

# Incremental Density Approximation and Kernel-Based Bayesian Filtering for Object Tracking

Bohyung Han<sup>†</sup>

<sup>†</sup> Dept. of Computer Science

University of Maryland

College Park, MD 20742, USA

{bhhan, lsd}@cs.umd.edu

Dorin Comaniciu<sup>‡</sup>

<sup>‡</sup> Real-Time Vision and Modeling Dept.

Siemens Corporate Research

Princeton, NJ 08540, USA

{Dorin.Comaniciu, Ying.Zhu}@siemens.com

Ying Zhu<sup>‡</sup>

Larry Davis<sup>†</sup>

## Abstract

*Statistical density estimation techniques are used in many computer vision applications such as object tracking, background subtraction, motion estimation and segmentation. The particle filter (Condensation) algorithm provides a general framework for estimating the probability density functions (pdf) of general non-linear and non-Gaussian systems. However, since this algorithm is based on a Monte Carlo approach, where the density is represented by a set of random samples, the number of samples is problematic, especially for high dimensional problems. In this paper, we propose an alternative to the classical particle filter in which the underlying pdf is represented with a semi-parametric method based on a mode finding algorithm using mean-shift. A mode propagation technique is designed for this new representation for tracking applications. A quasi-random sampling method [14] in the measurement stage is used to improve performance, and sequential density approximation for the measurements distribution is performed for efficient computation. We apply our algorithm to a high dimensional color-based tracking problem, and demonstrate its performance by showing competitive results with other trackers.*

## 1 Introduction

Many visual features such as intensity, color, gradient, texture or motion are commonly modeled using density estimation. Object tracking, background subtraction, segmentation, and motion estimation are typical examples that involve statistical estimation and propagation of the underlying density.

Real-time object tracking is a challenging computer vision task. Tracking based on the mean-shift algorithm [5] searches for the local maximum of the object appearance model. However, because it is a deterministic algorithm, it generally cannot recover from a failure. This problem can be ameliorated by probabilistic trackers using the Kalman filter and its extensions [16, 17, 18], or more generally particle filters [8, 9, 12, 13, 15] that achieve robustness to clutter and occlusion by maintaining multiple hypotheses in the state

space.

Particle filtering provides a convenient framework for estimating and propagating the density of state variables regardless of the underlying distribution and the given system. The particle filter can manage multi-modal density functions effectively. Because the sampling must be sufficient to capture the variations in the state space, a very large number of samples is often necessary to guarantee sufficient accuracy.

There have been many parametric density representations proposed for tracking. In [11, 16], the authors suggest Gaussian mixture models, but their method requires knowledge of the number of components, which is difficult to know in advance. Additionally, it is not appropriate if there are a large number of modes in the underlying pdf or the number of modes changes frequently. A more elaborate target model is described in [10], where a 3-component mixture for the stable process, the outlier data and the wandering term is designed to capture rapid temporal variations in the model. Cham and Rehg [2] introduce a piecewise Gaussian representation 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. Kernel density estimation [7] is a widely used non-parametric approach in computer vision. Its major advantage is the flexibility to represent very complicated densities effectively. But its very high memory requirements and computational complexity inhibit the use of this method.

This paper introduces a density approximation methodology that is an alternative to kernel density estimation, but computationally as simple as parametric methods. It is based on the mode finding algorithm [4, 6] by variable-bandwidth mean-shift. The density is represented with a weighted sum of Gaussians, whose number, weights, means and covariances are automatically determined. Instead of a batch implementation, we describe a much more efficient incremental density approximation method.

We next discuss how this density approximation technique is incorporated into the particle filter framework. Quasi-

random sampling [14] and kernel-based particles contribute to decrease the required number of samples, allowing us to address higher dimensional problems. The new kernel-based particle filter algorithm is applied to video tracking, and its performance is compared with the classical particle filter.

This paper is organized as follows. Section 2 describes our mode detection and density approximation method. Section 3 introduces the new mode propagation techniques in the particle filter framework, and section 4 presents experiments for object tracking in video.

## 2 Mode Detection and Density Approximation

In this section, we present the iterative procedure for mode detection based on the variable-bandwidth mean-shift [6], and the batch density approximation using the mode detection technique. Then, an efficient alternative method – incremental approximation – is presented.

