Title	A new visual object tracking algorithm using Bayesian Kalman filter
Author(s)	Zhang, S; Chan, SC; Liao, B; Tsui, KM
Citation	The 2014 IEEE International Symposium on Circuits and Systems (ISCAS), Melbourne, Australia, 1-5 June 2014. In IEEE International Symposium on Circuits and Systems Proceedings, 2014, p. 522-525
Issued Date	2014
URL	http://hdl.handle.net/10722/204109
Rights	Creative Commons: Attribution 3.0 Hong Kong License

A New Visual Object Tracking Algorithm Using Bayesian Kalman Filter

Shuai Zhang, S. C. Chan, Bin Liao and K. M. Tsui
Dept. Electrical and Electronic Engineering
The University of Hong Kong
Hong Kong, P. R. China

***Abstract*—This paper proposes a new visual object tracking algorithm using a novel Bayesian Kalman filter (BKF) with simplified Gaussian mixture (BKF-SGM). The new BKF-SGM employs a GM representation of the state and noise densities and a novel direct density simplifying algorithm for avoiding the exponential complexity growth of conventional KFs using GM. Together with an improved mean shift (MS) algorithm, a new BKF-SGM with improved MS (BKF-SGM-IMS) algorithm with more robust tracking performance is also proposed. Experimental results show that our method can successfully handle complex scenarios with good performance and low arithmetic complexity.**

Keywords—Object tracking, Baysian Kalman filter, mean shift.

I. INTRODUCTION

Visual object tracking is an important component in video surveillance, because many high-level video analytic tasks and other applications use object tracking results as a part of their initial inputs. An extensive survey of various state-of-the-art object tracking algorithms can be found in [1]. In general, visual object tracking can be classified into two categories: deterministic and probabilistic tracking. For the former framework, mean shift (MS) [3] is the most popular tracker because of its simplicity and effectiveness. It only keeps a single hypothesis/candidate and utilizes the gradient of the data distribution for seeking the maximum possible candidate. Consequently, it is very computationally efficient. However, conventional MS tracker is prone to losing tracks due to rapid movement of the object. Moreover, its performance degrades considerably if significant occlusion occurs or there are more than one similar objects in the scene.

On the other hand, probabilistic tracking methods model the important information of the tracked object as a probability density function (pdf), which is recursively updated with new observations arrived. Usually, a dynamic state-space approach is employed where the states of the object are typically taken as its location, velocity, acceleration and bounding box, etc. The observations in each image frame are usually obtained by an initial detector or tracker. In this category, particle filtering (PF) algorithm [2] has received great attention because it is capable of dealing with non-Gaussian state densities by approximating them as a set of particles with associated weight. Unlike MS, these methods keep multiple hypotheses and thus are more suitable for tracking in clutter or occlusions. Despite of its effectiveness in tracking, the computational complexity of such methods grows rapidly when the dimension of the states and the number of particles grows. In addition, due to the problems of degeneracy and sampling impoverishment, the PF is rather inefficient in sampling.

In this paper, we approximate the non-Gaussian state and noise densities by Gaussian mixtures (GMs) and propose a new Bayesian Kalman filter (BKF)-based visual object tracking algorithm. The proposed BKF is based on the classical formulation of Ho et al. [5] who showed that the classic Gaussian KF formulation can be extended by means of the Bayesian framework to handle more general pdf. However, for the non-Gaussian and/or non-linear system, the Bayesian recursion cannot be generally computed using closed-form formula due to difficulty in evaluating the multidimensional integral analytically. Therefore, one needs to approximate the state and noise pdfs using various techniques such as particles or GMs [4]. For the former, it gives rise to the PF technique discussed above. For the GMs, the filtering process of the original problem can be aproximately solved by a number of KFs operating in parallel. Unfortunately, its main drawback is that the filter number and the component number of the states will exponentially increase with time. To tackle this problem, a new direct density simplification approach and further the BKF with simplified GM (BKF-SGM) are proposed and applied to the visual object tracking. As the GM is simplified directly without resampling, the proposed BKF-SGM avoids performance degradation due to sampling degeneracy and impoverishment in conventional PF. Furthermore, coupled with an improved MS tracker for extracting measurements from image frame, the original MS is extended under the BKF-SGM framework to a bank of parallel MS trackers, which reduces the possibility of the MS tracker being trapped in local maxima at the background or similar objects. The proposed approach, called the BKF-SGM with improved MS (BKF-SGM-IMS) can also be viewed as a fusion of the deterministic and probabilistic object tracking frameworks, where local deterministic approach is used to provide the measurements for the probabilistic approach based on the BKF-SGM. Experimental results tested on public available datasets PETS2001 [8] & PETS2009 [9] and HD IP surveillance camera captured dataset show that the proposed approach outperforms the conventional PF-based and MS methods with low computational complexity.

