subpixel level which translate in slightly different dissimilarity values. Implemented in C++, on a state of the art Linux platform, the algorithm managed about 5,700 iterations per minute.

In order to quantify the depth order, Table 2 lists the probabilities p_{ij}^k of object i in frame k lying closer to the camera than square $j \neq i$ in the same frame, estimated over a further 50,000 states obtained by proposing permutation updates only and subsampling each 100 steps at temperature 0.118. By standard combinatorial arguments, the correctness of these empirical values may be verified. For example, for the ground truth, $p_{13}^1 = 5/8$ as in five of the eight possible permutations of the connected component that includes the squares with indices 1 and 3 in Table 1 consistent with the data, object 1 has a smaller index than object 3.

6.2 Table Tennis Sequence

To test the method on real data, we study an example in sports tracking: the ball and bat in a table tennis sequence (see Fig. 3). Both objects of interest can be conveniently described mathematically by a colored ellipse with three shape parameters: the half lengths of



(a) (b)

Fig. 3. Data masked by (a) annealed object sequence and (b) annealed object sequence overlaid upon the data.

both axes and the orientation. In other words, $L = [a_{\min}, a_{\max}] \times [b_{\min}, b_{\max}] \times [0, \pi] \times \{0, \ldots, 255\}^3$, where the first two components correspond to the ellipse minor and major axis, respectively, the third one to its orientation; the three discrete components specify the ellipse's RGB color.

We set $a_{\min} = b_{\min} = 5.0$ for the minimum half axis length, $a_{\max} = b_{\max} = 40.0$ for the maximum one. The color of an ellipse lies in a discrete RGB space equipped with the equal weight mixture of data frame histograms, i.e., the probability distribution that chooses each frame with equal probability and then samples a nonbackground color according to the histogram of that frame.

Regarding the parameters in model (6), the background RGB component values are taken to be the marginal modes of the color histogram of the first data frame. Note that very similar values would be obtained from any other data frame. In V_1, an L_2 criterion is used with $\sigma = 128$ in Gaussian noise terms. The function V_2 adds a penalty 50 for each object, five for each pair of overlapping ellipses. In V_3, let $\lambda(\cdot) \equiv 5 = \rho$ be constant; the dissimilarity term $\tau(x_1, x_2)$ is the sum of the absolute differences in half axes lengths and orientation (modulo π), the normalized absolute differences in RGB space with normalization $1/255$, and the squared Euclidean distance between centers divided by 800.

After a burn-in of 30,000 Metropolis-Hastings steps, annealing was performed for temperatures $T_n = 1.0/(1 + 0.005 * n)$ with 50 steps for each $n = 0, \ldots, 1,000$. The near-optimal configuration thus obtained is depicted in Fig. 3. The correct permutation is found that places the bat under the ball in each frame. If we would not have included a temporal cohesion term, bat and ball could have been ordered in each of the two possible ways with probability $1/2$ in the first and third frame. Note that the slight deviations from an elliptic shape in the bat are due to perspective. The computation took 6.5 minutes, which amounts to about 200 iterations per second.

7 Summary and Future Work

In this paper, we presented an application of Markov sequential object processes to the calculation of depth maps for scenes involving a variable number of interacting objects that may change over time with a view to 3D-TV. The model proposed here is able to cope with the occlusion caused by having objects at different depths, maintains the identity of objects as well as their relative depth over consecutive video frames, and ensures fit to the data. The computational complexity of the model can be handled by a suitably designed Metropolis-Hastings algorithm. In contrast to commonly used filtering methods, the sampler goes back and forth between frames, gathering depth information when objects overlap and transferring this information on to adjacent frames that do not provide depth cues. As most interest focuses on the optimal relative depth probabilities, a simulated annealing scheme may be used. The approach was illustrated on a toy example and a real life table tennis video sequence.

This work concentrated on objects that are described by a few shape parameters. However, the theoretical framework presented here is not limited to such cases, indeed includes, e.g., polygons of arbitrary shape, or even completely general closed sets [18]. In the future, we intend to formalize such a segmentation-based approach and evaluate its effectiveness for scenes that are not composed of simple objects against a homogeneous background.

Acknowledgments

This research is supported by the Technology Foundation STW, the applied science division of NWO, and the technology program of the Ministry of Economic Affairs (project CWI.6156 "Markov sequential point processes for image analysis and statistical physics"). The author would like to thank Dr. C. Varekamp (Philips Research) for data and interesting discussions, and A.G. Steenbeek for expert programming assistance.