### 2.1 Batch Density Approximation

Denote by  $\mathbf{x}_i$  ( $i = 1 \dots n$ ) a set of means of Gaussians in  $R^d$  and by  $\mathbf{P}_i$  a symmetric positive definite  $d \times d$  covariance matrix associated with the corresponding Gaussian. Let each Gaussian have a weight  $\kappa_i$  with  $\sum_{i=1}^n \kappa_i = 1$ . The sample point density estimator computed at point  $\mathbf{x}$  is given by

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

where

$$D^2(\mathbf{x}, \mathbf{x}_i, \mathbf{P}_i) \equiv (\mathbf{x} - \mathbf{x}_i)^\top \mathbf{P}_i^{-1} (\mathbf{x} - \mathbf{x}_i) \quad (2)$$

is the Mahalanobis distance from  $\mathbf{x}$  to  $\mathbf{x}_i$ . As one can see, the density at  $\mathbf{x}$  is obtained as the *average of Gaussian densities* centered at each data point  $\mathbf{x}_i$  and having the covariance  $\mathbf{P}_i$ .

The variable-bandwidth mean-shift vector is defined by

$$\begin{aligned} \mathbf{m}(\mathbf{x}) &\equiv \mathbf{P}_h(\mathbf{x}) \sum_{i=1}^n \omega_i(\mathbf{x}) \mathbf{P}_i^{-1} \mathbf{x}_i - \mathbf{x} \\ &= \left( \sum_{i=1}^n \omega_i(\mathbf{x}) \mathbf{P}_i^{-1} \right)^{-1} \left( \sum_{i=1}^n \omega_i(\mathbf{x}) \mathbf{P}_i^{-1} \mathbf{x}_i \right) - \mathbf{x} \end{aligned} \quad (3)$$

where

$$\mathbf{P}_h^{-1}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) \mathbf{P}_i^{-1} \quad (4)$$

and the weights

$$w_i(\mathbf{x}) = \frac{\kappa_i |\mathbf{P}_i|^{-1/2} \exp\left(-\frac{1}{2}D^2(\mathbf{x}, \mathbf{x}_i, \mathbf{P}_i)\right)}{\sum_{i=1}^n \kappa_i |\mathbf{P}_i|^{-1/2} \exp\left(-\frac{1}{2}D^2(\mathbf{x}, \mathbf{x}_i, \mathbf{P}_i)\right)} \quad (5)$$

satisfy  $\sum_{i=1}^n w_i(\mathbf{x}) = 1$ .

It can be shown that by iteratively computing the mean-shift vector (3) and translating the location  $\mathbf{x}$  by  $\mathbf{m}(\mathbf{x})$ , a mode seeking algorithm is obtained which converges to a stationary point of the density (1). Since the maxima of the density are the only stable points of the iterative procedure, the convergence usually occurs at a mode of the underlying density. A formal check for the maximum involves the computation of the Hessian matrix

$$\begin{aligned} \hat{\mathbf{H}}(\mathbf{x}) &= (2\pi)^{-d/2} \sum_{i=1}^n \kappa_i |\mathbf{P}_i|^{-1/2} \exp\left(-\frac{1}{2}D^2(\mathbf{x}, \mathbf{x}_i, \mathbf{P}_i)\right) \times \\ &\quad \mathbf{P}_i^{-1} \left( (\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^\top - \mathbf{P}_i \right) \mathbf{P}_i^{-1} \end{aligned} \quad (6)$$

which should be negative definite (having all eigenvalues negative).

Suppose that the approximate density has  $n'$  unique modes of  $\tilde{\mathbf{x}}_j$  ( $j = 1 \dots n'$ ) with associated weight  $\tilde{\kappa}_j$  and covariance  $\tilde{\mathbf{P}}_j$  after the mode finding procedure. Since  $\tilde{\mathbf{P}}_j$  might be quite different from the actual covariance in the underlying distribution, the Hessian matrix  $\hat{\mathbf{H}}_j$  of each mode is used for the computation of  $\tilde{\mathbf{P}}_j$

$$\tilde{\mathbf{P}}_j = \frac{\tilde{\kappa}_j^{\frac{2}{d+2}}}{|2\pi(-\hat{\mathbf{H}}_j^{-1})|^{\frac{d+2}{4}}} (-\hat{\mathbf{H}}_j^{-1}) \quad (7)$$

The basic idea of equation (7) is to fit the covariance using the curvature in the neighborhood of the mode.

The final density approximation is then given by

$$\tilde{f}(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n'} \frac{\tilde{\kappa}_i}{|\tilde{\mathbf{P}}_i|^{1/2}} \exp\left(-\frac{1}{2}D^2(\mathbf{x}, \tilde{\mathbf{x}}_i, \tilde{\mathbf{P}}_i)\right) \quad (8)$$

and  $n' \ll n$  is satisfied in most cases.

### 2.2 Incremental Density Approximation

The density approximation technique described in section 2.1 is accurate and memory efficient, but computationally expensive because the mode detection procedure for  $n$  components requires  $O(n^2)$  time. Moreover, for each sample point, a large number of mean-shift iterations might be required to converge. To overcome this computational costs, we suggest an alternative method, an incremental density approximation, described below.

