Visual Tracking by Continuous Density Propagation in Sequential Bayesian Filtering Framework

Bohyung Han, *Member*, *IEEE*, Ying Zhu, *Member*, *IEEE*,
Dorin Comaniciu, *Senior Member*, *IEEE*, and Larry S. Davis, *Fellow*, *IEEE*

Abstract—Particle filtering is frequently used for visual tracking problems since it provides a general framework for estimating and propagating probability density functions for nonlinear and non-Gaussian dynamic systems. However, this algorithm is based on a Monte Carlo approach and the cost of sampling and measurement is a problematic issue, especially for high-dimensional problems. We describe an alternative to the classical particle filter in which the underlying density function has an analytic representation for better approximation and effective propagation. The techniques of *density interpolation* and *density approximation* are introduced to represent the likelihood and the posterior densities with Gaussian mixtures, where all relevant parameters are automatically determined. The proposed analytic approach is shown to perform more efficiently in sampling in high-dimensional space. We apply the algorithm to real-time tracking problems and demonstrate its performance on real video sequences as well as synthetic examples.

Index Terms—Bayesian filtering, density interpolation, density approximation, mean shift, density propagation, visual tracking, particle filter.

1 Introduction

Particle filtering is a Bayesian approach to dynamic state estimation based on a sequential Monte Carlo technique. Although it provides tractable solutions to nonlinear and non-Gaussian systems, it frequently suffers from practical issues such as *degeneracy*, *loss of diversity*, and/or *loss of multimodality* [3], [35]. Moreover, to achieve reliable filtering performance, the sample size can grow exponentially as the dimension of the state space increases. To overcome these issues, we explore an analytic approach to approximate and propagate density functions with a mixture of Gaussians and introduce *kernel-based Bayesian filtering* (KBF), which was originally proposed in [16], [17]. The main idea of this work is to maintain an analytic representation of relevant density functions and propagate them over time throughout all steps in a sequential Bayesian filtering framework.

1.1 Related Work

The seminal work for dynamic state estimation is the Kalman filter [20], which provides the optimal solution for linear dynamic systems with Gaussian noise. However, the linear and Gaussian assumption does not hold in many real-world problems, so several suboptimal solutions have been proposed based on various approximations. The Extended Kalman filter (EKF) can handle nonlinear and non-Gaussian systems by linearizing the process and measurement model, but such a first-order approximation has significant limitations for accurate state estimation. More recently, the Unscented Kalman filter (UKF) [19], [36] provides a methodology to better approximate the nonlinearity by deterministic propagation of sigma points and parameter estimation based on the sigma points. However, both the EKF and UKF just estimate and propagate a unimodal Gaussian distribution over time, while many real visual tracking problems involve multimodal distributions.

To overcome such limitations, particle filters [3], [14], [13], [18] based on the sequential Monte Carlo method have been proposed. Particle filtering is very effective in estimating the state in nonlinear and non-Gaussian dynamic systems in a recursive manner, but a significant number of particles are typically required for accurate estimation, especially in high-dimensional problems. To alleviate this problem, various techniques for improving sampling performance and reducing the number of particles have been proposed [12], [23], [27], [34], [31], [32], [33], but they are limited to focusing on the sampling stage in the particle filter framework. Also, particle filtering frequently suffers from degeneracy and loss of diversity problems and requires a resampling step to avoid substantial performance degradation. The Unscented particle filter (UPF) [24], [29] is a combination of an Unscented transformation (UT) and a particle filter and it can reduce the inefficiency of the original particle filter. However, it is

- *B. Han is with the Advanced Project Center, Mobileye Vision Technologies, 12 Venderventer Ave., Princeton, NJ 08542. E-mail: bhhan@cs.umd.edu.*
- *Y. Zhu is with the Real-Time Vision Modeling Department, Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540. E-mail: yingzhu@siemens.com.*
- *D. Comaniciu is with the Integrated Systems Department, Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540. E-mail: dorin.comaniciu@siemens.com.*
- *L.S. Davis is with the Department of Computer Science, University of Maryland, College Park, MD 20742. E-mail: lsd@cs.umd.edu.*

Manuscript received 14 Aug. 2007; revised 14 Apr. 2008; accepted 5 May 2008; published online 20 May 2008.
Recommended for acceptance by F. Dellaert.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-08-0506.
Digital Object Identifier no. 10.1109/TPAMI.2008.134.

also a Monte Carlo technique based on discrete density functions and shares the same problems as particle filters.

A possible solution to obtaining better sampling quality is to integrate continuous multimodal proposal distributions instead of discrete ones [3]. A straightforward solution is the regularized particle filter [13], in which a proposal distribution is constructed by kernel density estimation of current samples. A variation of the regularized particle filter is the kernel particle filter [6], where density modes in the posterior are detected for more effective sampling. However, kernel density estimation and mode detection with a large number of samples is computationally very expensive. On the other hand, Cham and Rehg [5] introduce a piecewise Gaussian function to specify the tracker state in which the selected Gaussian components characterize the neighborhoods around the modes. This idea is applied to multiple hypothesis tracking in a high-dimensional space body tracker, but the sampling and the posterior computation are not straightforward and the accuracy of the posterior density function is not verified. The closest work to ours is [21], where all relevant density functions in sequential Bayesian filtering are represented with Gaussian mixtures. However, that solution may not provide a compact representation for the posterior and the prediction and update steps heavily depend on heuristics.

1.2 Our Approach

Instead of utilizing discrete density functions or some ad hoc integration of continuous density functions, we describe how continuous density functions—Gaussian mixtures—are naturally integrated in a sequential Bayesian filtering framework. In this kernel-based Bayesian filtering, density functions morph and density modes are propagated over time. Density approximation and density interpolation techniques are employed to represent density functions in the state space efficiently and effectively. In both techniques, density functions are represented by Gaussian mixtures, where the number of components and their weights, means, and covariances are automatically determined. The density approximation which is employed to represent the posterior density function is based on a mode-finding algorithm using the mean-shift procedure [10], [11]. Mean-shift mode finding provides a methodology for constructing a compact representation with a small number of Gaussian kernels. A density interpolation technique is introduced to obtain a continuous representation of the measurement density function, which is also a mixture of Gaussians.

For handling nonlinear state transition models with a continuous representation of density functions in a sequential Bayesian filtering framework, we adopt the Unscented transformation [19], [24]. The main advantage of maintaining an analytic representation of the density functions lies in efficient sampling, which is important for solving high-dimensional problems. A multistage sampling strategy is introduced within the density interpolation technique for accurate approximation of the measurement density function. The kernel-based representation for the likelihood of each sample increases the coverage of the state space with a small number of samples. The algorithm is applied to real-time visual tracking, and its performance is demonstrated through various experiments.

This paper is organized as follows: Section 2 introduces the new density propagation technique in the sequential Bayesian filtering framework. Sections 3 and 4 explain the density interpolation and the density approximation methods, respectively. Section 5 demonstrates its performance by various simulation results with synthetic examples. Finally, Section 6 shows object tracking in synthetic examples and real videos.

2 Kernel-Based Bayesian Filtering

In this section, we present the overall structure of KBF, where the relevant density functions are approximated by a kernel-based representation and propagated over time.

2.1 Overview

In a dynamic system, the process and measurement models are given by

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{u}_t), \tag{1}$$

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{v}_t), \tag{2}$$

where $\mathbf{v}_t$ and $\mathbf{u}_t$ are the process and the measurement noises, respectively. The state variable $\mathbf{x}_t$ $(t = 0, \ldots, n)$ is characterized by its probability density function (pdf) estimated from the sequence of measurements $\mathbf{z}_t$ $(t = 1, \ldots, n)$. In the sequential Bayesian filtering framework, the conditional density of the state variable given the measurements is propagated through prediction and update stages as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1}, \tag{3}$$

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{1}{k} p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}), \tag{4}$$

where $k = \int p(\mathbf{z}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) d\mathbf{x}_t$ is a normalization constant independent of $\mathbf{x}_t$. Also, $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$ represents the prior pdf and $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ and $p(\mathbf{z}_t|\mathbf{x}_t)$ are the predicted pdf and the measurement likelihood function, respectively. The posterior pdf at time step t, $p(\mathbf{x}_t|\mathbf{z}_{1:t})$, is used as the prior pdf at time step $t+1$.

At each time step, the conditional distribution of the state variable $\mathbf{x}$ given a sequence of measurements $\mathbf{z}$ is represented by a Gaussian mixture. Our goal is to retain such a representation through the stages of prediction and update and to represent the posterior probability in the following step with the same mixture form.

The proposed filtering framework is described as follows: First, the UT [19], [24] is used to derive a mixture representation of the predicted pdf $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$. Second, a density interpolation technique with multistage sampling is integrated to approximate the likelihood function with a mixture form. By multiplying two mixture functions, the posterior pdf is obtained through (4). To prevent the number of mixands from growing too large, a density approximation algorithm based on mode finding is applied to construct a compact representation for the posterior pdf.

2.2 Prediction by Unscented Transform

The state estimation problem in general dynamic systems typically involves highly nonlinear process models, which requires additional observations for better construction of the posterior density function. Therefore, accurate prediction is critical for reducing the cost of the observation and improving estimation accuracy. However, it is generally hard to find a closed-form solution for the nonlinear process model. The UT is an appropriate technique for handling nonlinear process models since it is accurate up to the second order of Taylor expansion. This section describes the unscented transformation.