References

[1] A.J. Baddeley and M.N.M. van Lieshout, "Object Recognition Using Markov Spatial Processes," *Proc. 11th IAPR Int'l Conf. Pattern Recognition*, pp. B136-B139, 1992.
[2] A.J. Baddeley and M.N.M. van Lieshout, "Stochastic Geometry Models in High-Level Vision," *Statistics and Images, Vol. 1*, vol. 20, K.V. Mardia and G.K. Kanji, eds., Advances in Applied Statistics, a supplement to *J. Applied Statistics*, pp. 231-256, 1993.
[3] R.L. Eubank, *A Kalman Filter Primer*. Chapman & Hall/CRC, 2006.
[4] C.J. Geyer and J. Møller, "Simulation Procedures and Likelihood Inference for Spatial Point Processes," *Scandinavian J. Statistics*, vol. 21, pp. 359-373, 1994.
[5] I.R. Goodman, R.P.S. Mahler, and H.T. Nguyen, *Mathematics of Data Fusion*, Vol. 39 of Series B: Mathematical and Statistical Methods, Kluwer, 1997.
[6] N. Gordon, D. Salmond, and A. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE Proc.*, vol. 140, pp. 107-113, 1993.
[7] P.J. Green, "Reversible Jump MCMC Computation and Bayesian Model Determination," *Biometrika*, vol. 82, pp. 711-732, 1995.
[8] P.V.C. Hough, "Method and Means for Recognizing Complex Patterns," US Patent 3069654, 1962.
[9] C. Hue, J.-P. Le Cadre, and P. Pérez, "Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion," *IEEE Trans. Signal Processing*, vol. 50, pp. 309-325, 2002.
[10] J. Illingworth and J. Kittler, "A Survey of the Hough Transform," *Computer Vision, Graphics, and Image Processing*, vol. 44, pp. 87-116, 1988.
[11] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *J. Basic Eng.*, vol. 82, pp. 35-45, 1960.
[12] Z. Khan, T. Balch, and F. Dellaert, "MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1805-1819, 2005.
[13] C. Lacoste, X. Descombes, and J. Zerubia, "Point Processes for Unsupervised Line Network Extraction in Remote Sensing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1568-1579, 2005.
[14] M.N.M. van Lieshout, "Markovianity in Space and Time," *Dynamics & Stochastics: Fest-Schrift in honor of M.S. Keane*, D. Denteneer, F. den Hollander, and E. Verbitskiy, eds., Lecture Notes—Monograph Series, Inst. for Math. Statistics, vol. 48, pp. 154-168, 2006.
[15] M.N.M. van Lieshout, "Campbell and Moment Measures for Finite Sequential Spatial Processes," *Proc. Prague Stochastics 2006*, M. Hušková and M. Janžra, eds., pp. 215-224, 2006.
[16] J. Lund, A. Penttinen, and M. Rudemo, "Bayesian Analysis of Spatial Point Patterns from Noisy Observations," research report, Dept. of Math. and Physics, The Royal Veterinary and Agricultural Univ., 1999.
[17] K.V. Mardia, W. Qian, D. Shah, and K.M.A. Desouza, "Deformable Template Recognition of Multiple Occluded Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 9, pp. 1035-1042, Sept. 1997.
[18] G. Matheron, *Random Sets and Integral Geometry*. John Wiley and Sons, 1975.
[19] R. Molina and B.D. Ripley, "Using Spatial Models as Priors in Astronomical Image Analysis," *J. Applied Statistics*, vol. 16, pp. 193-206, 1989.
[20] B.D. Ripley and F.P. Kelly, "Markov Point Processes," *J. London Math. Soc.*, vol. 15, pp. 188-192, 1977.
[21] B.D. Ripley and A.I. Sutherland, "Finding Spiral Structures in Images of Galaxies," *Philosophical Trans. Royal Soc. London, Series A*, vol. 332, pp. 477-485, 1990.
[22] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 1999.
[23] H. Rue and M.A. Hurn, "Bayesian Object Identification," *Biometrika*, vol. 86, pp. 649-660, 1999.
[24] R. Stoica, X. Descombes, and J. Zerubia, "A Gibbs Point Process for Road Extraction in Remotely Sensed Images," *Int'l J. Computer Vision*, vol. 57, pp. 121-136, 2004.
[25] L.D. Stone, C.A. Barlow, and T.L. Corwin, *Bayesian Multiple Target Tracking*. Artech House, 1999.
[26] M. Vihola, "Random Sets for Multitarget Tracking and Data Fusion," licentiate thesis, Tampere Univ. of Technology, 2004.