The rest of the paper is organized as follows. Section II is devoted to the proposed BKF-SGM-IMS tracking algorithm.

978-1-4799-3432-4/14/$31.00 ©2014 IEEE

The implementation details, experimental results and comparisons with conventional algorithms are given in Section III. Finally, conclusion is drawn in Section IV.

II. BKF-SGM-IMS Visual Object Tracking

A. Baysian Kalman Filter with Gaussian Mixture (BKF-GM)

The probabilistic object tracking problem can be modeled by the discrete-time linear state-space model as follows:

$$\boldsymbol{x}_k = \boldsymbol{A}_k \boldsymbol{x}_{k-1} + \boldsymbol{w}_k , \tag{1}$$

$$\boldsymbol{z}_k = \boldsymbol{C}_k \boldsymbol{x}_k + \boldsymbol{v}_k , \tag{2}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ denote respectively the state and observation (measurement) vectors at time k. $\boldsymbol{A}_k$ denotes the state transition matrix and $\boldsymbol{C}_k$ constitutes the observation model which relates the measurement with the state. $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ denote respectively the process and observation noise vectors, and are assumed to be mutually independent.

In this paper, the state vector and transition matrix are defined as $\boldsymbol{x}_k = [x_k, y_k, Hx_k, Hy_k, \dot{x}_k, \dot{y}_k, \dot{H}x_k, \dot{H}y_k]^T$ and $\boldsymbol{A}_k = [[\boldsymbol{I}_{4\times4}, \boldsymbol{0}]^T \quad [\boldsymbol{I}_{4\times4}, \boldsymbol{I}_{4\times4}]^T]$, where (x_k, y_k) and $(\dot{x}_k, \dot{y}_k)$ are respectively the location of the object center and its velocities, (Hx_k, Hy_k) are the half axes of the tracking eclipse (bounding box) and $(\dot{H}x_k, \dot{H}y_k)$ are the corresponding scale changes. The observation model which relates the measurement $\boldsymbol{z}_k = [x_k, y_k, Hx_k, Hy_k]^T$ with the state is defined as $\boldsymbol{C}_k = [\boldsymbol{I}_{4\times4} \quad \boldsymbol{0}_{4\times4}]$. More details about the computation of $\boldsymbol{z}_k$ using the improved MS will be discussed in Section II(C).

In the BKF-GM, the pdfs of the process and observation noises are characterized by the GMs as: $p(\boldsymbol{w}_k) = \sum_{l=1}^{\eta_k} \beta_{kl} N(\boldsymbol{w}_k; \boldsymbol{u}_l, \boldsymbol{Q}_{kl})$ and $p(\boldsymbol{v}_k) = \sum_{l=1}^{\overline{\eta}_k} \overline{\beta}_{kl} N(\boldsymbol{v}_k; \overline{\boldsymbol{u}}_l, \overline{\boldsymbol{Q}}_{kl})$, where η_k ($\overline{\eta}_k$) is the number of components, β_{kl} ($\overline{\beta}_{kl}$) is the probability of the l-th component, $\sum_{i=1}^{\eta_k} \beta_{kl} = 1$ ($\sum_{i=1}^{\overline{\eta}_k} \overline{\beta}_{kl} = 1$), and $N(\boldsymbol{w}_k; \boldsymbol{u}_l, \boldsymbol{Q}_{kl})$ represents a Gaussian distribution with mean $\boldsymbol{u}_l$ and covariance $\boldsymbol{Q}_{kl}$. In the rest of the paper, the observation noise is assumed to be zero-mean Gaussian with covariance matrix $\boldsymbol{R}_k$, i.e., $p(\boldsymbol{v}_k) = N(\boldsymbol{v}_k; \boldsymbol{0}, \boldsymbol{R}_k)$, for simplicity. Now suppose further that $\boldsymbol{Z}_{k-1}$ denotes the observations collected up to time instant k-1 and the a prior pdf $p(\boldsymbol{x}_k \mid \boldsymbol{Z}_{k-1})$ is modeled by the following GM with ξ'_k components:

$$p(\boldsymbol{x}_k \mid \boldsymbol{Z}_{k-1}) = \sum_{i=1}^{\xi'_k} \alpha'_{ki} N(\boldsymbol{x}_k; \boldsymbol{u}'_{ki}, \boldsymbol{P}'_{ki}) , \tag{3}$$

where α'_{ki} is the weight of the corresponding component with $\sum_{i=1}^{\xi'_k} \alpha'_{ki} = 1$. Then, given the k-th observation, the a posteriori density is given by

$$\begin{aligned} p(\boldsymbol{x}_k \mid \boldsymbol{Z}_k) &= c_k p(\boldsymbol{z}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{Z}_{k-1}) \\ &= c_k N(\boldsymbol{z}_k - \boldsymbol{C}_k \boldsymbol{x}_k, \boldsymbol{R}_k)[\textstyle\sum_{i=1}^{\xi'_k} \alpha'_{ki} N(\boldsymbol{x}_k; \boldsymbol{u}'_{ki}, \boldsymbol{P}'_{ki})] \\ &= \textstyle\sum_{i=1}^{\xi_k} \alpha_{ki} N(\boldsymbol{x}_k; \boldsymbol{u}_{ki}, \boldsymbol{P}_{ki}), \end{aligned} \tag{4}$$

where $c_k^{-1} = \int p(\boldsymbol{z}_k \mid \boldsymbol{x}_k) p(\boldsymbol{x}_k \mid \boldsymbol{Z}_{k-1}) d\boldsymbol{x}_k$ is a normalization constant. The second equality in (4) consists of the products of two Gaussian distributions and each of these products can be simplified to a Gaussian distribution. As a result, one gets $\xi_k = \xi'_k$, and the KF for Gaussian processes [4] is directly applicable to obtain α_{ki}, $\boldsymbol{u}_{ki}$ and $\boldsymbol{P}_{ki}$. Please refer to Table I for details. Similar to the derivation in (4), the a prior density is:

$$\begin{aligned} p(\boldsymbol{x}_{k+1} \mid \boldsymbol{Z}_k) &= \int p(\boldsymbol{x}_k \mid \boldsymbol{Z}_k) p(\boldsymbol{x}_{k+1} \mid \boldsymbol{x}_k) d\boldsymbol{x}_k \\ &= \textstyle\sum_{j=1}^{\overline{\xi}'_{k+1}} \overline{\alpha}'_{(k+1)j} N(\boldsymbol{x}_{k+1}; \overline{\boldsymbol{u}}'_{(k+1)j}, \overline{\boldsymbol{P}}'_{(k+1)j}), \end{aligned} \tag{5}$$

where $\overline{\xi}'_{k+1} = \xi_k \eta_k$. The update of $\overline{\alpha}'_{(k+1)j}$, $\overline{\boldsymbol{u}}'_{(k+1)j}$ and $\overline{\boldsymbol{P}}'_{(k+1)j}$ are summarized in TABLE I. It can be seen that the number of mixture components $\overline{\xi}'_{k+1}$ and hence the complexity grow exponentially after each recursion. To keep a constant complexity, the GM model in (5) should be approximated as the form of (3).

B. BKF with Simplified GM (BKF-SGM)

Consider the GM model in (5) with $\overline{\xi}'$ components:

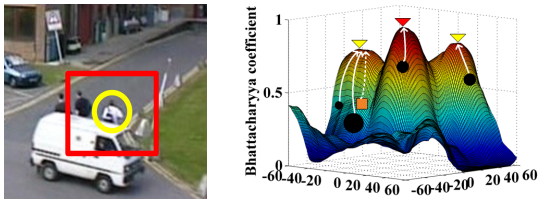
$$F(\boldsymbol{x}) = \textstyle\sum_{j=1}^{\overline{\xi}'} \overline{\alpha}'_j f_j(\boldsymbol{x}) = \sum_{j=1}^{\overline{\xi}'} \overline{\alpha}'_j N(\boldsymbol{x}; \overline{\boldsymbol{u}}'_j, \overline{\boldsymbol{P}}'_j) . \tag{6}$$

