Approximate Bayesian methods for kernel-based object tracking

Zoran Zivkovic, Ali Taylan Cemgil, Ben Kröse

Intelligent Systems Lab Amsterdam
University of Amsterdam, The Netherlands
email:{zivkovic,cemgil,krose}@science.uva.nl

Abstract

We present a framework for real-time tracking of complex non-rigid objects. The shape of the object is approximated by an ellipse and its appearance by histogram based features derived from local image properties. We use an efficient local search scheme (based on mean-shift) to find the image region with a histogram most similar to the histogram of the tracked object. The efficient search can be integrated into a Bayesian filtering scheme. We compare a number of schemes: the Kalman filter, the mixture Kalman filter and other sequential importance sampling (particle filtering) techniques.

Key words: object tracking, approximate Bayesian filtering
PACS:

1 Introduction

In broad terms, Bayesian approaches to object tracking rely on two main components: a transition (object motion) model, that describes kinematic constrains on the evolution of the objects state, and an observation model that defines the likelihood of the object configuration given current measurements. In principle, once a model is decided upon, tracking boils down to posterior inference that can be carried out recursively using some Bayesian filtering scheme [1,8]. In this paper we will address the following issues that are often raised in the context of visual object tracking.

The first issue is the challenging problem of defining a realistic yet practical observation model for non-rigid objects in vision based tracking. Some approaches explicitly model the relation between the state of the object and the

appearance of each pixel from the image region occupied by the tracked object: for example models tailored specifically for humans [9] or more generic models to be learned from data [17,16]. Contour based algorithms [6,13] focus on detailed modelling but only of the outer contour shape of the tracked object. An alternative line of approach employs appearance models robust to deformations: for example the histogram-based representation [22] or extensions [3,11,23,12,21]. In this paper we follow [7] where the shape of the tracked object is approximated by an ellipse and the appearance within the ellipse is described by a histogram based model. The obvious advantage of such a model is its simplicity and general applicability. Another advantage, that made this observation model rather popular, is the existence of efficient local search schemes to find the image region with a histogram most similar to the histogram of the tracked object [10,7,19,24]. We will use the natural extension of the mean-shift procedure from [24] that efficiently solves previous problems with sudden object scale and shape changes.

The second issue is regarding the intractability of exact Bayesian filtering due to the complex observation model: it is not obvious how to choose among many possible approximate techniques. This paper analyzes and compares a range of approximate Bayesian tracking schemes listed below. First, the efficient local search [24] is used to find the likely object configuration, the complex observation model is summarized by local Gaussian (Laplace) approximation and the Kalman filter is used as in [7]. Often there are several likely object configurations. The local search can be used to find these configurations by starting the search from various random starting points. The observation model is then approximated by a Gaussian mixture [2] and the mixture Kalman filter [4] can be used. Another approach is is using a sampling scheme. While the bootstrap particle filter is an obvious candidate [13,18], it makes not use of the search scheme. In this paper, we illustrate how the Gaussian mixture approximation of the observation model obtained using the search scheme can be used within a sampling scheme as a proposal distribution. In this way, we obtain the computational advantages without compromising theoretical convergence properties of a particle filter [8].

The paper is organized as follows: In Section 2, we introduce the observation model and the efficient local search scheme based on the results from [24]. The similarity between a histogram from an elliptical image region with the histogram of the tracked object is formulated as a probability model. This presents an obvious generalization of similarity measures that have been used previously [7,11,24], but nevertheless allows us to combine features derived from different image modalities. In Section 3, we discuss how the search procedure can be used within a range of Bayesian tracking schemes. Experiments are given in Section 4.

2 Observation model

In this section we define a probability model that relates the state s_t of an object at time t with the video frame I_t observed at time t. Throughout the paper the index $t = 1 \dots T$ denotes the discrete time (frame) index. Occasionally, when the time index is not relevant we will omit it. We will denote the value at the i'th pixel by $I_t(x_i)$. Here, x_i denotes the image pixel location.