Denote by $\mathbf{x}_t^i$ $(i = 1, \ldots, n_t)$ the set of means in $\mathbb{R}^d$ of the components of a Gaussian mixture and by $\mathbf{P}_t^i$ the corresponding covariance matrices at time step t. Let each Gaussian have a weight κ_t^i, with $\sum_{i=1}^{n_t} \kappa_t^i = 1$, and let the prior density function be given by

$$\begin{aligned} &p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) \\ &= \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n_{t-1}} \frac{\kappa_{t-1}^i}{|\mathbf{P}_{t-1}^i|^{1/2}} \exp\left(-\frac{1}{2} D^2\left(\mathbf{x}_{t-1}, \mathbf{x}_{t-1}^i, \mathbf{P}_{t-1}^i\right)\right), \end{aligned} \tag{5}$$

where

$$D^2(\mathbf{x}_{t-1}, \mathbf{x}_{t-1}^i, \mathbf{P}_{t-1}^i) \equiv \left(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^i\right)^\top \left(\mathbf{P}_{t-1}^i\right)^{-1} \left(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^i\right). \tag{6}$$

The unscented transformation [19], [24] is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation. For each mode, it first chooses $2d + 1$ sigma points and their weights, which are given by

$$\begin{aligned} \mathcal{X}_{t-1}^{(i,0)} &= \mathbf{x}_{t-1}^i, \\ \mathcal{X}_{t-1}^{(i,j)} &= \mathbf{x}_{t-1}^i - \left(\sqrt{(d+\lambda)\mathbf{P}_{t-1}^i}\right)_j \quad j = 1, \ldots, d, \\ \mathcal{X}_{t-1}^{(i,j)} &= \mathbf{x}_{t-1}^i + \left(\sqrt{(d+\lambda)\mathbf{P}_{t-1}^i}\right)_{j-d} \quad j = d+1, \ldots, 2d, \\ W_{(i,0)} &= \lambda/(d+\lambda), \\ W_{(i,j)} &= 1/2(d+\lambda) \quad j = 1, \ldots, 2d, \end{aligned} \tag{7}$$

where λ is a scaling parameter, and $(\sqrt{(d+\lambda)\mathbf{P}_{t-1}^i})_j$ is the jth row or column of the matrix square root of $(d+\lambda)\mathbf{P}_{t-1}^i$. $W_{(i,j)}$ is the weight associated with the jth sigma point, where $\sum_{j=0}^{2d} W_{(i,j)} = 1$. These sigma vectors are propagated through the nonlinear function

$$\mathcal{X}_t^{(i,j)} = g\left(\mathcal{X}_{t-1}^{(i,j)}, \mathbf{u}_t\right) \quad j = 0, \ldots, 2d \tag{8}$$

and the mean $\bar{\mathbf{x}}_t^i$ and covariance $\bar{\mathbf{P}}_t^i$ of the ith mode in the predicted density are approximated using a weighted sample mean and covariance of the sigma points

$$\begin{aligned} \bar{\mathbf{x}}_t^i &= \sum_{j=0}^{2d} W_{(i,j)} \mathcal{X}_t^{(i,j)}, \\ \bar{\mathbf{P}}_t^i &= \sum_{j=0}^{2d} W_{(i,j)} \left(\mathcal{X}_t^{(i,j)} - \bar{\mathbf{x}}_t^i\right)\left(\mathcal{X}_t^{(i,j)} - \bar{\mathbf{x}}_t^i\right)^\top + \mathbf{Q}, \end{aligned} \tag{9}$$

where $\mathbf{Q}$ is the covariance matrix for the process noise.

The unscented transformation is applied to every mode in the density (5) independently and the density after prediction is given by

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n_{t-1}} \frac{\bar{\kappa}_t^i}{|\bar{\mathbf{P}}_t^i|^{1/2}} \exp\left(-\frac{1}{2} D^2\left(\mathbf{x}_t, \bar{\mathbf{x}}_t^i, \bar{\mathbf{P}}_t^i\right)\right), \tag{10}$$

where $\bar{\kappa}_t^i = \kappa_{t-1}^i$. The advantage of the unscented transformation is illustrated clearly in [24, Fig. 1].

2.3 Multistage Sampling and Interpolation of Measurement Likelihood

In the measurement step of KBF, a continuous approximation of the likelihood function is interpolated from discrete samples. A multistage sampling scheme is introduced to improve the approximation progressively. The advantage of the analytic representation and multistage sampling is that it provides a global view of the landscape of the likelihood function and supports efficient sample placement.

2.3.1 Multistage Sampling

Unlike the Sampling Importance Resampling (SIR) algorithm [18], which uses the predicted pdf as the proposal distribution, we employ a multistage sampling strategy and progressively update the proposal function based on observations. The predicted pdf is used as the initial proposal distribution q_0 as

$$q_0(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{z}_{1:t-1}). \tag{11}$$

Assume that, in total, N samples are to be drawn to obtain the measurement density function. In our multistage sampling scheme, N/m samples are drawn in the first stage from the initial proposal distribution (11), where m is the number of sampling stages. An initial approximation of the likelihood function $p_1(\mathbf{z}_t|\mathbf{x}_t)$ is obtained through interpolation with Gaussian kernels. Details of the density interpolation algorithm are provided in Section 3. The proposal function is then updated by a linear combination of the initial proposal distribution and the current approximation of the likelihood function $p_1(\mathbf{z}_t|\mathbf{x}_t)$. We repeatedly approximate the likelihood function from available samples and update the proposal distribution using samples with nonzero weights as follows:

$$p_j(\mathbf{z}_t|\mathbf{x}_t) = \frac{1}{(2\pi)^{d/2}} \sum_{\tau_i \neq 0} \frac{\tau_t^i}{|\mathbf{R}_t^i|} \exp\left(-\frac{1}{2} D^2\left(\mathbf{x}_t, \mathbf{x}_t^i, \mathbf{R}_t^i\right)\right), \tag{12}$$

$$q_j(\mathbf{x}_t) = (1 - \alpha_j) q_{j-1}(\mathbf{x}_t) + \alpha_j \frac{p_j(\mathbf{z}_t|\mathbf{x}_t)}{\int p_j(\mathbf{z}_t|\mathbf{x}_t) d\mathbf{x}_t}, \tag{13}$$

where $i = 1, \ldots, \frac{j}{m}N$, $j = 1, \ldots, m$, and $\alpha_j \in [0, 1]$ is the *adaptation rate*. The kernel bandwidth $\mathbf{R}_t^i$ for each sample $\mathbf{x}_t^i$ is determined based on the distance to the kth nearest neighbor in each dimension; the details are described in Section 3.1.

Since the observation information is incorporated into the proposal distribution to guide sampling, the multistage sampling strategy explores the likelihood surface more efficiently than conventional particle filters. Thus, it is especially advantageous in dealing with a high-dimensional state space.

2.3.2 *Approximation of Likelihood Function*

As discussed previously, the measurement likelihood is estimated through multistage sampling. With samples drawn from the improved proposal distributions, intermediate likelihood functions are constructed and used to update the proposal distributions. In each stage, the intermediate likelihood density function is obtained by a Non-Negative Least Square (NNLS) method [22]. After an m-step repetition of this procedure, the final measurement distribution is obtained. Algorithm 1 presents the complete procedure to compute the likelihood function, and the final measurement function with m_t Gaussians at time t is given by

$$p(\mathbf{z}_t|\mathbf{x}_t) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{m_t} \frac{\tau_t^i}{\left|\mathbf{R}_t^i\right|^{1/2}} \exp\left(-\frac{1}{2} D^2\left(\mathbf{x}_t, \mathbf{x}_t^i, \mathbf{R}_t^i\right)\right), \quad (14)$$

where τ_t^i, $\mathbf{x}_t^i$, and $\mathbf{R}_t^i$ are the weight, mean, and covariance matrix of the ith kernel. Note that the measurement density is a function of the state variable $\mathbf{x}_t$ and that the measurement variable $\mathbf{z}_t$ is not shown in the RHS of (14).

Algorithm 1 Measurement Step

```
1:  $S_t = \phi$, where $S_t$ represents the sample set.
2:  Set the initial proposal distribution $q_0(\mathbf{x}_t)$ to the
    predicted pdf
    $q_0(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$.
3:  for $i = 1$ to $m$ do
4:    Draw samples from proposal distribution and update
      sample set
      $S_t^i = \{s_i^{(j)}|s_i^{(j)} \sim q_{i-1}(\mathbf{x}_t), j = 1, \dots, \frac{N}{m}\}$, $S_t = S_t \cup S_t^i$.
5:    Assign a Gaussian kernel for each element in $S_t^i$
      $\mathbf{m}_{(i-1)\frac{N}{m}+j} = s_i^{(j)}$
      $\mathbf{Q}_{(i-1)\frac{N}{m}+j} = c \, \mathrm{diag}(\mathrm{KNN}_1(k) \dots \mathrm{KNN}_d(k))^2 \, \mathbf{I}$.      (23)
6:    Compute likelihood of each new sample using a
      measurement function $h$
      $l_{(i-1)\frac{N}{m}+j} = h(\mathbf{m}_{(i-1)\frac{N}{m}+j}, \mathbf{v}_t)$, where $\mathbf{v}_t$ is a variable for
      measurement noise.
7:    Obtain $\mathbf{A}$ and $\mathbf{b}$ for every element in $S_t$.      (25) (26)
8:    Compute the weight for each kernel by NNLS
      $\mathbf{w} = nnls(\mathbf{A}, \mathbf{b})$.      (27)
9:    Obtain the measurement density function at the
      current step
      $p_i(\mathbf{z}_t|\mathbf{x}_t) = \sum_{\tau_t^j \neq 0} \mathcal{N}(\tau_t^j, \mathbf{x}_t^j, \mathbf{R}_t^j)$, where $\tau_t = \mathbf{w}$, $\mathbf{x}_t = \mathbf{m}$,
      and $\mathbf{R}_t = \mathbf{Q}$.
10:   Update new proposal distribution
      $q_i(\mathbf{x}_t) = (1-\alpha_i) q_{i-1}(\mathbf{x}_t) + \alpha_j \frac{p_i(\mathbf{z}_t|\mathbf{x}_t)}{\int p_i(\mathbf{z}_t|\mathbf{x}_t) d\mathbf{x}_t}$.      (13)
11: end for
12: Final measurement density function
    $p(\mathbf{z}_t|\mathbf{x}_t) = p_m(\mathbf{z}_t|\mathbf{x}_t)$.
```