where $f_j(\boldsymbol{x}) = N(\boldsymbol{x}; \overline{\boldsymbol{u}}'_j, \overline{\boldsymbol{P}}'_j)$ is the j-th component and $\sum_{j=1}^{\overline{\xi}'} \overline{\alpha}'_j = 1$. For notational convenient, the additional subscript k+1 in (5) is dropped. In the BKF-SGM, our goal is to approximate $F(\boldsymbol{x})$ with a simplified mixture model with fewer components $G(\boldsymbol{x}) = \sum_{i=1}^{\xi'} \alpha'_i g_i(\boldsymbol{x})$, where $g_i(\boldsymbol{x}) = N(\boldsymbol{x}; \boldsymbol{u}'_i, \boldsymbol{P}'_i)$ and $\sum_{i=1}^{\xi'} \alpha'_i = 1$ with $\xi' < \overline{\xi}'$. The error of approximating $F(\boldsymbol{x})$ with $G(\boldsymbol{x})$ is given by $D(F(\boldsymbol{x}), G(\boldsymbol{x})) = (\int (F(\boldsymbol{x}) - G(\boldsymbol{x}))^2 d\boldsymbol{x})^{0.5}$. Conventionally, the simplification is done by first resampling and then clustering using the K-means or EM algorithm. However, the complexity depends exponentially on the dimension of the state and hence it will soon become infeasible. In this paper, we employ the two-step algorithm developed in [6] for model order reduction while avoiding the additional resampling process. At the m-th iteration of the two-step algorithm, the component mixture is partitioned into ξ' groups $\{S_1^{(m)}, S_2^{(m)}, ..., S_{\xi'}^{(m)}\}$.

Step 1 (Mean update): The representative component $C_i^{(m)}(\boldsymbol{x})$ for each $S_i^{(m)}$ that minimizes the local quantization error is $C_i^{(m)}(\boldsymbol{x}) = \arg\min_{C(\boldsymbol{x})} \sum_{j \in S_i^{(m)}} \alpha_j \int (C(\boldsymbol{x}) - f_j(\boldsymbol{x}))^2 d\boldsymbol{x}$. Interested readers are referred to [6] and references therein for solving this problem using coordinate descent.

Step 2 (Clustering): Given $C^{(m)} = \{C_i^{(m)}(\boldsymbol{x})\}_{i=1}^{\xi'}$, one re-assigns $f_j(\boldsymbol{x})$ to the nearest $C_i^{(m)}(\boldsymbol{x})$ based on the distortion measure $D_{i,j} = D(C_i^{(m)}(\boldsymbol{x}), f_j(\boldsymbol{x}))$, and then update $S_i^{(m)}$. The above process is repeated until either 1) the change in total distortion or $C^{(m)} = \{C_i^{(m)}(\boldsymbol{x})\}_{i=1}^{\xi'}$ is less than a certain threshold, or 2) a maximum number of iteration is reached.

The major advantage of the proposed method over other conventional resampling methods is the use of an efficient mixture simplification method with lower complexity. For example, denote the number of component and dimension of state respectively as n and d. Then, the complexity of the greedy EM algorithm [7] is $O(\eta_k \overline{\eta}_k n d^3 + \kappa n^2 (d^2 + 2d + 1))$, where κ is related to the number of candidates and data size in the greedy EM algorithm. On the other hand, the complexity of the two-step algorithm mentioned above is $O(T(L+m)nd^3)$, where T and L denote the numbers of iterations and are usually small as suggested in [6]. Since κ is typically very large, say from 1000 to 5000, for the greedy EM algorithm, the method in



(a) (b)

Fig.1 Illustration of the converge behavior of the BKF-SGM-IMS. (a) shows a tracked object (yellow) which centers at a small area (red); (b) shows the similarity surface (values of the Bhattacharyya coefficient) corresponding to the red rectangle marked in (a), where red triangle point to the global maxima (true object center) and yellow triangles are two local maximum points; The orange rectangle is the estimated object center of the GM and 4 black circles are object centers of the GM components.

[7] has a higher complexity.