2.1 Object shape

Suppose we are given an arbitrary shape $\mathcal{S}$ in an image specified by a set of pixel locations x_i, i.e., $\mathcal{S} \equiv \{x_i : i\text{'th pixel belongs to the object}\}$. We approximate the shape of a non-rigid object in an image by its first and second order moments – an elliptical region we denote by $\mathcal{S}^e$. The original shape $\mathcal{S}$ may have been initially selected manually or detected using some other algorithm, for example background subtraction [20]. If there are $N_{\mathcal{S}}$ pixels that belong to the object of interest, we define

$$\theta \equiv \frac{1}{N_{\mathcal{S}}} \sum_{x_i \in \mathcal{S}} x_i \text{ and } V \equiv \frac{1}{N_{\mathcal{S}}} \sum_{x_i \in \mathcal{S}} (x_i - \theta)(x_i - \theta)^T. \quad (1)$$

Here, the first moment vector θ denotes the center of the object in the image I. The matrix of second moments V, that encodes scale and orientation, is symmetric and positive definite. Consequently, the θ and V describe an arbitrary elliptical region. We use here the following parametrization $s \equiv [\theta^T, scale_x, scale_y, skew]^T$ where $scale_x$ and $scale_y$ are the scaling and $skew$ is the skew transformation obtained from from V using the unique Cholesky factorization $V = \begin{bmatrix} scale_x & skew \\ 0 & scale_y \end{bmatrix}^T \begin{bmatrix} scale_x & skew \\ 0 & scale_y \end{bmatrix}$. Occasionally, by a slight abuse of notation we will refer to the state s as $s = (\theta, V)$ to explicitly highlight the dependence on θ and V. Similarly, $\mathcal{S}^e(s)$ will denote the elliptical shape defined by s.

2.2 Object appearance using histogram based features

The appearance of an object is described by a set of M scalar features $r_1, ..., r_M$ that are extracted from the local area of an image I defined by $\mathcal{S}^e(s)$. We view each r_m as a "bin" of a histogram. Let $\mathbb{P}$ be the set of pixel values $I(x_i)$, for example $\mathbb{P} = [0, 255]^3$ for RGB images. We define a quantization function

$b : \mathbb{P} \rightarrow \{1 \dots M\}$, that associates with each observed pixel value a particular bin index m.

The value r_m of the m-th bin is calculated from the elliptical image region $\mathcal{S}^e(s = (\theta, V))$ using:

$$r_m(I, s) \equiv |V|^{\gamma/2} \sum_{x_i \in \mathcal{S}^e(s)} \mathcal{N}(x_i; \theta, V) \;\; \delta\left[b(I(x_i)) - m\right], \tag{2}$$

where δ is the Kronecker delta function. The kernel function $\mathcal{N}$ is chosen such that pixels in the middle of the object have higher weights than pixels at the borders of the objects. A natural choice is a Gaussian kernel defined by: $\mathcal{N}(x; \theta, V) = |2\pi V|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\theta)^T V^{-1}(x-\theta)\right)$. The prefactor $|V|^{\gamma/2}$ in (2) discounts for the fact that in practice we use only the N_s pixels from a finite neighborhood of the kernel center. We disregard samples further than 2.5-sigma and it is easy to show that one should use $\gamma \approx 0.1$ in this case. The smooth kernel function will suppress the influence of the (arguably less reliable) pixels near the borders. But more importantly it enables a fast gradient based search described at the end of this section.

2.3 Probabilistic observation model

We introduce for each feature r_m a probability density function $p(r_m(I, s))$, the particular form to be defined later. We assume that the features r_m are independent. Furthermore, we assume that each feature r_m is uninformative if computed outside the region defined by s. The log-likelihood of s in an image I can be defined by:

$$\log \mathcal{L}(s) = \log p(I|s) \propto \sum_{m=1}^{M} \log p(r_m(I, s)). \tag{3}$$

This likelihood function for an image frame I_t is the observation model. The likelihood can be viewed as a generalization of many different histogram similarity measures that are used in literature. For example if $p(r_m(I, s))$ is chosen to be a Gaussian $\mathcal{N}(r_m(I, s); o_m, \sigma^2)$, the log-likelihood becomes the sum of squared distances as in [11]. The mean o_m and the standard deviation σ can be estimated from a set of test images of the object. The often used Bhattacharyya coefficient based model [18,2]: $p(I|s) \propto \exp(\sum_{m=1}^{M} \sqrt{r_m(I, s)}\sqrt{o_m}/\sigma^2)$ can also be seen as a particular choice.