2.4 Update

Since both the predicted pdf and the measurement functions are represented by Gaussian mixtures, the posterior pdf, as the product of two Gaussian mixtures, can also be represented by a Gaussian mixture. Denoting the Gaussian components of the predicted pdf and the likelihood function by $\mathcal{N}(\bar{\kappa}_t^i, \bar{\mathbf{x}}_t^i, \bar{\mathbf{P}}_t^i)$ $(i = 1, \dots, n_{t-1})$ and $\mathcal{N}(\tau_t^j, \mathbf{x}_t^j, \mathbf{R}_t^j)$ $(j = 1, \dots, m_t)$, respectively, the product of the two distributions is given as follows:

$$\begin{aligned} &\left(\sum_{i=1}^{n_{t-1}} \mathcal{N}\left(\bar{\kappa}_t^i, \bar{\mathbf{x}}_t^i, \bar{\mathbf{P}}_t^i\right)\right)\left(\sum_{j=1}^{m_t} \mathcal{N}\left(\tau_t^j, \mathbf{x}_t^j, \mathbf{R}_t^j\right)\right) \\ &\quad = \sum_{i=1}^{n_{t-1}} \sum_{j=1}^{m_t} \mathcal{N}\left(\omega_t^{ij}, \mathbf{m}_t^{ij}, \mathbf{\Sigma}_t^{ij}\right), \end{aligned} \quad (15)$$

where

$$\omega_t^{ij} = \frac{\bar{\kappa}_t^i \tau_t^j \exp\left(-\frac{1}{2} D^2\left(\mathbf{x}_t^j, \bar{\mathbf{x}}_t^i, \bar{\mathbf{P}}_t^i + \mathbf{R}_t^j\right)\right)}{(2\pi)^{d/2}\left|\bar{\mathbf{P}}_t^i + \mathbf{R}_t^j\right|^{1/2}}, \quad (16)$$

$$\mathbf{m}_t^{ij} = \mathbf{\Sigma}_t^{ij}\left(\left(\bar{\mathbf{P}}_t^i\right)^{-1} \mathbf{x}_t^i + \left(\mathbf{R}_t^j\right)^{-1} \mathbf{x}_t^j\right), \quad (17)$$

$$\mathbf{\Sigma}_t^{ij} = \left(\left(\bar{\mathbf{P}}_t^i\right)^{-1} + \left(\mathbf{R}_t^j\right)^{-1}\right)^{-1}. \quad (18)$$

The resulting density function in (15) is a weighted mixture of Gaussians with $n_{t-1} \times m_t$ components. However, the exponential increase in the number of components over time could make the whole procedure intractable. In order to avoid this situation, a density approximation technique is proposed to maintain a compact yet accurate density representation, even after density propagation through many time steps. Details of the density approximation algorithm are given in Section 4.

After the update step, the final posterior distribution is given by

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n_t} \frac{\kappa_t^i}{\left|\mathbf{P}_t^i\right|^{1/2}} \exp\left(-\frac{1}{2} D^2\left(\mathbf{x}_t, \mathbf{x}_t^i, \mathbf{P}_t^i\right)\right), \quad (19)$$

where n_t is the number of components at time step t.

Fig. 1 illustrates the density functions throughout kernel-based Bayesian filtering.

2.5 State Estimation

In the conventional particle filters such as the Condensation algorithm, the final target state is typically determined by a function of particle locations and their weights; the common choices are weighted mean or maximum a posteriori (MAP) solution. However, the output of the weighted mean approach may be located in the middle of a multimodal distribution, which can be far from any local maximum, and MAP overly depends on local information (or a small number of samples), which is not sufficiently resistant to measurement noise. So, they suffer from instability of the estimation, especially when multiple modes compete due to occlusion and clutter. To mitigate this problem, we employ a fusion technique for estimating the target state, where the most significant mode in the underlying density function is found by tracking the mode in a multiscale optimization framework [9]. In this technique, the local and global information in the posterior are considered together to obtain the final target state. This can easily be done since the

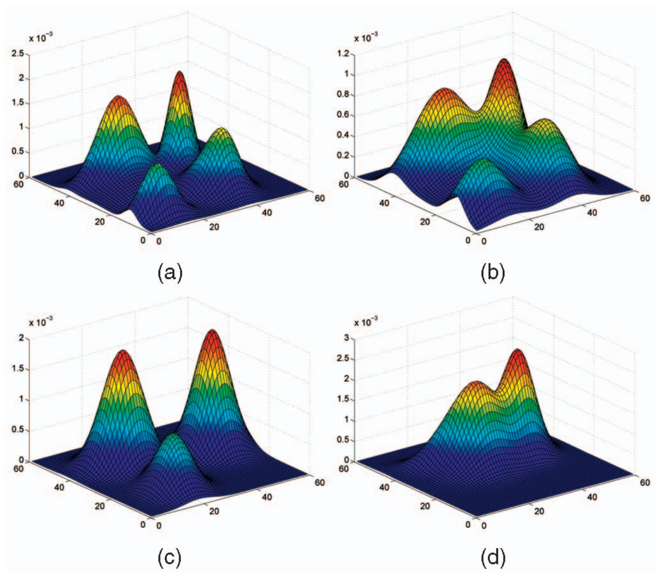

Fig. 1. Density function in each step of kernel-based Bayesian filtering. (a) Prior. (b) Prediction. (c) Measurement. (d) Posterior.

posterior density function is represented with a mixture of Gaussians as in (20).

We first perform mode detection using large covariance matrices of the form $\mathbf{C}_i = (1+\beta)\mathbf{P}_i$, where the parameter β is large enough to ensure that the density function is unimodal. Then, the sample point density estimator computed at point $\mathbf{x}$ is given by

$$\hat{f}(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n} \frac{\kappa_i}{|\mathbf{C}_i|^{1/2}} \exp\left(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}_i, \mathbf{C}_i)\right), \qquad (20)$$

where the time variable t has been dropped from (19) for convenience of representation. Since the initial density function with large β is assumed to be unimodal, the convergence point of the mean-shift mode-finding procedure is actually the global maximum.

Once the initial mode is determined, the mode location is tracked across scales by successively reducing β and performing mode detection again and again until there is no further change in convergence location. Note that β decreases to zero in the last mode detection iteration, and the bandwidth matrix associated with each data point becomes equal to the point covariance matrix, i.e., $\mathbf{C}_i = \mathbf{P}_i$, $i = 1, \ldots, n$.

Denote by $\hat{\mathbf{x}}_m$ the location of the most significant mode. Since the gradient at $\hat{\mathbf{x}}_m$ is zero, we have $\mathbf{m}(\hat{\mathbf{x}}_m) = \mathbf{0}$, which means

$$\hat{\mathbf{x}}_m = \left(\sum_{i=1}^{n} \omega_i(\hat{\mathbf{x}}_m)\mathbf{C}_i^{-1}\right)^{-1} \sum_{i=1}^{n} \omega_i(\hat{\mathbf{x}}_m)\mathbf{C}_i^{-1}\mathbf{x}_i, \qquad (21)$$

where

$$\omega_i(\mathbf{x}) = \frac{\frac{1}{|\mathbf{C}_i|^{1/2}} \exp\left(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}_i, \mathbf{C}_i)\right)}{\sum_{i=1}^{n} \frac{1}{|\mathbf{C}_i|^{1/2}} \exp\left(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}_i, \mathbf{C}_i)\right)}. \qquad (22)$$

The most significant mode obtained by multiscale mode tracking using Gaussian mixture density estimation corresponds to the target location in our application. However, note that the detected mode may not be equal to the point with the highest density or the mode with the largest local weight. This means that the most significant mode is determined by neighborhood information, in addition to local information, which is realized by the multiscale implementation. The procedure of the final state estimation and its property is illustrated in Fig. 2.

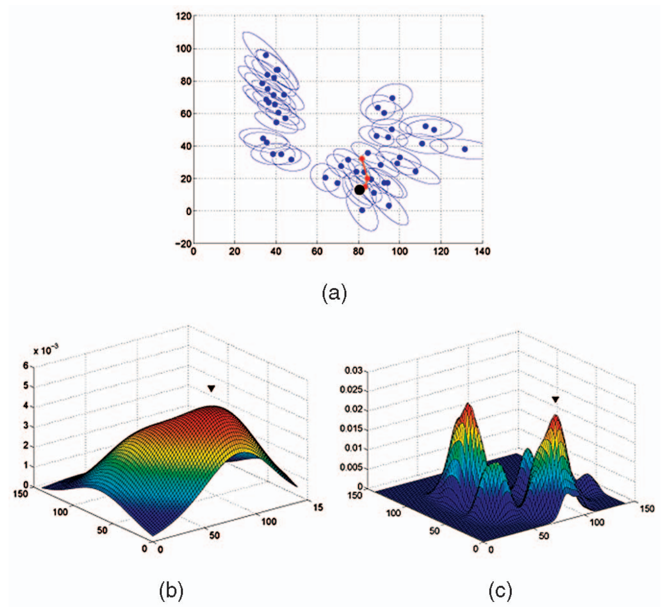

Fig. 2. Simulation result of the fusion-based state estimation. (a) Initial sample locations with covariance matrix estimation and trajectory of mode tracking across scales (red dots and lines). The black dot represents the final convergence location. (b) Initial density function with $\beta = 10$. (c) Final density estimate with $\beta = 0$. (The triangles in (b) and (c) indicate mode tracking estimates.)