Usually, a large number of samples are required to estimate the density correctly, but there are only several modes in the underlying density function. The incremental approximation algorithm is an empirical solution exploiting this fact. Suppose  $n$  samples are to be used for the density approximation. We will process samples, "one at a time." If a kernel associated with each sample can be merged incrementally with

others in the same mode, then the time to compute the mean-shift vector will be decreased dramatically.

The algorithm proceeds as follows. When the Gaussian kernel for the next sample is added to the current density function, the density will be updated by the variable-bandwidth mean-shift. For example, if the component for the  $k$ th sample is added to the current density function  $\hat{f}_{(k-1)}$ , the density after the insertion is given by

$$\hat{f}_k(\mathbf{x}) = \frac{\kappa_k}{K_k} \frac{1}{(2\pi)^{d/2} |\mathbf{P}_k|^{1/2}} \exp\left(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}_k, \mathbf{P}_k)\right) + \left(1 - \frac{\kappa_k}{K_k}\right) \sum_{i=1}^{n_{k-1}} \frac{\bar{\kappa}_i}{(2\pi)^{d/2} |\bar{\mathbf{P}}_i|^{1/2}} \exp\left(-\frac{1}{2} D^2(\mathbf{x}, \bar{\mathbf{x}}_i, \bar{\mathbf{P}}_i)\right) \quad (9)$$

where  $K_k = \sum_{i=1}^k \kappa_i$ ,  $\bar{\kappa}_i$  and  $\bar{\mathbf{P}}_i$  are the weight and the covariance associated with  $\bar{\mathbf{x}}_i$  in the current density respectively, and  $n_{k-1}$  is the number of modes after the  $(k-1)$ th component insertion. In each step, the mode detection procedure and covariance computation need to be applied, and the new density function  $\hat{f}_k(\mathbf{x})$  is estimated. After  $n$  steps, the weight of each sample is adjusted to its original weight, and the incremental density approximation can be obtained.

During the incremental procedure, two or more modes which are close to each other in the underlying density may be merged, and some of them may be lost by the final iteration. This situation should be avoided since it increases the approximation error. We avoid this problem by using a 2-stage algorithm. In the first stage, the incremental density approximation technique is used with a small bandwidth. This may result in several *spurious* modes which do not exist in the underlying density. After the final step, let each component in the approximate density be  $N(\bar{\kappa}_i, \bar{\mathbf{x}}_i, \bar{\mathbf{P}}_i)$  ( $i = 1 \dots n_n$ ) where  $N(\cdot)$  is a Gaussian distribution having a (*weight, mean, covariance*) triple. In the second stage, the batch density approximation algorithm described in 2.1 is performed with the  $\bar{\mathbf{x}}_i$ 's as starting points. The correct mode locations and their covariance matrices can be computed accurately in the second stage.

The 2-stage incremental algorithm is very efficient since the intermediate and the final density function in the first stage have a small number of modes ( $n_k \ll n$ ).

### 2.3 Performance of Approximation

The accuracy of these approximations is demonstrated in Figure 1. From a one-dimensional distribution composed of five weighted Gaussians, 200 samples are drawn. Figure 1(a) shows the result of kernel density estimation. The results of batch approximation with variable-bandwidth mean-shift are presented in Figure 1(b). The incremental approximation is presented in Figure 1(c) and the number of modes in each incremental step is shown in Figure 1(d).

Table 1 compares accuracy and speed of the approximations. Three different cases are tested 20 times each, and

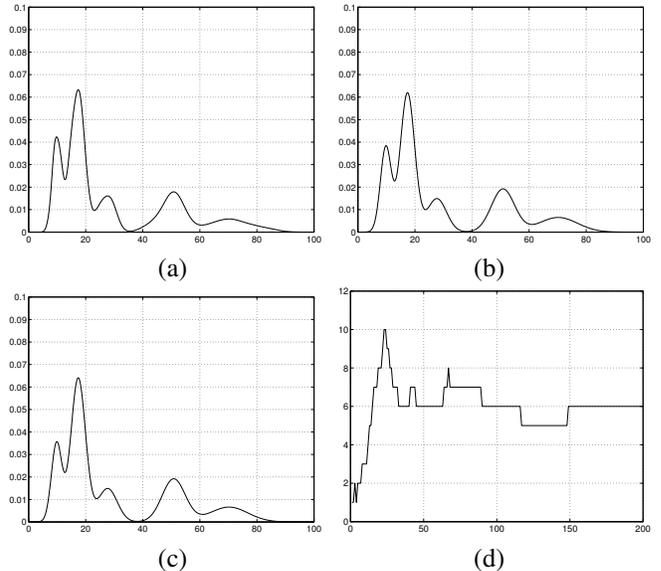


Figure 1: Comparisons between the kernel density estimation and its approximations (1D). For the approximation, 200 samples are drawn from the original distribution –  $N(0.2, 10, 2)$ ,  $N(0.35, 17, 4)$ ,  $N(0.15, 27, 8)$ ,  $N(0.2, 50, 16)$ , and  $N(0.1, 71, 32)$ . (a) kernel density estimation (b) batch approximation (c) incremental approximation (d) number of modes in each incremental step