2.4 Additional features

In our experiments, we have used videos from a static camera. Therefore, we can use the features from a simple background/foreground segmentation scheme similar to [20]. We view the result of the background/foreground segmentation as an additional and independent observed image $\tilde{I}$ where $\tilde{I}(x_i) \in \{0,1\} = \tilde{\mathbb{P}}$. We define a new quantization function $\tilde{b} : \tilde{\mathbb{P}} \rightarrow \{1,2\}$ where $\tilde{m} = 1,2$ denotes, say, background and foreground. We define a new set of features $\tilde{r}_{\tilde{m}}(\tilde{I}, s)$ as in (2). Similarly, we define $p(\tilde{I}|s)$ by defining $\tilde{o}$ and $\tilde{\sigma}$. Intuitively, this latter feature measures the ratio of background pixels to the foreground pixels in the elliptic region. Due to independent observation assumption, the contributions to the likelihood function will be additive, i.e., $\log \mathcal{L}(s) = \log p(I|s) + \log p(\tilde{I}|s)$. Clearly, the set of features could be extended further: normalized color values, optical flow results, etc. Choosing the particular type of local image property, i.e., feature selection, is not the focus of this paper as this highly depends upon the situation [5].

2.5 Iterative search for the most likely configuration

We propose here an efficient and specialized gradient descent procedure to search for the likely object configurations (3) given an image I. The local search starts with some starting point $s^{\{k\}}$. Here the superscript $\{k\}$ denotes the iteration index. Similar to [7,11,23,12,21] the gradient descent step is calculated using two stages that are repeated iteratively: first the similarity measure (3) is approximated locally using a Taylor expansion, then the gradient step is calculated.

The Taylor expansion of (3)(in r_m around $r_m(I, s^{\{k\}})$) is:

$$\log \mathcal{L}(s) \approx c + \sum_{m=1}^{M} \frac{p'(r_m(I, s^{\{k\}}))}{p(r_m(I, s^{\{k\}}))} r_m(I, s) \tag{4}$$

where c is a constant term. We denote the variable term from above by $f(s)$ and replace (2):

$$f(s) = \sum_{m=1}^{M} \frac{p'(r_m(I, s^{\{k\}}))}{p(r_m(I, s^{\{k\}}))} |V|^{\gamma/2} \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \mathcal{N}(x_i; \theta, V) \delta\left[b(I(x_i)) - m\right] = |V|^{\gamma/2} \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \omega_i \mathcal{N}(x_i; \theta, V) \tag{5}$$

where

$$\omega_i = \sum_{m=1}^{M} \frac{p'(r_m(I, s^{\{k\}}))}{p(r_m(I, s^{\{k\}}))} \delta\left[b(I(x_i)) - m\right]. \tag{6}$$

For the Bhattacharyya coefficient metric [7], which can be seen as a particular choice, we have:

$$\omega_i = \sum_{m=1}^{M} \sqrt{\frac{o_m}{r_m(I, s^{\{k\}})}} \delta\left[b(I(x_i)) - m\right]. \tag{7}$$

The functional form (5) resembles a kernel based density estimate. The mean-shift algorithm [15] can be used to calculate the gradient step on (5) with respect to the object position θ. In the next iteration the above approximation is repeated for the new position $\theta^{\{k+1\}}$ and a new gradient step is calculated as in [7]. Instead of the mean-shift we use the extended version (for Gaussian kernels) [24] to get the gradient step for the full parameterization of the ellipse s.