C. BKF-SGM with Improved MS (BKF-SGM-IMS)

As mentioned before, the observation vector z_k defined in (2) can be obtained by the MS tracker [3]. It models the appearance of the tracked object by a weighted color histogram and the object center (x_k, y_k) is calculated by iteratively maximizing the similarity between the tracked object and its candidate. Histogram similarity is defined in terms of the Bhattacharya coefficient and distance [3]. At each iteration, the mean shift vector is computed such that the histogram similarity is increased. This process is repeated until convergence is achieved. (Hx_k, Hy_k) is calculated by scale adaptation, which modifies (Hx_{k-1}, Hy_{k-1}) with a certain fraction ($\pm 10\%$) and lets the MS converge again. As we can see from (4), the posterior density is approximated by GM. Therefore, it makes sense to perform MS separately for each GM component so as to obtain the z_k. A major difficulty in using such approach is that there are many measurements $\tilde{z}_{k,i}$, one from each Gaussian component with $i = 1, \cdots, \xi'_k$. To simplify the computation, we pick the most likely measurement from each MS tracker and use it to update the state density. More precisely, z_k is obtained from minimizing a cost function for each MS tracker associated with each Gaussian component. The cost function $E_{i,k}$ is constructed from the component weight and the Bhattacharyya distance: $E_{i,k} = \omega(1-\alpha'_{i,k}) + (1-\omega)d_{i,k}$, where $\alpha'_{i,k}$ and $d_{i,k} \in [0,1]$ are respectively the component weight and normalized Bhattacharyya distance for the *i*-th GM component at time *k*, and $\omega = 1/\xi'_k$ is a weighting constant to ensure that $E_{i,k}$ has a magnitude less than or equal one. The final measurement z_k is chosen as $z_k = \tilde{z}_{k,i=\zeta_k}$, where $\zeta_k = \arg\min_i E_{i,k}$.

The main advantage of this approach is that it reduces the possibility of the MS tracker in getting trapped in a local maximum point on the background or other similar objects. This leads to more accurate measurement and hence tracking results. For example, consider a tracked object inside a red rectangle of Fig. 1(a) and the histogram similarity surface is shown in Fig. 1(b). The object center estimated by a GM is marked as orange rectangle and object centers of four corssponding Gaussian components are marked as black circles in Fig. 1(b). In addition, the weight $\alpha'_{i,k}$ of the *i*-th GM component is represented by the size of the black point and white arrows indicate the converge behavior of the conventional MS. It can be seen from Fig. 1(b) that the similarity surface is not unimodal. Hence the orange point will converge to a local maximum point in the conventional MS, which is indicated by the white dotted arrow. However, in the improved MS, it is possible to find the global maxima because MS is performed separately for each component. This suggests the usefulness of the proposed improved MS tracker for BKF-SGM. Finally, Table I summarizes the proposed BKF-SGM-IMS in which the parameters of the simplified model are denoted by ξ'_{k+1}, $\alpha'_{(k+1)i}$, $\boldsymbol{u}'_{(k+1)i}$ and $\boldsymbol{P}'_{(k+1)i}$.

TABLE I. BKF-SGM-IMS Visual Object Tracking Algorithm

For $k = 1,2,...$

1) BKF-GM:

For $i = 1 : \xi'_k$

$$\boldsymbol{u}_{ki} = \boldsymbol{u}'_{ki} + \boldsymbol{K}_{ki}(z_k - \boldsymbol{C}_k \boldsymbol{u}'_{ki})$$

$$\boldsymbol{K}_{ki} = \boldsymbol{P}'_{ki}\boldsymbol{C}_k^T (\boldsymbol{C}_k \boldsymbol{P}'_{ki} \boldsymbol{C}_k^T + \boldsymbol{R}_k)^{-1}$$

$$\alpha_{ki} = \frac{\alpha'_{ki} N(z_k - \boldsymbol{C}_k \boldsymbol{u}'_{ki}, \boldsymbol{C}_k \boldsymbol{P}'_{ki} \boldsymbol{C}_k^T + \boldsymbol{R}_k)}{\sum_{j=1}^{\xi'_k} \alpha'_{kj} N(z_k - \boldsymbol{C}_k \boldsymbol{u}'_{kj}, \boldsymbol{C}_k \boldsymbol{P}'_{kj} \boldsymbol{C}_k^T + \boldsymbol{R}_k)}$$

End

$\bar{\xi}_{k+1} = \xi_k \eta_k = \xi' {}_k \eta_k$

For $j = 1 : \bar{\xi}_{k+1}$

$\bar{\alpha}'_{(k+1)j} = \alpha_{ki}\beta_{kl}$, $\bar{\boldsymbol{u}}_{(k+1)j} = \boldsymbol{A}_{k+1}\boldsymbol{u}_{ki} + \boldsymbol{w}_{kl}$,