Mean Integrated Squared Error (MISE) and execution time speedups are calculated. Denote by  $E_{kde}$  the error between the kernel density estimation and the original distribution, and by  $E_{bat}$  ( $E_{inc}$ ) the error between the batch (incremental) approximation and the kernel density estimation. Both density approximations produce small errors comparable to kernel density estimation, and the incremental approximation is much faster with errors comparable to the batch approximation.

Figure 2 shows that both approximation methods are accurate enough to replace kernel density estimation in the multi-dimensional case. In 2D, the incremental approximation also has comparable accuracy to the batch approximation, but it is practically much faster.

## 3 Mode Propagation through Bayesian Filtering

In this section we will show how to use the approximation technique to propagate the density modes in the particle filter framework.

The particle filter [8] is a stochastic framework to propagate the conditional density; it originated from statistics and control theory. The algorithm combines the dynamical models and measurement by sampling to propagate an entire probability distribution for the state over time.

We next explain how the semi-parametric density repre-

Table 1: Performance comparison between batch and incremental approximation

case	MISE ( $\times 10^{-5}$ )			speedup (batch/incremental)
	$E_{kde}$	$E_{bat}$	$E_{inc}$	
1	5.0772	1.4512	3.1007	8.3502
2	2.2909	0.5323	1.2463	7.0119
3	1.0138	0.6900	1.7869	6.2597

- case 1:  $N(0.2, 10, 2)$ ,  $N(0.35, 17, 4)$ ,  $N(0.15, 27, 8)$ ,  $N(0.2, 50, 16)$ ,  $N(0.1, 71, 32)$   
- case 2:  $N(0.15, 12, 5)$ ,  $N(0.1, 50, 4)$ ,  $N(0.35, 70, 8)$ ,  $N(0.25, 90, 16)$ ,  $N(0.15, 119, 32)$   
- case 3:  $N(0.15, 25, 10)$ ,  $N(0.1, 37, 8)$ ,  $N(0.15, 65, 16)$ ,  $N(0.25, 77, 9)$ ,  $N(0.15, 91, 30)$ ,  $N(0.2, 154, 15)$

sensation is incorporated into the particle filter, and how to propagate the density through Bayesian filtering based on variable-bandwidth mean-shift [6].

### 3.1 Bayesian Filtering

The state variable  $\mathbf{x}_t$  ( $t = 0 \dots n$ ) is characterized by its probability density function estimated from the sequence of measurements  $\mathbf{z}_t$  ( $t = 0 \dots n$ ).

The process and measurement model are given by

$$\mathbf{x}_t = g(\mathbf{x}_{t-1}, \mathbf{u}_t) \quad (10)$$

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{v}_t) \quad (11)$$

where  $\mathbf{u}_t$  and  $\mathbf{v}_t$  are the process and a measurement noise, respectively.

The conditional density of the state variable given the measurements is propagated through *prediction* and *update* stages by a Bayesian framework.

$$p(\mathbf{x}_t | \mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1} \quad (12)$$

$$p(\mathbf{x}_t | \mathbf{z}_t) = \frac{1}{k} p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}) \quad (13)$$

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  $\mathbf{x}_t$ . The posterior probability at time step  $t$ ,  $p(\mathbf{x}_t | \mathbf{z}_t)$ , is used as a prior in the next step.

### 3.2 Prediction

Suppose that the prior of the state variable  $\mathbf{x}$  under the measurement variable  $\mathbf{z}$  at some time step is represented by a weighted mixture of Gaussians. Our goal is to retain this representation through the prediction and update stages, and to represent the posterior probability in the next step with the same form.

Denote by  $\mathbf{x}_t^i$  ( $i = 1 \dots n_t$ ) a set of means in  $R^d$  and by  $\mathbf{P}_t^i$  the corresponding covariance matrices at time step  $t$ . Let