Fig. 3 illustrates the performance of state estimation based on the proposed method. A synthetic example is provided in Fig. 3a, where the estimated state based on the weighted mean approach is in the middle of three major modes in the density function, but the result of our method finds the most significant mode. Note that the most significant mode in this example is different from the global maximum in the density function. A real example is presented in Figs. 3b and 3c in which a human face is tracked by KBF. The states estimated by the weighted mean and density fusion method are indicated by blue and white rectangle in Fig. 3b. Also, the contour of the posterior density function[1] is superimposed on the image in Fig. 3c, which clearly illustrates the difference between the results of the weighted mean (square marker) and density fusion (circle marker). It would be more interesting to see the detail of the posterior density function. The mixture weights of the modes on the left and right are around 0.4 and 0.6, respectively,[2] and the probabilities at the peaks are almost the same because the mode on the right has a wider covariance. So, the state estimation in this case is quite ambiguous, but the result based on the density fusion

1. For visualization, the original 3D density function is projected to 2D image plane.
2. Note that the mode with a smaller weight is selected in the fusion process.

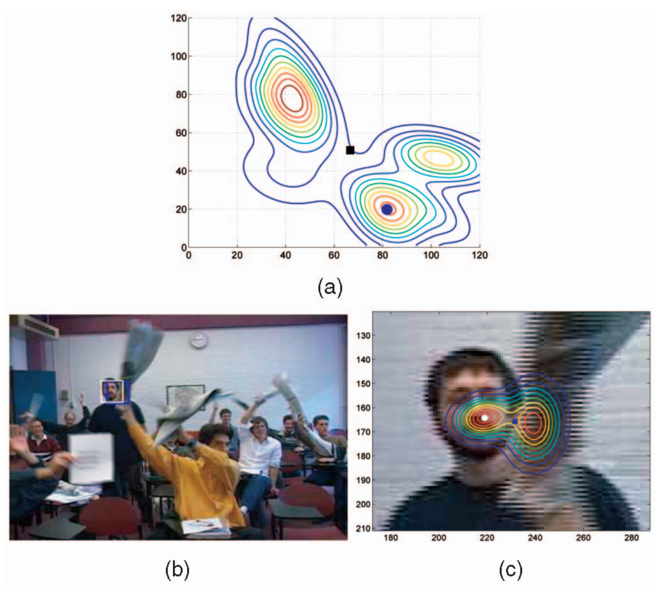

Fig. 3. State estimation in KBF. (a) Synthetic example. The square and circle represent the estimated state by the weighted mean and the most significant mode, respectively. (b) and (c) Real example. The blue and white rectangle (or marker) in both figures represent the estimated state by the weighted mean and the most significant mode, respectively.

approach is more reasonable than the weighted mean solution. We performed tracking in the same sequence with Condensation algorithm and the estimated state coincides with the location estimated by the weighted mean solution.

3 Density Interpolation

In this section, we describe the density interpolation algorithm, which represents the measurement density function with a mixture of Gaussians. In the measurement step of sequential Bayesian filtering, the likelihood values are known for a set of samples, and the likelihood density surface can be interpolated from sample likelihoods using a Gaussian mixture. The main objective of this step is to interpolate measurement density surface using Gaussian kernels as in [28], where the number of Gaussians is much less than the number of samples.

3.1 Initial Scale Selection

One of the limitations of kernel-based methods is that they involve the specification of a scale parameter. Various research has been performed for the scale selection problem [1], [25], [30], but it is very difficult to find the optimal scale in general. Below, we present a strategy to determine the scale parameter for density estimation based on the statistics of the *k-nearest neighbors*.

The basic idea of this method is very simple, and similar approaches are discussed in [7] and [8]. Each sample is intended to cover the local region around itself in the d-dimensional state space with its scale. For this purpose, kth nearest neighbors (KNNs) are used and the kernel bandwidth (scale) is proportional to the distance to the KNN of a sample. Define $\mathrm{KNN}_j^i(k)$ $(1 \le j \le d)$ to be the distance to the KNN from sample i in the jth dimension; then, the covariance matrix $\mathbf{P}_i$ for the ith sample is given by

$$\mathbf{P}_i = c \operatorname{diag}\big(\mathrm{KNN}_1^i(k)\ \mathrm{KNN}_2^i(k) \ldots \mathrm{KNN}_d^i(k)\big)^2 \mathbf{I}, \tag{23}$$

where c is a constant that depends on the number of samples and the dimensionality and $\mathbf{I}$ is the d-dimensional identity matrix.

By this method, samples in dense areas have small scales and the density will be represented accurately, but sparse areas that are not represented as accurately convey only relatively rough information about the density function.

3.2 Interpolation

A Gaussian kernel is assigned to each sample; the mean and covariance are set to the sample location and the scale initialized by the method in Section 3.1, respectively. When the likelihood value of each sample is given, the weight for each kernel can be computed by the Non-Negative Least Squares (NNLS) method [22].

Denote $\mathbf{x}_i$ as the mean location and $\mathbf{P}_i$ as the covariance matrix for the ith sample $(i = 1, \ldots, n)$. Also, suppose that l_i is the likelihood value of the ith sample. The likelihood at $\mathbf{x}_j$ induced by the ith kernel is given by

$$p_i(\mathbf{x}_j) = \frac{1}{(2\pi)^{d/2} |\mathbf{P}_i|^{1/2}} \exp\left(-\frac{1}{2} D^2(\mathbf{x}_j, \mathbf{x}_i, \mathbf{P}_i)\right). \tag{24}$$

Define an $n \times n$ matrix $\mathbf{A}$ having entry $p_i(\mathbf{x}_j)$ in (i, j) and an $n \times 1$ vector $\mathbf{b}$ having l_i in its ith row as

$$\mathbf{A} = \begin{pmatrix} p_1(\mathbf{x}_1) & p_1(\mathbf{x}_2) & \cdots & p_1(\mathbf{x}_n) \\ p_2(\mathbf{x}_1) & p_2(\mathbf{x}_2) & \cdots & p_2(\mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ p_n(\mathbf{x}_1) & p_n(\mathbf{x}_2) & \cdots & p_n(\mathbf{x}_n \end{pmatrix}, \tag{25}$$

$$\mathbf{b} = (l_1, l_2, \cdots, l_n)^\top. \tag{26}$$

Then, the weight vector $\mathbf{w} = (\tau_1, \tau_2, \ldots, \tau_n)^\top$ can be computed by solving the following constrained least square problem:

$$\begin{aligned} &\min_{\mathbf{w}} \|\mathbf{A}\mathbf{w} - \mathbf{b}\|^2 \\ &\text{subject to } \ \tau_i \ge 0 \quad \text{for } i = 1, \ldots, n, \end{aligned} \tag{27}$$

and it is denoted by $\mathbf{w} = nnls(\mathbf{A}, \mathbf{b})$. The size of matrix $\mathbf{A}$ is determined by the number of samples. When the sample size is large, sparse matrix methods can be used to solve for $\mathbf{w}$ efficiently [2], [4]. Although the number of unknowns and the number of equations in the optimization problem in (27) are equal, we need a least square method since all solutions must be nonnegative.

Usually, many of the weights will be zero and the final density function will be a mixture of Gaussians with a small number of components. The density interpolation simulates the heavy-tailed density function more accurately than the density approximation introduced in Section 4, while the density approximation generally produces a more compact representation.

3.3 Performance of Interpolation

Fig. 4 shows one-dimensional density interpolation results. For each case, 100 samples are drawn, and the initial scale for each sample is given as explained in Section 3.1. The estimated density function approximates the original density very accurately, as seen in Fig. 4. Two

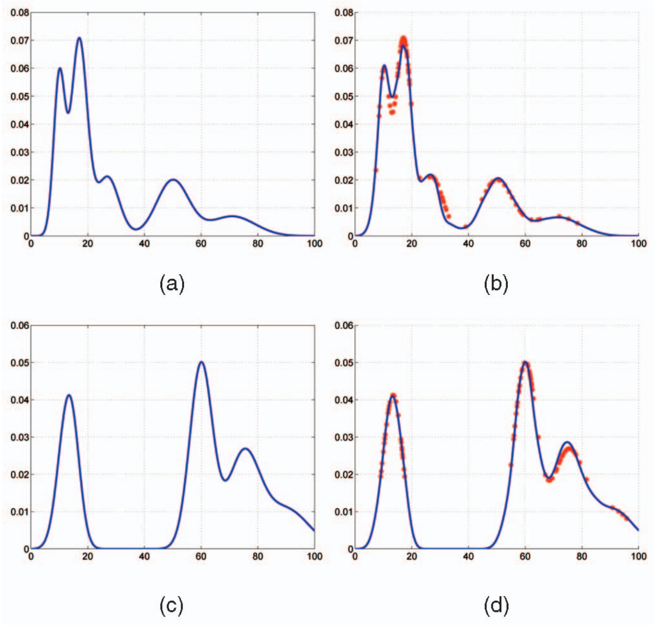

Fig. 4. Two examples of original density functions and their interpolations. In the interpolation graphs (right), red dots represent the sample locations and likelihoods (100 samples). In cases (a) and (b), 32 and 29 components have nonzero weights, respectively. (a) Example 1. (b) Example 2.

different Gaussian mixtures—$\mathcal{N}(0.2, 10, 2^2)$, $\mathcal{N}(0.35, 17, 4^2)$, $\mathcal{N}(0.15, 27, 8^2)$, $\mathcal{N}(0.2, 50, 16^2)$, and $\mathcal{N}(0.1, 71, 32^2)$ in example 1 and $\mathcal{N}(0.15, 12, 5^2)$, $\mathcal{N}(0.1, 15, 4^2)$, $\mathcal{N}(0.35, 60, 8^2)$, $\mathcal{N}(0.25, 75, 16^2)$, and $\mathcal{N}(0.15, 90, 32^2)$ in example 2—are tested for the interpolation. The important difference between density approximation and density interpolation can be found in Fig. 4d, where the area around 90 is accurately estimated by density interpolation although no local maximum is detected there.