$\bar{\boldsymbol{P}}_{(k+1)j} = \boldsymbol{A}_{k+1}\boldsymbol{P}_{ki}\boldsymbol{A}_{k+1}^T + \boldsymbol{Q}_{kl}$

End

2) BKF-SGM: use two-steps algorithm to obtain:

$\{a'_{(k+1)i}\}_{i=1}^{\xi_{k+1}}$, $\{\boldsymbol{u}'_{(k+1)i}\}_{i=1}^{\xi_{k+1}}$, $\{\boldsymbol{P}'_{(k+1)i}\}_{i=1}^{\xi_{k+1}}$

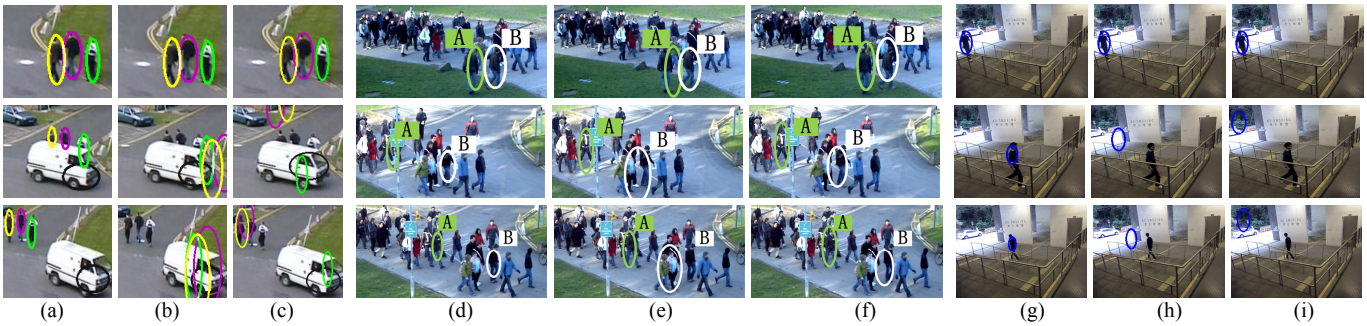
3) Improved MS: use improved MS tracker to obtain z_{k+1}

End

III. Experimental Results

To evaluate the performance of our BKF-SGM-IMS algorithm, two tests have been carried out using public datasets PETS2001 and PETS2009 for quantitative and visual evaluation. Furthermore, our algorithm is tested on a HD and low frame rate (15 fps) surveillance video captured by an IP camera. Since the BKF-SGM-IMS can be viewed as a fusion of deterministic and probabilistic frameworks, we will compare it with the MS [3] and PF-based algorithm [2], which are respectively two classical methods in deterministic and probabilistic tracking. For fair comparison, all methods use a weighted color histogram which is defined in [3] to represent the appearance of the tracked object. In addition, 8 bins for each dimension are used within the RGB space to represent the color histogram. The state noise $\boldsymbol{w}_k$ defined in Section II(A) is modeled as a GM with two components: $p(\boldsymbol{w}_k) = \beta N(\boldsymbol{w}_k; \boldsymbol{u}_1, \boldsymbol{Q}_{k1}) + (1-\beta)N(\boldsymbol{w}_k; \boldsymbol{u}_2, \boldsymbol{Q}_{k2})$, where $\beta = 0.1$, $\boldsymbol{u}_1 = \boldsymbol{u}_2 = \boldsymbol{0}$, $\boldsymbol{Q}_{k1} = 10\boldsymbol{I}$, and $\boldsymbol{Q}_{k2} = \boldsymbol{I}$. The initial state $p(\boldsymbol{x}_0 \mid \boldsymbol{Z}_{-1}) = p(\boldsymbol{x}_0)$ is a GM with 6 components having the same weight. Covariance of each component is set to $\boldsymbol{P}'_{0i} = \boldsymbol{I}$. The total execution time of our BKF-SGM-IMS is nearly equal to the PF-based method with particle number *N*=60 based on C++ implementation. As the number of particles affects the accuracy of the approximation of the pdf and hence the final tracking performance, we use a much larger particle number of *N*=1000 in the comparison.