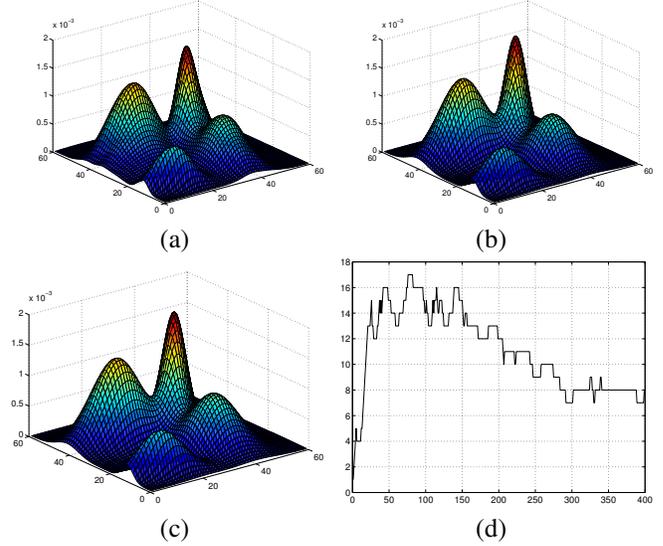


Figure 2: Comparison between the kernel density estimation and its approximations (2D). The incremental approximation is about 11 times faster than the batch approximation when 400 samples are drawn. (a) kernel density estimation (b) batch approximation (MISE =  $1.4889 \times 10^{-8}$ ) (c) incremental approximation (MISE =  $1.2337 \times 10^{-8}$ ) (d) number of modes in each incremental step

each Gaussian have a weight  $\kappa_t^i$  with  $\sum_{i=1}^{n_t} \kappa_t^i = 1$ , and let the prior density function be given by

$$p(\mathbf{x}_{t-1} | \mathbf{z}_{t-1}) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n_{t-1}} \frac{\kappa_{t-1}^i}{|\mathbf{P}_{t-1}^i|^{1/2}} \exp\left(-\frac{1}{2} D^2(\mathbf{x}_{t-1}, \mathbf{x}_{t-1}^i, \mathbf{P}_{t-1}^i)\right) \quad (14)$$

Suppose the motion model is assumed to be a linear function  $g$  with Gaussian noise of covariance  $\mathbf{Q}$ . We actually use a zero-order function in our tracker because it is ordinarily difficult to identify a correct dynamic model. The predicted density function is then also a mixture of Gaussians

$$p(\mathbf{x}_t | \mathbf{z}_{t-1}) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n_{t-1}} \frac{\bar{\kappa}_t^i}{|\bar{\mathbf{P}}_t^i|^{1/2}} \exp\left(-\frac{1}{2} D^2(\mathbf{x}_t, \bar{\mathbf{x}}_t^i, \bar{\mathbf{P}}_t^i)\right) \quad (15)$$

where  $\bar{\kappa}_t^i$ ,  $\bar{\mathbf{x}}_t^i$  and  $\bar{\mathbf{P}}_t^i$  are obtained from the modes of  $p(\mathbf{x}_{t-1} | \mathbf{z}_{t-1})$  and the linear process equation (10).

### 3.3 Sampling

Sampling determines the speed and the accuracy of the particle filter since it directly affects the posterior probability distribution. Instead of using the predicted density function in equation (15) as the proposal distribution, we employ the quasi-random sampling method [14] and the final proposal

distribution is

$$q(\mathbf{x}_t | \mathbf{z}_t) = \frac{\alpha}{(2\pi)^{d/2}} \sum_{i=1}^{n_{t-1}} \frac{\bar{\kappa}_t^i}{|\bar{\mathbf{P}}_t^i|^{1/2}} \exp\left(-\frac{1}{2}D^2(\mathbf{x}_t, \bar{\mathbf{x}}_t^i, \bar{\mathbf{P}}_t^i)\right) + (1-\alpha)U(\mathbf{x}_t) \quad (16)$$

where  $U(\mathbf{x}_t)$  is the uniform distribution in  $\mathbf{x}_t$  and  $\alpha$  is the ratio for importance sampling.

### 3.4 Measurement

In the conventional particle filter, the measurement distribution  $p(\mathbf{z}_t | \mathbf{x}_t)$  is completely dependent on the weight of each particle. This representation for the density results in the depletion of samples, and requires a lot of particles for accurate estimation. Here, we explain how to parameterize the measurement density with a mixture of Gaussians so that the posterior density is also represented with a mixture of Gaussians. If the measurement for each particle is assumed to be a Gaussian kernel, the measurement density can be represented by kernel density estimation. The kernel-based particle has the advantage of allowing us to compute the density of all points in the continuous space. This is a nice property especially for high dimensional problems because the number of samples required for accurate estimation is smaller than the classical particle filter algorithm. However, kernel density estimation is slow and memory inefficient, and is not applicable to real-time applications.

In order to avoid the inefficiency of kernel density estimation, the density approximation technique introduced in section 2 is used. In short, the mean-shift vector is computed for each sample point and moves in the gradient ascent direction until it converges to a local maximum. Then, we can find all the modes that exist in the underlying density, and the covariance matrices using the Hessian. This allows us to decrease the memory requirement to represent the underlying distribution by using only a small number of Gaussians.