When 50 independent realizations are performed, Mean Integrated Squared Error (MISE) and its variance are very small for both examples as shown in Table 1.

Also, a multidimensional density function is interpolated in the same manner, and its performance is discussed next. In Fig. 5, the density interpolation produces a very accurate and stable result when 200 samples are drawn from the original density function ($\mathrm{MISE} = 4.5467 \times 10^{-9}$ and $\mathrm{VAR} = 7.3182 \times 10^{-18}$ on average over 50 runs).

These results show that the density interpolation has reasonable accuracy to approximate a density function given samples and their corresponding likelihoods.

4 Density Approximation

The product of two pdfs—Gaussian mixtures—from the prediction and measurement steps is also a Gaussian mixture, but the output density function needs to be simplified to avoid an exponential increase in the number of Gaussian components. The density approximation technique described in this section provides a method to maintain a compact representation of the density by an iterative mode detection procedure and curvature-based covariance estimation [15].

TABLE 1
Error of Density Interpolation

	MISE	VAR
example 1	3.9479×10^{-5}	2.5613×10^{-9}
example 2	2.6871×10^{-5}	9.5103×10^{-10}

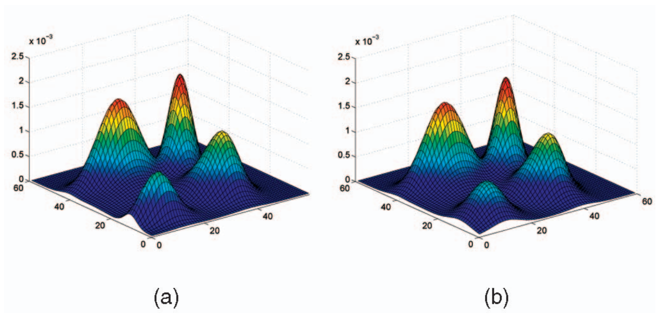

Fig. 5. Comparison between the original density function and density interpolation (2D). (a) Original density function. (b) Density interpolation with 30 nonzero weight components.

4.1 Mode Detection and Density Approximation

Suppose that κ_i, $\mathbf{x}_i$, and $\mathbf{P}_i$ $(i = 1 \dots n)$ characterizes a Gaussian kernel with weight κ_i, mean $\mathbf{x}_i$, and covariance $\mathbf{P}_i$ in the d-dimensional state space, where $\sum_{i=1}^{n} \kappa_i = 1$. Then, we define the sample point density estimator computed at point $\mathbf{x}$ by

$$\hat{f}(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n} \frac{\kappa_i}{|\mathbf{P}_i|^{1/2}} \exp\left(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}_i, \mathbf{P}_i)\right). \tag{28}$$

Our purpose is to obtain a compact representation of the density function which is a Gaussian mixture. The mode location and its weight are found by a mean-shift algorithm, and the covariance matrix associated with each mode is computed using the Hessian matrix.

To find the gradient ascent direction at $\mathbf{x}$, the variable-bandwidth mean-shift vector at $\mathbf{x}$ is given by

$$\mathbf{m}(\mathbf{x}) = \left(\sum_{i=1}^{n} \omega_i(\mathbf{x}) \mathbf{P}_i^{-1}\right)^{-1} \left(\sum_{i=1}^{n} \omega_i(\mathbf{x}) \mathbf{P}_i^{-1} \mathbf{x}_i\right) - \mathbf{x}, \tag{29}$$

where the weights

$$\omega_i(\mathbf{x}) = \frac{\kappa_i |\mathbf{P}_i|^{-1/2} \exp\left(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}_i, \mathbf{P}_i)\right)}{\sum_{i=1}^{n} \kappa_i |\mathbf{P}_i|^{-1/2} \exp\left(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}_i, \mathbf{P}_i)\right)} \tag{30}$$

satisfy $\sum_{i=1}^{n} \omega_i(\mathbf{x}) = 1$. By computing the mean-shift vector $\mathbf{m}(\mathbf{x})$ and translating the location $\mathbf{x}$ by $\mathbf{m}(\mathbf{x})$ iteratively, a local maximum of the underlying density function is detected. A formal check for the maximum involves the computation of the Hessian matrix

$$\begin{aligned} \hat{\mathbf{H}}(\mathbf{x}) = {} & \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n} \frac{\kappa_i}{|\mathbf{P}_i|^{1/2}} \exp\left(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}_i, \mathbf{P}_i)\right) \\ & \times \mathbf{P}_i^{-1} \Big((\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^\top - \mathbf{P}_i\Big) \mathbf{P}_i^{-1}, \end{aligned} \tag{31}$$

which should be negative definite. If it is not negative definite, the convergence point might be a saddle point or a local minimum. In this case, kernels associated with such

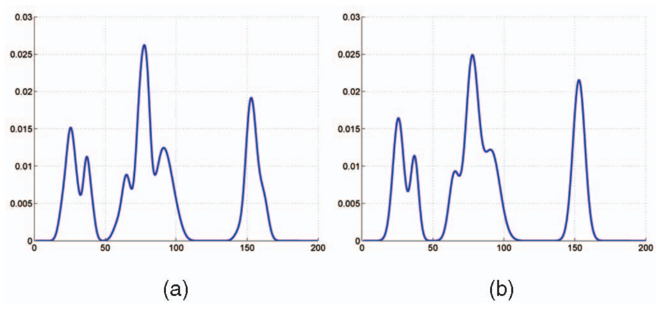

Fig. 6. Comparisons between kernel density estimation and density approximation (1D). For the approximation, 200 samples are drawn from the original distribution—$\mathcal{N}(0.15, 25, 10^2)$, $\mathcal{N}(0.1, 37, 8^2)$, $\mathcal{N}(0.15, 65, 16^2)$, $\mathcal{N}(0.25, 77, 9^2)$, and $\mathcal{N}(0.15, 91, 30^2)$, $\mathcal{N}(0.2, 154, 15^2)$. (a) Kernel density estimation $(\mathrm{MISE} = 1.2556 \times 10^{-4}$. (b) Density approximation $(\mathrm{MISE} = 1.7011 \times 10^{-4})$.

modes should be restored and considered as separate modes for further processing.

The approximate density is obtained by detecting the mode location for every sample point $\mathbf{x}_i$ and assigning a single Gaussian kernel for each mode. Suppose that the approximate density has n' unique modes of $\tilde{\mathbf{x}}_j$ $(j = 1 \ldots n')$ with an associated weight $\tilde{\kappa}_j$, which is equal to the sum of the kernel weights that converge to $\tilde{\mathbf{x}}_j$. The Hessian matrix $\hat{\mathbf{H}}_j$ of each mode is used for the computation of $\tilde{\mathbf{P}}_j$ as follows:

$$\tilde{\mathbf{P}}_j = \frac{\tilde{\kappa}_j^{\frac{2}{d+2}}}{\left|2\pi(-\hat{\mathbf{H}}_j^{-1})\right|^{\frac{1}{d+2}}}\left(-\hat{\mathbf{H}}_j^{-1}\right). \quad (32)$$

The basic idea of (32) is to fit the covariance using the curvature in the neighborhood of the mode. The final density approximation is then given by

$$\tilde{f}(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}}\sum_{i=1}^{n'}\frac{\tilde{\kappa}_i}{|\tilde{\mathbf{P}}_i|^{1/2}}\exp\left(-\frac{1}{2}D^2(\mathbf{x}, \tilde{\mathbf{x}}_i, \tilde{\mathbf{P}}_i)\right). \quad (33)$$

Note that n' is typically much smaller than n and that no significant increase in the number of components in the mixture is observed in all of our tracking examples. The approximation error $\|\hat{f}(\mathbf{x}) - \tilde{f}(\mathbf{x})\|$ can be evaluated straightforwardly.

4.2 Performance of Approximation

In Bayesian filtering, the preservation and propagation of density modes in the posterior is very important for accurate sequential state estimation. The accuracy of the density approximation is demonstrated in Fig. 6. From a one-dimensional distribution composed of six weighted Gaussians, 200 samples are drawn and the density function is constructed by the density approximation technique based on the samples. The result is presented in Fig. 6, which shows the performance of density approximation. In both examples, there are minor errors compared with the density functions constructed by kernel density estimation, but major mode locations are well preserved. The MISE between the original density function and estimated density functions is calculated and presented for the error estimation. Fig. 7 illustrates the performance of the density approximation with 400 samples in a 2D example.