The mean-shift algorithm can be used to calculate the gradient step on a function that have form resembling a kernel density estimate such as (5) from Section 2.5. The mean-shift step is derived from a lower bounding function of (5). The lower bound follows from the convexity of the kernel function [15,7]. From the Jensen's inequality, typical for variational approaches, we can get a different lower bound:

$$\log f(s) \geq G(s,q_1,...,q_N) = \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \log\left(\frac{\omega_i |V|^{\gamma/2} \mathcal{N}(x_i;\theta,V)}{q_i}\right)^{q_i} \tag{8}$$

where $\sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} q_i = 1$ and $q_i \geq 0$. The superscript $\{k\}$ denotes the iteration index. It is easy to show that for a given s the equality sign in (8) is achieved for:

$$q_i = \frac{\omega_i \mathcal{N}(x_i; \theta^{\{k\}}, V^{\{k\}})}{\sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \omega_i \mathcal{N}(\vec{x}_i; \theta^{\{k\}}, V^{\{k\}})}. \tag{9}$$

Given the q_i-s we maximize the part of G that depends on the parameters:

$$g(s) = \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} q_i \log(|V|^{\gamma/2} \mathcal{N}(x_i;\theta,V)). \tag{10}$$

For the Gaussian kernel and $\frac{\partial}{\partial \theta} g(s) = 0$ we get:

$$\theta^{\{k+1\}} = \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} q_i x_i = \frac{\sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \vec{x}_i \omega_i \mathcal{N}(x_i; \theta^{\{k\}}, V^{\{k\}})}{\sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} \omega_i \mathcal{N}(x_i; \theta^{\{k\}}, V^{\{k\}}).} \tag{11}$$

Note that this update equation for the position estimate is equivalent to the mean-shift update for the Gaussian kernels. An advantage is that we can now

derive simple equations for updating V. For Gaussian kernel from $\frac{\partial}{\partial V} g(\theta, V) = 0$ we get:

$$\vec{V}^{\{k+1\}} = \beta \sum_{x_i \in \mathcal{S}^e(s^{\{k\}})} q_i (x_i - \theta^{\{k\}})(x_i - \theta^{\{k\}})^T \tag{12}$$

where $\beta = 1/(1-\gamma)$.

For the sake of clarity we present here the whole algorithm for local search for the most likely configuration (maximum of (3) from Section 2.3):

Input: starting shape $s^{\{k\}}$ $(k = 0)$, the new image I.

(1) For $s^{\{k\}}$ calculate the r_m-s (2) and the weights (6).
(2) Calculate q_i-s using (9) and the new estimates $\theta^{\{k+1\}}$,$V^{\{k+1\}}$ using (11-12).
(3) Line search if needed.
(4) If no new pixels are included using the new elliptical region defined by the new estimates $\theta^{\{k+1\}}$ and $V^{\{k+1\}}$ stop, otherwise set $k \leftarrow k+1$ and go to 1.

Because of the additional local approximation (4) we need an additional a line search step to get a proper gradient descent procedure. For many distance measures the second condition from (8) is not satisfied since the weights ω_i from (6) are not always nonnegative. In such cases the update steps are still in the gradient direction but the line search is advisable. For the Bhattacharayya coefficient measure the weights are always positive and in practice it turns out that line search is not necessary since (4) appears to be often a good local approximation [7,24].

3 Tracking - inference

To track an object, we wish to estimate the *the filtering density* $p(s_t|I_{1:t})$ for each $t = 1, 2 \ldots$ given the sequence of measurements $I_{1:t}$ [1]. The basic observation in Bayesian filtering is that $p(s_t|I_{1:t})$ can be calculated recursively if the dynamic model is described by a first order Markov process $p(s_t|s_{t-1}) = p(s_t|s_{1:t-1})$ and the measurements are independent from each other given the latent dynamic process, i.e., $p(I_t|s_t) = p(I_t|s_{1:T})$. Recursive updates are performed using:

[1] For additional image modalities as in Section 2.4 we have $\{I, \tilde{I}\}$ instead of I

$$\text{the prediction stage: } p(s_t|I_{1:t-1}) = \int p(s_t|s_{t-1})p(s_{t-1}|I_{1:t-1})ds_{t-1} \quad (13)$$

$$\text{the update stage: } p(s_t|I_{1:t}) = \frac{1}{c}p(I_t|s_t)p(s_t|I_{1:t-1}) \quad (14)$$

where $c = \int p(I_t|s_t)p(s_t|z_{1:t-1})ds_t$ and $p(s_{t-1}|I_{1:t-1})$ is the previous estimate.

3.1 Approximate Bayesian filtering

We use a simple random walk model $p(s_t|s_{t-1}) = \mathcal{N}(s_t; s_{t-1}, Q)$. We further assume transition noise covariance Q to be diagonal, with values estimated from data. Clearly, more elaborate dynamical models can be envisaged, e.g., see [13,14]. Usually in machine vision applications it is the complex form of the observation function (3) that renders the update step (14) analytically intractable. We will investigate two approximation strategies:

Approximating the observation model: The $p(I_t|s_t)$ when viewed as a function of s_t given I_t is often multimodal. The key to a good approximation of $p(I_t|s_t)$ is in capturing the modes. We initialize the local search from Section 2.5 from K different start positions in order to find the modes. The starting points are generated as in [2] from $\alpha p(s_t|I_{1:t-1}) + (1-\alpha)u(s_t)$. Here α is a mixing parameter that allows samples both from the prediction $p(s_t|I_{1:t-1})$ and from some wide distribution $u(s_t)$ (for example uniform over the whole image). By tuning α, we adjust the amount of "surprise", as well as discount for the fact that we have only an approximate $p(s_t|I_{1:t-1})$ (we use $\alpha = 0.9$). The result of the local search are $K(I_t) \leq K$ modes $m(I_t)^j$ (as in [2] we detect two searches ending in the same mode and determine the covariance matrices $R(I_t)^j$ by local fitting):

$$p(I_t|x_t) \approx \sum_{j=1}^{K(I_t)} \rho(I_t)^j \mathcal{N}(x_t; m(I_t)^j, R(I_t)^j) \quad (15)$$

where the superscript $j = 1 \ldots K(I_t)$ denotes the components of the mixture. Here, $\rho(I_t)^j$ denotes the weight of the j'th mixture component. The Kalman filter (KF) is obtained if we use one search (and $\alpha = 1$) as in [7]. On the other hand, especially in case of occlusion, $p(I_t|s_t)$ has a number of modes. The correct trajectory can only be disambiguated after observing the future data. Discarding a mode may cause the tracker to miss the track. Therefore, keeping all the $K(I_t)$ modes and performing inference using a mixture Kalman filter (MKF) [4] is more effective.

Sampling based methods - Sequential monte Carlo (SMC): In SMC, the filtering distribution is represented by a set of particles (samples) $s_{t-1}^{(i)}$ and their associated weights $\{\tilde{w}_{t-1}^{(i)}, i = 1 \ldots N\}$. The idea of SMC is to evolve this

frame 0 | Mean-shift tracking (frames 50,150) | Extended mean-shift tracking (frames 50,150)

Fig. 1. Illustrating the performance of the extended mean-shift [24] compared to the mean-shift with simple scale adaptation [7]. The estimated position and shape of the tracked object is represented by the dashed ellipse.

representation into a new set of weights and particles $\{\tilde{w}_t^{(i)}, s_t^{(i)}, i = 1 \dots N\}$ via (13) and (14) when the observation z_t becomes available at time t. The common practice is to use importance sampling to resolve the following basic issues: (1) How to generate a new set of samples, and (2) How to compute the new weights. The samples are generated using a proposal density q. Many well known particle filtering algorithms are equivalent to choosing a particular form of q; e.g., the bootstrap particle filter (BPF) of [13] is obtained when we sample from the transition model, i.e., $q(s_t) = p(s_t|s_{t-1}^{(i)})$.In this paper, we use the approximation (15) as the proposal sampling distribution - which we name as particle filter with proposal (PFP) and refer the reader to [8] for the algorithmic details.