Either the batch or the incremental approximation can be used, and the measurement is also a mixture of  $m_t$  Gaussians in the state space at time step  $t$  as

$$p(\mathbf{z}_t | \mathbf{y}_t) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{m_t} \frac{\tau_t^i}{|\mathbf{R}_t^i|^{1/2}} \exp\left(-\frac{1}{2}D^2(\mathbf{y}_t, \mathbf{y}_t^i, \mathbf{R}_t^i)\right) \quad (17)$$

where  $\tau_t^i$  is the weight and  $\mathbf{R}_t^i$  is the covariance associated with the mean  $\mathbf{y}_t^i$  ( $i = 1 \dots m_t$ ). Note that  $\mathbf{y}$  is another state variable for the measurement equation.

### 3.5 Update

Since both the prediction and the measurement functions are composed of a mixture of Gaussians, the posterior can be also represented by a Gaussian mixture which is obtained by the products of the Gaussian pairs between prediction and measurement as seen in equation (13).

The products of two Gaussians,  $N(\eta_1, \mathbf{m}_1, \Sigma_1)$  and  $N(\eta_2, \mathbf{m}_2, \Sigma_2)$ , is also a Gaussian distribution  $N(\eta, \mathbf{m}, \Sigma)$  given by

$$\eta = \eta_1 \eta_2 \quad (18)$$

$$\mathbf{m} = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} (\Sigma_1^{-1} \mathbf{m}_1 + \Sigma_2^{-1} \mathbf{m}_2) \quad (19)$$

$$\Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} \quad (20)$$

Therefore, when the prediction and the measurement have Gaussian components  $N(\bar{\kappa}_t^i, \bar{\mathbf{x}}_t^i, \bar{\mathbf{P}}_t^i)$  ( $i = 1 \dots n_{t-1}$ ) and  $N(\tau_t^j, \mathbf{y}_t^j, \mathbf{R}_t^j)$  ( $j = 1 \dots m_t$ ) respectively, the product of the two distribution is as follows.

$$\left( \sum_{i=1}^{n_{t-1}} N(\bar{\kappa}_t^i, \bar{\mathbf{x}}_t^i, \bar{\mathbf{P}}_t^i) \right) \left( \sum_{j=1}^{m_t} N(\tau_t^j, \mathbf{y}_t^j, \mathbf{R}_t^j) \right) = \sum_{i=1}^{n_{t-1}} \sum_{j=1}^{m_t} N(\bar{\kappa}_t^i \tau_t^j, \mathbf{m}_t^{ij}, \Sigma_t^{ij}) \quad (21)$$

where

$$\mathbf{m}_t^{ij} = \Sigma_t^{ij} ((\bar{\mathbf{P}}_t^i)^{-1} \mathbf{x}_t^i + (\mathbf{R}_t^j)^{-1} \mathbf{y}_t^j) \quad (22)$$

$$\Sigma_t^{ij} = ((\bar{\mathbf{P}}_t^i)^{-1} + (\mathbf{R}_t^j)^{-1})^{-1} \quad (23)$$

The result of applying equation (21) is a weighted Gaussian mixture, but the number of modes in equation (21) can be reduced by the mode detection algorithm. Also, the covariance matrix  $\mathbf{P}_t^i$  for each detected mode location  $\mathbf{x}_t^i$  should be evaluated using the Hessian for accuracy. Therefore, the final posterior distribution is given by

$$p(\mathbf{x}_t | \mathbf{z}_t) = \frac{1}{(2\pi)^{d/2}} \sum_{i=1}^{n_t} \frac{\kappa_t^i}{|\mathbf{P}_t^i|^{1/2}} \exp\left(-\frac{1}{2}D^2(\mathbf{x}_t, \mathbf{x}_t^i, \mathbf{P}_t^i)\right) \quad (24)$$

where  $n_t$  is the number of modes at time step  $t$ .

## 4 Experiments

In this section, we first discuss a one dimensional tracking simulation, and then compare our algorithm's performance to the classical particle filter for object tracking in real video.

### 4.1 1D Simulation

For this experiment, the process model is given by the following equation,

$$\mathbf{x}_t = 1 + \sin(w\pi(t-1)) + \phi_1 \mathbf{x}_{t-1} + \mathbf{u}_t \quad (25)$$

where  $w = 4e - 2$ ,  $\phi_1 = 0.5$ , and  $\mathbf{u}_t \sim N(1, 0, 2)$  is the random variable for the process noise. The measurement model is given by a non-linear function

$$\mathbf{z}_t = \phi_2(\mathbf{x}_t^2 + \mathbf{x}_t) + \mathbf{v}_t \quad (26)$$

where  $\phi_2 = 0.5$  and the observation noise  $\mathbf{v}_t$  is drawn from a Gaussian distribution  $N(1, 0, 0.1)$ . One hundred particles are drawn by the quasi-random sampling method, and the density is estimated and propagated for each time step  $t$  ( $1 \leq t \leq 150$ ).