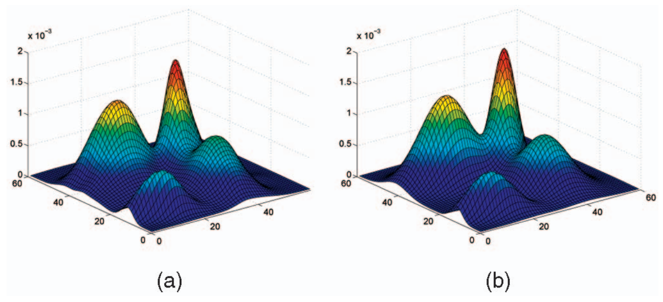

Fig. 7. Comparison between kernel density estimation and density approximations (2D, 400 samples). (a) Kernel density estimation $(\mathrm{MISE} = 1.1453 \times 10^{-8}$. (b) Density approximation $(\mathrm{MISE} = 1.5237 \times 10^{-8})$.

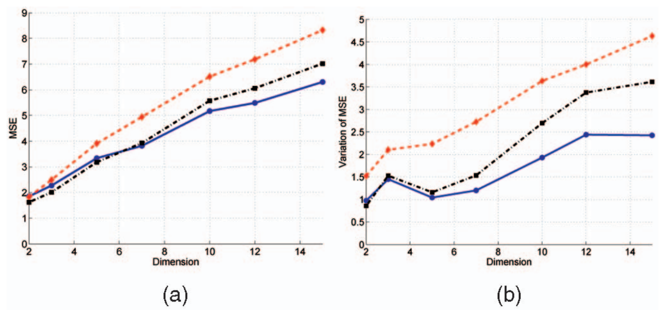

Fig. 8. MSE and variance of MSE. kernel-based Bayesian filtering with 20 particles (blue solid line), SIR with 20 particles (red dashed line), and SIR filter with 200 particles (black dotted line) for model 1. (a) Error. (b) Variance.

5 Simulation

In this section, synthetic tracking examples are simulated, and the performance of the KBF is compared with the SIR algorithm [3], [18]. Two different process models—one linear and the other nonlinear—are selected, and simulations are performed for various dimensions—2D, 3D, 5D, 10D, 12D, and 15D. The accumulated Mean Squared Error (MSE) through 50 time steps is calculated in each run, and 50 identical experiments are conducted based on the same data for error estimation.

The first process model is given by the following equation:

$$\mathbf{x}_t = \frac{\mathbf{x}_{t-1}}{2} + \frac{25\mathbf{x}_{t-1}}{1 + \mathbf{x}_{t-1}^\top\mathbf{x}_{t-1}} + 8\cos(1.2(t-1))\mathbf{1} + \mathbf{u}_t, \quad (34)$$

where $\mathbf{1}$ is the vector whose elements are all ones. The process noise $\mathbf{u}_t$ is drawn from a Gaussian distribution $\mathcal{N}(1, \mathbf{0}, (\sqrt{2}\mathbf{I})^2)$, where $\mathbf{I}$ is the identity matrix. The measurement model is given by a nonlinear function

$$\mathbf{z}_t = \frac{1}{2}\mathbf{x}_t^\top\mathbf{x}_t + \mathbf{v}_t, \quad (35)$$

where $\mathbf{v}_t$ is drawn from a Gaussian distribution $\mathcal{N}(1, \mathbf{0}, \mathbf{I}^2)$. For the estimation of the measurement function, 20 particles (10 particles $\times$ 2 stages) are drawn and the posterior is estimated and propagated through time t $(1 \leq t \leq 50)$.

Fig. 8 presents simulation results by comparing MSEs and variances of both algorithms. The SIR filter shows better or equivalent performance in low dimensions such as

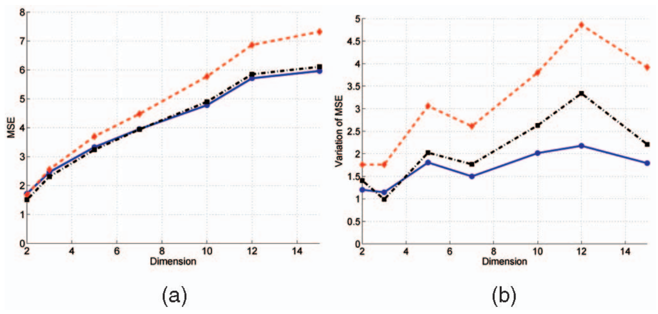

Fig. 9. MSE and variance of for MSE kernel-based Bayesian filtering with 20 samples (blue solid line), SIR filter with 20 particles (red dashed line), and SIR filter with 200 particles (black dotted line) for model 2. (a) Error. (b) Variance.

2D, but our method starts to outperform it in dimensions higher than 3D.

The second process model is a simple linear model given by

$$\mathbf{x}_t = \frac{\mathbf{x}_{t-1}}{2} + 2\cos(2(t-1))\mathbf{1} + \mathbf{u}_t, \tag{36}$$

where $\mathbf{u}_t \sim \mathcal{N}(1, \mathbf{0}, (\sqrt{2}\mathbf{I})^2)$. The same observation model as in (35) is employed and 20 samples are drawn for every simulation.

KBF yields smaller errors in high dimensions as in the previous case and the detailed results are presented in Fig. 9.

The two different process models produce similar results, and KBF shows better performance in high-dimensional cases, as expected. In order to demonstrate the benefit of kernel-based particles, we ran the SIR algorithm with 200 samples and compare the performance with KBF with 20 samples. Surprisingly, the MSEs in the two cases are almost the same, and our algorithm has a smaller variance of MSE than the SIR algorithm.

This result suggests that KBF can be applied effectively to high-dimensional applications, especially when many samples are not available and the observation process is very time consuming.

6 Visual Tracking

Particle filtering provides a convenient method for estimating and propagating the density of state variables regardless of the underlying distribution and the given system in the Bayesian framework. Additionally, our KBF has an advantage of managing multimodal density functions with a relatively small number of samples. In this section, we demonstrate the performance of the KBF by tracking objects in real videos.

The overall tracking procedure is equivalent to what is described in Section 5 and we explain the process and the measurement models briefly.

A random walk is assumed for the process model since, for real moving objects, it is very difficult to describe the motion before observation, even though our algorithm can accommodate the general nonlinear function by unscented transformation, as described in Section 2.2. So, the process model equation in (1) can be rewritten as follows:

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{v}_t, \tag{37}$$

where $\mathbf{v}_t$ is a zero-mean Gaussian random variable.

The likelihood is based on the similarity of the RGB histogram between the target and the candidates. Suppose that the histogram of the target is denoted by $c^\star(i)$ $(i = 1 \dots N)$, where N is the number of bins in the histogram and $\sum_{i=1}^{N} c^\star(i) = 1$. The Bhattacharyya distance in (38) is used to measure the similarity between two histograms

$$D[c^\star, c(\mathbf{x}_t)] = \left(1 - \sum_{i=1}^{N} \sqrt{c^\star(i)c(\mathbf{x}_t; i)}\right) \tag{38}$$

and the measurement function at time t is given by

$$p(\mathbf{z}_t|\mathbf{x}_t) \propto \exp\big(-\gamma D^2[c^\star, c(\mathbf{x}_t)]\big), \tag{39}$$

where γ is a constant.

Multistage sampling is incorporated as introduced in Section 2.3 and the likelihood of each particle is computed by the inverse exponentiation of the Bhattacharyya distance between the target and the candidate histograms, as suggested in [26]. Based on the likelihood of each particle and the initial covariance matrix derived from the distance to the kth nearest neighbor, the measurement density is constructed by density interpolation.

Three sequences are tested in our experiment. In the first sequence, two objects—a hand carrying a can—are tracked with 50 samples (25 samples $\times$ 2 stages). The state space is described by a 10D vector, which is the concatenation of two 5D vectors representing two independent ellipses as follows:

$$(x_1, y_1, lx_1, ly_1, r_1, x_2, y_2, lx_1, ly_2, r_2),$$

where x_i and y_i $(i = 1, 2)$ are the location of ellipses, lx_i is the length of the x-axis, ly_i is the length of the y-axis, and r_i is the rotation variable. The tracking performance for three different algorithms are tested—kernel-based Bayesian filtering, SIR particle filter, and sequential Gaussian mixture filtering with the fixed number of components.[3] The tracking results are presented in Fig. 10; our algorithm successfully tracks two objects, while the SIR particle filter and the five-component sequential Gaussian mixture filtering show relatively unstable performance with the same number of samples.

The bodies of two people are tracked in the second sequence in which one occludes the other completely several times. The state vector is constructed by the same method as in the *can* sequence, but two rectangles are used instead of ellipses. An 8D vector—(x, y, w, h) for each rectangle—is used to describe the state, and 50 samples (25 samples $\times$ 2 stages) are used. Fig. 11a demonstrates the tracking results; our algorithm shows good performance in spite of severe occlusions. The trackers based on the other two algorithms are compared with our algorithm. As seen in Fig. 11b, the performance of the SIR particle filter and the five-component sequential Gaussian mixture filter are worse than our method in this sequence.

3. In the update step, five Gaussian components with the highest weights are propagated to the next time step instead of using the density approximation technique.