4 Experiments

4.1 Extended mean shift and the mean-shift local search

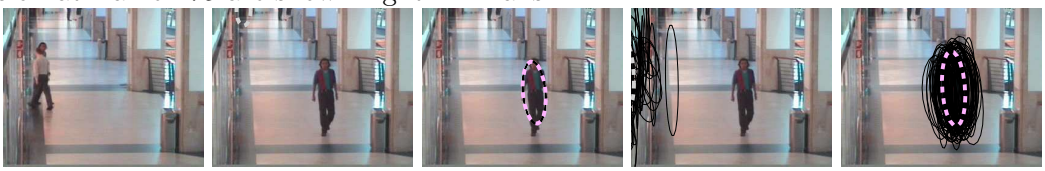
A sequence presented in Figure 1 is used to illustrate our local search procedure. The glass is tracked successfully while its position, shape and orientation are changing rapidly. We used the "best" position from the previous frame to start the search for the new image. In contrast to the standard mean-shift [7] which uses simple scale adaptation, our extended algorithm [24] performs a full 5-DOF gradient search to adapt the shape of the object much better (Figure 1) with only a slight increase in computational complexity. Per frame, regular mean-shift uses on average 4 iterations (or 12 with simple scale adaptation [7]) while our extended algorithm uses on average 6 iterations per frame. Empirical comparison of running time of the three approaches also validate this observation where the extended procedure [24] takes on average only 2 times more then a single mean-shift search but 50% less than the procedure with the simple scale adaptation [7].

occlusion KF MKF BPF PFP

a) In the sequence "FightRunAway1" the tracked person is occluded during a fighting scene, frame 320 shown left (zoom in). The tracking results after the occlusion at frame 410 are shown right. KF fails.

occlusion KF MKF BPF PFP

b) In the sequence "ShopAssistant1front" the person gets occluded by walking behind the pillar, frame 80 shown left (zoom in). The tracking results after the occlusion at frame 175 are shown right. KF fails.



occlusion KF MKF BPF PFP

c) In the sequence "EnterExitCrossingPaths2cor" the tracked person get occluded by another person, frame 210 shown left (zoom in). The tracking results after the occlusion at frame 315 are shown right. KF and BPF fail.

Fig. 2. Illustrating how the various tracking schemes handle occlusion. The maximum of the estimated density, represented by the dashed ellipse, is used as the estimated position and shape. For the PFP and BPF the particles are shown as black ellipses (100 particles used). For MKF the black ellipses are different modes.

4.2 Comparing different Bayesian tracking schemes

We used CAVIAR dataset which contains various surveillance videos with ground truth bounding boxes of the walking people, see Figure 2 and http://homepages.inf.ed.ac.uk/rbf/CAVIAR. From this dataset we selected 12 sequences where persons get largely occluded. For each sequence we tracked a single person that gets occluded. The frame at which the person appears and the corresponding ground truth bounding box were used to initialize the object model (the histogram-based appearance and the elliptical shape). The person is then tracked until it leaves the field of view. We use the following relative overlap measure to evaluate the algorithms. Let R_{gt} be the image region defined by the ground truth bounding box. Let R_e be the estimated elliptical region (the ellipse corresponding to the maximum of the estimated filtering density $p(s_t|I_{1:t-1})$). The relative overlap is defined by: $overlap = \frac{R_e \cap R_{gt}}{R_e \cup R_{gt}}$ where $R_e \cap R_{gt}$ is the intersection and $R_e \cup R_{gt}$ is the union of the two image regions. The relative overlap can have values between 0 and 1 and we report the average value of the overlap for the sequences. The average overlap might not

frame 1550 and the foreground mask

frame 1580 and BPF particle distribution without and with color

Fig. 3. Illustrating feature combination. In the sequence "WakByShop1cor" the tracked person on the left is occluded, frame 1550 (zoom in) shown left. After the occlusion, frame 1580 shown the particles (black ellipses) of the foreground blob tracker can't distinguish the blobs. BPF which combines foreground and color has the particles still concentrated on the person on the left. The dashed ellipse represents the mean of the distribution.