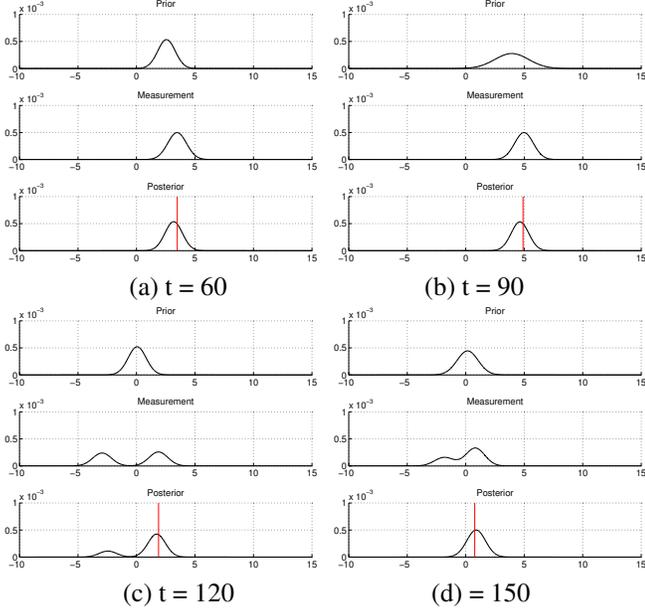


Figure 3: The sequence of 1D tracking simulation. The top of each figure shows the prior probability, the second is the measurement function, and the last one is the posterior probability. In the posterior pdf, the (red) vertical bar denotes the true location of target.

As seen in Figure 3, the multi-modal densities are effectively represented with the mixture of Gaussians, and the state density is propagated through the measurement and update stages. The same experiment was repeated 100 times, and the Mean Squared Error (MSE) between the true and the estimated target location was computed. The MSE and the variance of our algorithm are 0.284 and 0.136 respectively, which are better than the classical particle filter (MSE = 0.340, variance = 0.294).

## 4.2 Object Tracking in Video

Color-based trackers such as [3, 5] search the image space deterministically, and they might fall into a local minimum. To overcome this limitation, the color-based multi-hypothesis tracking was proposed in [13] which is based on the particle filter. We have implemented the probabilistic color-based tracker using the classical particle filter and the kernel-based particle filter with the density approximation, and compare their performance on tracking two objects – a hand carrying a can – in this section.

For both trackers, the state is described by a 10 dimensional vector which is the concatenation of two 5 dimensional

vectors representing two independent ellipses, one for each object.

$$(x_1, y_1, lx_1, ly_1, r_1, x_2, y_2, lx_2, ly_2, r_2) \quad (27)$$

where  $x_i$  and  $y_i$  ( $i = 1, 2$ ) are the location of ellipses,  $lx_i$  is the length of  $x$ -axis,  $ly_i$  is the length of  $y$ -axis, and  $r_i$  is the rotation variable.

As stated previously, we do not assume any specific process model, so that the next position of the tracked object is predicted to be within the Gaussian noise area from the previous position. This assumption is natural for the motion of objects in video, and simple to manage because it is linear. So, the process model equation (10) can be rewritten as follows.

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{u}_t \quad (28)$$

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

The likelihood of each step is based on the similarity of the normalized RGB histogram between the target and the candidates. Supposed that the histogram of the target is denoted by  $c^*(i)$  ( $i = 1 \dots N$ ), where  $N$  is the number of bins in the histogram and  $\sum_{i=1}^N c^*(i) = 1$ . The Bhattacharyya distance in equation (29) is used to measure the similarity between two histograms

$$D[c^*, c(\mathbf{x}_t)] = \left( 1 - \sum_{i=1}^N \sqrt{c^*(i)c(\mathbf{x}_t; i)} \right)^{1/2} \quad (29)$$

and the measurement function at time  $t$  is given by

$$p(\mathbf{z}_t | \mathbf{x}_t) \propto \exp(-\lambda D^2[c^*, c(\mathbf{x}_t)]) \quad (30)$$

where  $\lambda = 10$  is a constant.

Each sample itself is the mean of the particle, and all the particles have equal weights. The covariance matrix is determined by Abramson's law [1] based on the probability computed by equation (30). 400 particles are drawn from the proposal distribution in equation (16), and the incremental density approximation method discussed in section 2.2 is used to compute the measurement density.

The results for both trackers are shown in Figure 4. As seen in the figure, the classical particle filter algorithm fails in tracking early, probably due to the insufficient number of samples, but our algorithm successfully tracks through the whole sequence.