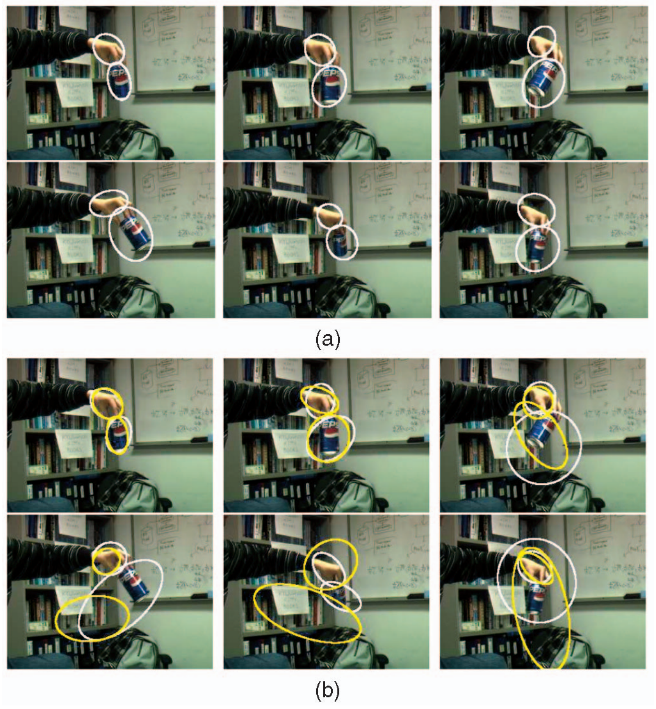

Fig. 10. Object tracking comparison among kernel-based Bayesian filtering, the SIR particle filter, the five-component sequential Gaussian mixture filtering with *can* sequence at $t =$ 1, 24, 41, 96, 152, 200. (a) Result by our method. (b) Result by the SIR filter (white) and the sequential Gaussian mixture filtering (yellow).

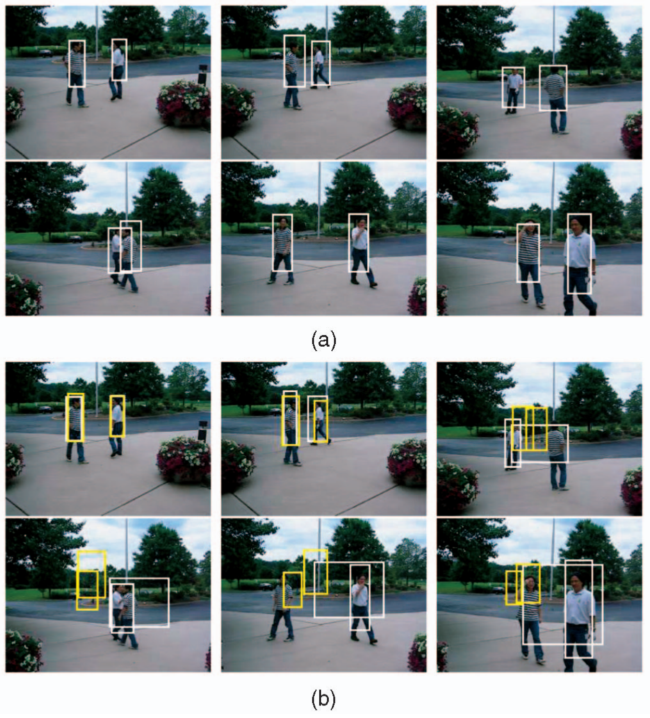

Fig. 11. Object tracking comparison among kernel-based Bayesian filtering, the SIR particle filter, the five-component sequential Gaussian mixture filtering with *person* sequence at $t =$ 1, 94, 140, 192, 236, 300. (a) Result by our method. (b) Result by the SIR filter (white) and the sequential Gaussian mixture filtering (yellow).

The last sequence is more challenging since it involves significant occlusion, clutter, and compression noises. There are many faces in the scene and some of them are very close to the target face, which makes tracking difficult. Tracking is performed in 3D space (location and scale) and 50 samples are used. The SIR filter failed after the target face almost overlapped with another face, but our method recovered from a short-term failure.

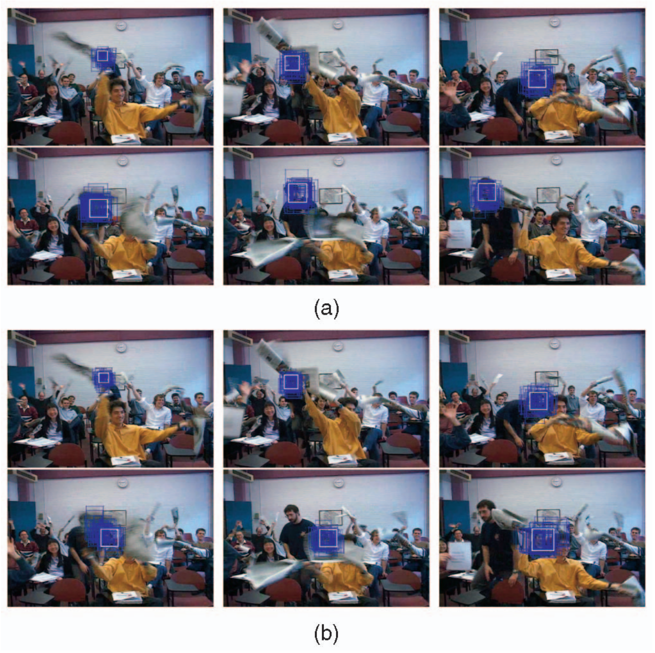

Fig. 12. Object tracking comparison between KBF and conventional particle filter with the *classroom* sequence at $t =$ 1, 42, 97, 104, 113, 153. Particle locations are illustrated as blue rectangles. (a) Result by our method. (b) Result by the SIR filter.

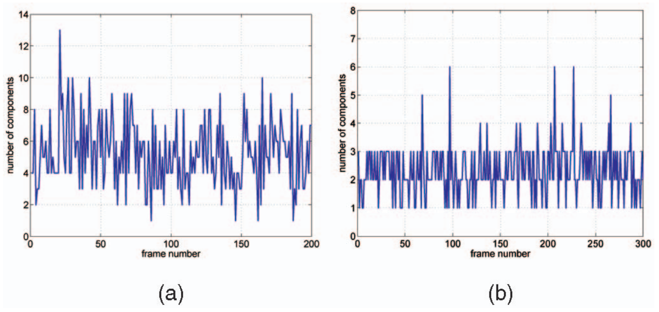

Fig. 13. The number of components in the posterior at each time step. (a) *Can* sequence. (b) *Person* sequence.

The preservation of multimodality in the posterior, which is the advantage of our method over the conventional particle filter (Fig. 12), is crucial for the overall tracker performance since multiple hypotheses can be handled effectively. We illustrated the number of components in the posterior at each time step in Fig. 13; the number of components changes significantly over time, especially in the *can* sequence. These results suggest that our method is potentially more robust to exceptional cases such as occlusions and clutter by effectively modeling a time-varying number of multiple hypotheses. Fig. 14 demonstrates sample frames in the *can* sequence which involve complex posterior density functions; each mode in the density function with 0.1 or higher weight is illustrated as a green and blue ellipse, where the intensity of the ellipse is proportional to the weight of corresponding mode. Potential reasons for highly multimodal posteriors are severe

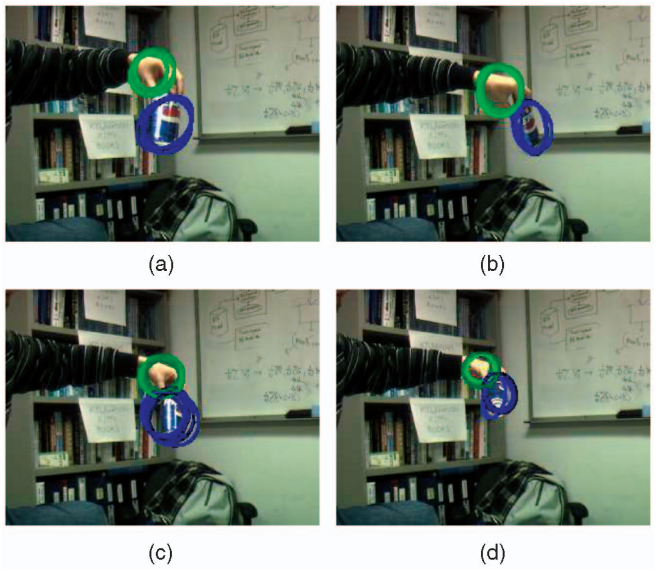

Fig. 14. Sample frames in the *can* sequence with many components in the posterior. (a) $t = 30$, 10 components. (b) $t = 135$, nine components. (c) $t = 165$, ten components. (d) $t = 186$, nine components.

appearance changes due to reflections ((a) and (c)) and significant orientation changes ((b) and (d)). On the other hand, the highly multimodal posterior in the *person* and *classroom* sequence is typically observed after occlusions, clutter, large movements of the camera and/or target, and so on.

Kernel-based Bayesian filtering does have additional computational overhead, so it is worthwhile to compare the performance of both algorithms when the same amount of computational resources is used for tracking. According to our experiments, one would need to run the SIR algorithm using 150-200 particles to obtain comparable results with our algorithm using 50 samples for all three sequences. The running time of the SIR particle filter with 100-150 samples is equivalent to our algorithm with 50 samples. Of course, the relative computation time of our algorithm compared to the simple SIR particle filter depends on the complexity of the measurement process. Tracking based on kernel-based Baysian filtering using the histogram of multiple elliptical areas, as in the *can* sequence, is more advantageous than tracking with the observation from a simple rectangular region, as in the *classroom* sequence. The detailed analysis of the relative performance for all sequences is presented in Table 2.

TABLE 2
Relative Performance of KBF and SIR

Sequence name	Number of particles for equivalent accuracy	Number of particles for equivalent computations cost
can	150	100
person	200	150
class	150	150

*All numbers are approximate.

7 Discussion and Conclusion

We have described a novel sequential Bayesian filtering framework—kernel-based Baysian filtering—where analytic representations are used to approximate relevant density functions. Density interpolation and approximation techniques based on a mixture of Gaussians were introduced for density representation and propagation and a fusion-based state estimation method was also presented. By maintaining analytic representations of the density functions, we can sample in the state space more effectively and more efficiently. This advantage of the proposed method is most significant for high-dimensional problems. The kernel-based Baysian filtering was applied to the visual tracking problem and the effectiveness of the technique was demonstrated through various simulations and tests on real video sequences. Our future work is focused on analyzing the approximation error in the posterior distribution and its propagation over time.