Figs. 2(a)–2(c) show three moving objects we have tracked



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig.2 Tracking results of PETS2001 dataset at frames 783, 872 and 940, PETS2009 dataset at frames 178, 216 and 232 and low frame rate IP camera captured video sequence at frames 35, 60 and 90, which are obtained using BKF-SGM-IMS (a, d, g), MS (b, e, h) and PF-based method with N=1000 (c, f, i).

in PETS2001 dataset. It can be seen that the color distributions of the yellow and purple eclipses are similar which is difficult for MS to track. When tracking targets experience significant occlusion, only our method can successfully track all three objects afterwards. The PF-based method loses one object and MS loses the two objects. The trajectory ground truth of PETS2001 dataset is used to calculate the velocity RMSE and center RMSE of tracked objects. The BKF-SGM-IMS outperform PF-based methods in velocity tracking. In Table II, we summarize the mean center RMSE and velocity RMSE values of various objects, which are evaluated over 10 trails. The label of "Lost Track" is marked with "Y" if the tracker loses track during any trial. "NaN" indicates the tracker loses tracks in all trials, and hence, we cannot obtain any reasonable value. The velocity RMSE results obtained using the BKF-SGM-IMS are smaller than those of PF-based method for all objects, while offering comparable center RMSE results.

We further tested our method in a more cluster environment, where the selected targets in PTES2009 dataset suffer from lighting changes, direction changes and nearly total occlusion. Figs. 2(d)–2(f) shows the visual tracking results of PETS2009 dataset. It can be seen that our method successfully tracks all objects, whereas MS tracker lose one object at the end of this experiment. The PF-based method does not lose track but the center of tracking ellipse deviates considerably from the real object center. The mean center RMSE and velocity RMSE values of tracked objects, which are evaluated over 10 trails, are shown in TABLE II. It can be seen that the proposed method is significantly better in terms of velocity RMSE. The center RMSE of the proposed method is smaller than the MS tracker.

We now evaluate the performance of the various algorithms in an open environment obtained from a HD resolution but low frame rate IP camera. Due to the low frame rate, the object may move rapidly between two consecutive frames. From frame 6 to frame 37, the object experienced a sudden acceleration. Figs. 2(g)–2(i) shows the comparison results. It can be noticed that our method can successfully track the moving object even though it is slightly affected by the sudden velocity change and direction change. The MS tracker loses track when the object start accelerating. The PF-based method loses track in the middle of the acceleration period.

TABLE II. Comparision of Velocity RMSE, Center RMSE (pixels) and Lost Track

Object	Method	Velocity RMSE	Center RMSE	Lost Track
PETS2001 camera1 dataset				
Yellow	BKF-SGM-IMS	0.1217	14.2178	N
	PF1000	1.7909	12.2166	Y
	MS	NaN	NaN	Y
Purple	BKF-SGM-IMS	0.1119	13.6707	N
	PF1000	1.6616	14.0365	Y
	MS	Na	Na	N
Green	BKF-SGM-IMS	0.2943	18.6802	N
	PF1000	NaN	NaN	Y
	MS	NaN	NaN	Y
PETS2009_S2_L3 dataset				
A	BKF-SGM-IMS	0.2529	1.9327	N
	PF1000	1.7833	0.9874	N
	MS	2.3793	2.2457	N
B	BKF-SGM-IMS	0.3872	1.6786	N
	PF1000	1.8419	1.3588	N
	MS	NaN	NaN	Y

IV. CONCLUSION

A novel BKF-based method for visual objects tracking is presented. It employs GM representations of the state and noise densities and a novel density simplifying algorithm to avoid the exponential increase of the number of GM component. Coupled with an improved MS, the BKF-SGM-IMS visual object tracking algorithm is proposed.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 13-57, 2006.

[2] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. 7th European Conf. Compt. Vision*, 2002, pp. 661-675.

[3] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 564-577, May 2003.

[4] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximation," *IEEE Trans. Automatic Control*, vol. AC-17, Aug. 1972.

[5] Y. C. Ho and R. C. K. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Trans. Automat. Contr.*, vol. AC-9, pp. 333-339, 1964.

[6] K. Zhang and J. T. Kwok, "Simplifying mixture models through function approximation," *IEEE Trans. Neural Networks*, vol. 21, pp. 644-656, Apr. 2010.

[7] I. Bilik and J. Tabrikian, "MMSE-based filtering in presence of non-Gaussian system and measurement noise," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, pp. 1153-1170, Jul. 2010.

[8] PETS2001 [Online]. Available: ftp://ftp.pets.rdg.ac.uk/pub/PETS2001.

[9] PETS2009 [Online]. Available: http://www.cvg.rdg.ac.uk/PETS2009.