Histogram type	Rel. overlap with the ground truth	KF	MKF (5 modes)	BPF (100 particles)	PFP (100 particles)
color only	average overlap	0.29	0.33	0.31	0.33
	overlap>0.2	60.1%	63.1%	60.7%	62.3%
	overlap>0.4	40.0%	47.7%	43.5%	48.0%
fg./bg. only (blob tracking)	average	0.28	0.30	0.30	0.31
	overlap>0.2	58.3%	60.2%	60.1%	60.6%
	overlap>0.4	38.5%	46.7%	44.5%	45.5%
color+ fg./bg.	average overlap	0.33	0.47	0.40	0.48
	overlap>0.2	63.5%	89.8%	81.1%	93.2%
	overlap>0.4	45.9%	61.6%	56.1%	62.8%
color+ fg./bg. (simple sequences)	average overlap	0.47	0.50	0.51	0.51
	overlap>0.2	90.6%	92.7%	94.2%	94.6%
	overlap>0.4	59.3%	65.2%	68.1%	69.0%

Table 1
Evaluation results on a data set of 12 sequences, 2100 frames in total. Relative overlap with the ground truth bounding box is presented. The color histogram, foreground/background segmentation and their combination were used. The last row shows evaluation on 12 simpler sequences where the tracked person does not get occluded, 2300 frames in total.

be the "best"performance measure for tracking. Therefore, we also report the percentage of total number of frames where $overlap > 0.4$, and $overlap > 0.2$. We have chosen the relative overlap 0.4 as the threshold since the estimated ellipse and the ground truth bounding box then look visually quite close.

In Table 1 we report the results of four different tracking experiments for all sequences. In the first experiment, we used just color features within the

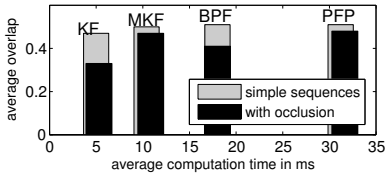


Fig. 4. Performance versus computation time for the scenes with and without occlusions.

tracking schemes from Section 3. The results were poor since the algorithms get sometimes confused by similarly colored objects. We used $8 \times 8 \times 8$ histogram in RGB space and no improvements were noticed for using more histogram bins. In the second experiment we used the fact that the camera was static and a background subtraction segmentation algorithm can be applied, e.g. [20]. A blob detection on the foreground mask can then be used for tracking. In Section 2.4 we show how the foreground mask features can be included into a histogram based scheme. The tracking schemes from Section 3 resemble then simple blob tracking where goal is to find the ellipse with certain number of foreground pixels. However, the blob tracking can not differentiate between different blobs especially during and after occlusions. This is illustrated in Figure 3. In the third experiment we combined the two modalities, see Section 2.4. This leads to large improvements. Some of the sequences and the tracking results are illustrated in Figures 2 and 3. Finally, in the last experiment we illustrate that the performance depends on the type of sequence. Therefore we selected another set of 12 simpler sequences from the CAVIAR data where there were no significant occlusions of the tracked person.

The performances of the different schemes on the CAVIAR sequences and the average computation times are summarized in Figure 4. The computation times in Figure 4 are measured for our implementation on a 2GHz computer but they should give a realistic relations between the different techniques. There are more mean-shift iterations needed to find the modes for the difficult sequences with occlusions. Therefore the computation time for KF, MKF and PFP is slightly higher for these sequences. We used 100 particles for the sampling approaches BPF and PFP, and 5 mean-shift searches for the MKF and PFP. With this settings all the algorithms can perform in real time. Another reason for using 5 mean-shift searches in MKF and PFP was that we did not observe significant performance improvement on the dataset when using more than 5 searches.