References

[1] I. Abramson, "On Bandwidth Variation in Kernel Estimates—A Square Root Law," *Annals of Statistics,* vol. 10, no. 4, pp. 1217-1223, 1982.

[2] M. Adlers, "Topics in Sparse Least Squares Problems," PhD dissertation, Linköpings Universitet, Sweden, http://www.math.liu.se/~milun/thesis, 2000.

[3] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for On-Line Non-Linear/Non-Gaussian Bayesian Tracking," *IEEE Trans. Signal Processing,* vol. 50, no. 2, pp. 174-189, 2002.

[4] J. Cantarella and M. Piatek, *tsnnls: A Solver for Large Sparse Least Squares Problem with Non-Negative Variables,* preprint, http://www.cs.duq.edu/~piatek/tsnnls/, 2004.

[5] T. Cham and J. Rehg, "A Multiple Hypothesis Approach to Figure Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 219-239, 1999.

[6] C. Cheng, R. Ansari, and A. Khokhar, "Multiple Object Tracking with Kernel Particle Filter," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[7] W. Cleveland, "Robust Locally Weighted Regression and Smoothing Scatterplots," *J. Am. Statistical Assoc.,* vol. 74, pp. 829-836, 1979.

[8] W. Cleveland and C. Loader, "Smoothing by Local Regression: Principles and Methods," *Statistical Theory and Computational Aspects of Smoothing,* pp. 10-49, 1996.

[9] D. Comaniciu, "Nonparametric Information Fusion for Motion Estimation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 59-66, 2003.

[10] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 603-619, May 2002.

[11] D. Comaniciu, V. Ramesh, and P. Meer, "The Variable Bandwidth Mean Shift and Data-Driven Scale Selection," *Proc. Eighth Int'l Conf. Computer Vision,* vol. 1, pp. 438-445, July 2001.

[12] J. Deutscher, A. Blake, and I. Reid, "Articulated Body Motion Capture by Annealed Particle Filtering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2000.

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

[14] A. Doucet, S. Godsill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing,* vol. 10, no. 3, pp. 197-208, 2000.

[15] B. Han, D. Comaniciu, Y. Zhu, and L.S. Davis, "Sequential Kernel Density Approximation and Its Applications to Real-Time Visual Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 7, pp. 1186-1197, July 2008.

[16] B. Han, D. Comaniciu, Y. Zhu, and L. Davis, "Incremental Density Approximation and Kernel-Based Bayesian Filtering for Object Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2004.

[17] B. Han, Y. Zhu, D. Comaniciu, and L. Davis, "Kernel-Based Bayesian Filtering for Object Tracking," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2005.

[18] M. Isard and A. Blake, "Condensation—Conditional Density Propagation for Visual Tracking," *Int'l J. Computer Vision,* vol. 29, no. 1, 1998.

[19] S. Julier and J. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," *Proc. SPIE,* vol. 3068, pp. 182-193, 1997.

[20] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans. Am. Soc. Mechanical Eng. D: J. Basic Eng.*, vol. 82, pp. 35-45, 1960.

[21] J. Kotecha and P. Djuric, "Gaussian Sum Particle Filtering," *IEEE Trans. Signal Processing,* vol. 51, no. 10, pp. 2602-2612, 2003.

[22] C.L. Lauwon and B.J. Hanson, *Solving Least Squares Problems.* Prentice-Hall, 1974.

[23] J. MacCormick and M. Isard, "Partitioned Sampling, Articulated Objects, and Interface-Quality Hand Tracking," *Proc. European Conf. Computer Vision,* pp. 3-19, 2000.

[24] R. Merwe, A. Doucet, N. Freitas, and E. Wan, "The Unscented Particle Filter," Technical Report CUED/F-INFENG/TR 380, Cambridge Univ. Eng. Dept., 2000.

[25] B. Park and J. Marron, "Comparison of Data-Driven Bandwidth Selectors," *J. Am. Statistical Assoc.,* vol. 85, pp. 66-72, 1990.

[26] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-Based Probabilistic Tracking," *Proc. European Conf. Computer Vision,* vol. 1, pp. 661-675, 2002.

[27] V. Philomin, R. Duraiswami, and L.S. Davis, "Quasi-Random Sampling for Condensation," *Proc. European Conf. Computer Vision,* vol. 2, pp. 134-149, 2000.

[28] T. Poggio and F. Girosi, "A Theory of Networks for Approximation and Learning," technical report, Artificial Intelligence Laboratory, Massachusetts Inst. of Technology, 1989.

[29] Y. Rui and Y. Chen, "Better Proposal Distributions: Object Tracking Using Unscented Particle Filter," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 786-793, 2001.

[30] S. Sheather and M. Jones, "A Reliable Data-Based Bandwidth Selection Method for Kernel Density Estimation," *J. Royal Statistical Soc. B,* vol. 53, pp. 683-690, 1991.

[31] C. Sminchisescu and B. Triggs, "Covariance Scaled Sampling for Monocular 3D Body Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 447-454, 2001.

[32] C. Sminchisescu and B. Triggs, "Hyperdynamics Importance Sampling," *Proc. European Conf. Computer Vision*, pp. 769-783, 2002.

[33] J. Sullivan and J. Rittscher, "Guiding Random Particles by Deterministic Search," *Proc. Eighth Int'l Conf. Computer Vision,* vol. 1, pp. 323-330, 2001.

[34] P. Torma and C. Szepesvari, "Enhancing Particle Filter Using Local Likelihood Sampling," *Proc. European Conf. Computer Vision,* pp. 16-27, 2004.

[35] J. Vermaak, A. Doucet, and P. Perez, "Maintaining Multi-Modality through Mixture Tracking," *Proc. Ninth Int'l Conf. Computer Vision,* vol. 1, 2003.

[36] E.A. Wan and R. van der Merwe, "The Unscented Kalman Filter for Non-Linear Estimation," *Proc. Symp. 2001 Adaptive Systems for Signal Proc. Comm. and Control,* 2000.

[37]

Bohyung Han received the BS and MS degrees from Seoul National University, Korea, in 1997 and 2000, respectively, and the PhD degree from the University of Maryland, College Park, in 2005. He was a senior research engineer at the Samsung Electronics R&D Center, Irvine, California, and a research scientist at the University of California, Irvine. He is currently with the Advanced Project Center, Mobileye Vision Technologies, Princeton, New Jersey. He is also a visiting scholar in the Computer Vision Laboratory at Princeton University. His research interests include statistical analysis in computer vision, visual tracking, object detection and recognition, sensor fusion, machine learning, image processing, and multimedia. He is a member of the IEEE.

Ying Zhu received the PhD degree from Princeton University in 2003. Since February 2003, she has been a member of the technical staff in the Real-Time Vision and Modeling Department at Siemens Corporate Research Inc., Princeton, New Jersey. Since October 2004, she has been a project manager in the Statistical Method in Vision Program and has been leading the research and development effort on Advanced Driver Assistance Systems. Her research focuses on statistical learning, pattern classification, statistical estimation, and Bayesian inference methods for real-time object detection, tracking, and recognition. She served on the program committee of the Second Workshop on Statistical Methods in Video Processing May 2004 and on the organizing committee of the International Workshop on Machine Vision for Intelligent Vehicles June 2005. She is a member of the IEEE.

Dorin Comaniciu received PhD degrees in electrical engineering from the Polytechnic University of Bucharest in 1995 and from Rutgers University in 1999. Since 1999, he has been with Siemens Corporate Research, Princeton, New Jersey, first as a member of the technical staff and then as a senior member of the technical staff and manager of the Statistical Methods for Vision Systems Program. He is currently the head of the Integrated Data Systems Department. His research interests include robust methods for computer vision, motion estimation, nonparametric analysis, robust information fusion, medical imaging, biomedical informatics, content-based access to visual data, and integrated information modeling. He holds 18 US patents and has coauthored more than 120 papers, conference papers, and book chapters in the area of visual information processing. For his work in object tracking, he has received the Best Paper Award at the 2000 IEEE Conference on Computer Vision and Pattern Recognition. For his innovations in the areas of medical imaging and intelligent vehicles, he has received the 2004 Siemens Inventor of the Year Award, the highest technical recognition of Siemens AG worldwide. He serves as an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and was an associate editor for *Pattern Analysis and Applications* between 2002 and 2004. He leads the scientific direction of one of the largest European projects in biomedical informatics, Health-e-Child. He is a senior member of the IEEE.

Larry S. Davis received the BA degree from Colgate University in 1970 and the MS and PhD degrees in computer science from the University of Maryland in 1974 and 1976, respectively. From 1977 to 1981, he was an assistant professor in the Department of Computer Science at the University of Texas, Austin. He returned to the University of Maryland as an associate professor in 1981. From 1985 to 1994, he was the director of the University of Maryland Institute for Advanced Computer Studies. He is currently a professor in the institute and in the Computer Science Department, as well as the chair of the Computer Science Department. He was named a fellow of the IEEE in 1997. He is known for his research in computer vision and high-performance computing. He has published more than 100 papers in journals and has supervised more than 20 PhD students. He is an associate editor of the *International Journal of Computer Vision* and an area editor for *Computer Models for Image Processing: Image Understanding*. He has served as the program or general chair for most of the field's major conferences and workshops, including the Fifth International Conference on Computer Vision, the 2004 Computer Vision and Pattern Recognition Conference, and the 11th International Conference on Computer Vision. He is a fellow of the IEEE.