## 5 Conclusions

We proposed a method for approximating a density function, and practically speeding up the approximation procedure. We incorporated these density approximation methods into the particle filter framework, and developed a kernel-based particle filter algorithm. The kernel-based particle filtering needs a relatively small number of particles, and the

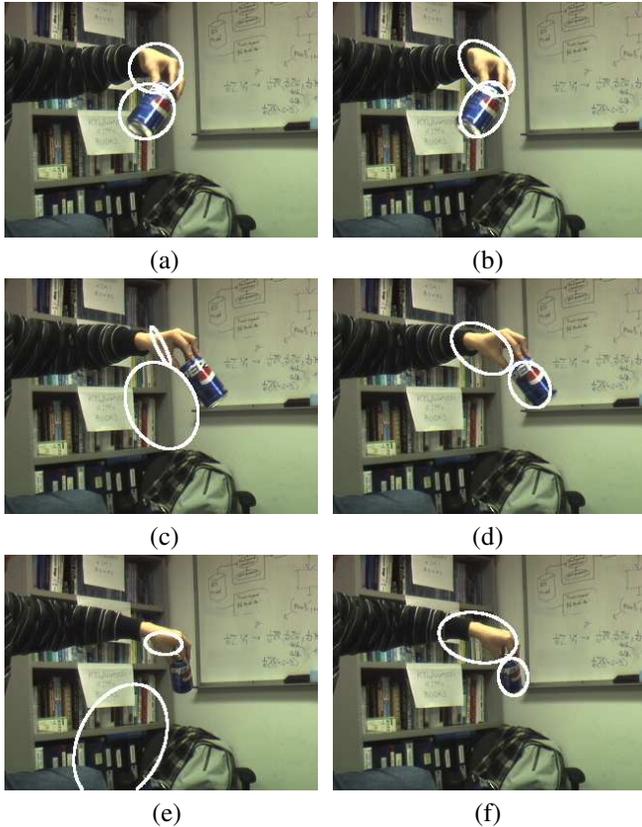


Figure 4: (a)(c)(e) are results of the classical particle filter, and (b)(d)(f) are results of kernel-based particle filter at time  $t = 50, 100, 150$ . The classical particle filter loses the target, but our algorithm tracks successfully through the whole sequence.

computational requirements are reduced by the incremental approximation.

The various simulations show the effectiveness of the density approximation methods and the kernel-based particle filtering, and our algorithm can outperform the classical particle filter for object tracking, using a small number of samples.

## Acknowledgment

The support by Siemens Corporate Research and of General Dynamics Corporation (CTA project, 9614P) is gratefully acknowledged.

## References

- [1] I. Abramson, "On bandwidth variation in kernel estimates - a square root law," *The Annals of Statistics*, vol. 10, no. 4, pp. 1217–1223, 1982.
- [2] T. Cham and J. Rehg, "A multiple hypothesis approach to figure tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Fort Collins, CO, volume II, 1999, pp. 239–249.
- [3] H. Chen and T. Liu, "Trust-region methods for real-time tracking," in *Proc. 8th Intl. Conf. on Computer Vision*, Vancouver, Canada, volume II, 2001, pp. 717–722.
- [4] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 5, pp. 603–619, 2002.
- [5] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Hilton Head, SC, volume II, June 2000, pp. 142–149.
- [6] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Proc. 8th Intl. Conf. on Computer Vision*, Vancouver, Canada, volume I, July 2001, pp. 438–445.
- [7] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of IEEE*, vol. 90, pp. 1151–1163, 2002.
- [8] M. Isard and A. Blake, "Condensation - Conditional density propagation for visual tracking," *Intl. J. of Computer Vision*, vol. 29, no. 1, 1998.
- [9] M. Isard and A. Blake, "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework," *Lecture Notes in Computer Science*, vol. 1406, pp. 893–908, 1998.
- [10] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Hawaii, volume I, 2001, pp. 415–422.
- [11] S. McKenna, Y. Raja, and S. Gong, "Tracking colour objects using adaptive mixture models," *Image and Vision Computing Journal*, vol. 17, pp. 223–229, 1999.
- [12] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, no. 1, pp. 99–110, 2003.
- [13] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. European Conf. on Computer Vision*, Copenhagen, Denmark, volume I, 2002, pp. 661–675.
- [14] V. Philomin, R. Duraiswami, and L. S. Davis, "Quasi-random sampling for condensation," in *ECCV (2)*, 2000, pp. 134–149.
- [15] Y. Rui and Y. Chen, "Better proposal distributions: Object tracking using unscented particle filter," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, Hawaii, volume II, 2001, pp. 786–793.
- [16] C. Stauffer and W. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 747–757, 2000.
- [17] B. Stenger, P. R. S. Mendonça, and R. Cipolla, "Model-based hand tracking using an unscented kalman filter," in *Proc. British Machine Vision Conference*, volume I, (Manchester, UK), September 2001, pp. 63–72.
- [18] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 780–785, 1997.