5 Conclusions

We presented an observation model where the shape of the tracked object is approximated by an ellipse in general position and its appearance by histogram based features. Advantages of the model are its simplicity and general applicability. An additional advantage is the provided efficient local search procedure. Finally, we demonstrated a simple combination of different modalities.

The paper also proposes to use the local search as a mode finding algorithm which is then integrated into a range of approximate Bayesian filtering schemes: KF, MKF, and PFP (see Section 3.1). The developed schemes are compared with each other. Being unimodal the KF performed the worst in all tests. The performance of the MKF, BPF and PFP was on average quite similar for the simpler sequences from the realistic surveillance dataset. However, for the more difficult sequences with occlusions the performance of the KF was very poor and the performance of BPF also degraded. The local search is used within the MKF and PFP to find the modes of the estimated distribution more effectively which is especially important after the tracked object was occluded. Theoretically, if the number of particles is increased, the BPF should also be able to find the modes more quickly but this would require more computation time.

We also provide empirical results on computation time of the presented schemes. The computation in the KF and MKF scale linearly with the number of local searches that are performed at each step. The computation costs for the BPF and PFP scale linearly with the number of particles. The PFP includes also the costs of the local searches. The measured times indicate that the PFP with 5 searches and 100 particles requires twice the time of the MKF with also 5 searches. The MKF is a better choice if the computation time is important. On the other hand, the PFP retains the theoretical convergence properties [8] and the tracking results improve slightly.

References

[1] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[2] B.Han, Y.Zhu, D.Comaniciu, and L.Davis. Kernel-based bayesian filtering for object tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[3] P. Chang and J. Krumm. Object recognition with color cooccurrence histograms. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1999.

[4] R. Chen and J. Liu. Mixture kalman filters. *Journal of Royal Stat. Soc. B*, 62:493–508, 2000.

[5] R. Collins and Y. Liu. On-line selection of discriminative tracking features. *In Proc. International Conference of Computer Vision*, 2003.

[6] T.F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: Their training and application. *Computer Vision Image Understanding*, 61(1):38–59, 1995.

[7] D.Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(5):564–575, 2003.

[8] A. Doucet, N. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[9] D. M. Gavrila. The visual analysis of human movement: A survey. *Comp. Vision and Im. Understanding*, 73(1):82–98, 1999.

[10] G.R.Bradski. Computer vision face tracking as a component of a perceptual user interface. *In Proc. IEEE Workshop on Applications of Computer vision*, pages 214–219, 1998.

[11] G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with SSD. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 790–797, 2004.

[12] H.Zhang, W.Huang, Z.Huang, and L.Li. Affine object tracking with kernel-based spatial-color representation. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[13] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *Int. J. of Computer Vision*, 29(1):5–28, 1998.

[14] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. *In Proceedings of the Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 1997.

[15] K.Fukunaga and L.D.Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Information Theory*, 21:32–40, 2002.

[16] K-C. Lee and D. Kriegman. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[17] M.J.Black and A.D.Jepson. Eigentracking: Robust matching and tracking articulated objects using a view based representation. *Int. J. of Computer Vision*, 26(1):63–84, 1998.

[18] K. Nummiaro, E. Koller-Meier, and L.J. van Gool. An adaptive color-based particle filter. *Image Vision Computing*, 21(1):99–110, 2003.

[19] R.Collins. Mean-shift blob tracking through scale space. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[20] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 246–252, 1999.

[21] S.T.Birchfield and S.Rangarajan. Spatiograms versus histograms for region-based tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[22] M. Swain and D. Ballard. Color indexing. *Intl. J. of Computer Vision*, 7(1):11–32, 1991.

[23] C. Yang, R. Duraiswami, and L. Davis. Efficient spatial-feature tracking via the mean-shift and a new similarity measure. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[24] Z. Zivkovic and B. Krose. An EM-like algorithm for color-histogram-based object tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[25] Y. Rui and Y. Chen. Better Proposal Distributions: Object Tracking Using Unscented Particle Filter. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.