

# Probabilistic Fusion Tracking Using Mixture Kernel-Based Bayesian Filtering

Bohyung Han\*

Seong-Wook Joo†

Larry S. Davis

Dept. of Computer Science  
University of Maryland  
College Park, MD 20742, USA  
{bhhan, swjoo, lsd}@cs.umd.edu

## Abstract

*Even though sensor fusion techniques based on particle filters have been applied to object tracking, their implementations have been limited to combining measurements from multiple sensors by the simple product of individual likelihoods. Therefore, the number of observations is increased as many times as the number of sensors, and the combined observation may become unreliable through blind integration of sensor observations — especially if some sensors are too noisy and non-discriminative. We describe a methodology to model interactions between multiple sensors and to estimate the current state by using a mixture of Bayesian filters — one filter for each sensor, where each filter makes a different level of contribution to estimate the combined posterior in a reliable manner. In this framework, an adaptive particle arrangement system is constructed in which each particle is allocated to only one of the sensors for observation and a different number of samples is assigned to each sensor using prior distribution and partial observations. We apply this technique to visual tracking in logical and physical sensor fusion frameworks, and demonstrate its effectiveness through tracking results.*

## 1. Introduction

The particle filter provides a natural methodology to fuse measurements from multiple sensors, and such fusion technique has been frequently applied to object tracking. In the general sensor fusion framework based on particle filters, a pre-defined number of samples are drawn and the final likelihood of each sample is determined by the product of the likelihoods measured in individual sensors.

More robust observations would be expected by such integration of multiple sensors, but the cost of the observation also increases in proportion to the number of sensors. Moreover, assigning a fixed number of particles to each sensor,

regardless of the reliability of the sensors, leads to a potential waste of samples, and the blind integration of multiple sensors may corrupt the entire observation if some non-discriminative sensors are involved in the measurement process. Also, the different characteristics of multiple sensors may not be maintained effectively because it is difficult to consistently preserve multi-modality with a single discrete density function in the particle filter framework [6, 19]. To overcome these problems, we propose a probabilistic sensor selection method for the measurement step and employ a mixture of kernel-based Bayesian filters [7].

### 1.1. Related Work

We distinguish between two different kinds of “sensors” — *physical* and *logical* sensors. A physical sensor is a hardware component (video camera, IR camera, ultrasound sensor, microphone, etc.), while a logical sensor refers to a feature (contour, edge, shape, motion, etc.) extraction mechanism applied to the raw data obtained from a physical sensor. Both physical and logical sensor fusion techniques have been applied to various object tracking algorithms.

The logical sensor fusion concept has been addressed in the following papers: Isard and Blake [8] combine skin color detection and contour tracking to improve tracker performance, and the edge and color features are integrated to track elliptical objects in [2, 20]. Also, multiple cues — motion, color, shape, etc. — are fused heuristically to overcome the limitations of the individual modalities in [16, 17]. There are a few papers applying Bayesian inference based on graphical models, but they either use an ad-hoc external indicator to estimate the reliability of a modality [15] or the contribution of the fusion process is limited to creating better proposal distribution in the particle filter framework [21]. In [1], color, motion and shape features are combined, and a variation of the Extended Kalman Filter (EKF) is used for tracking and fusion.

Physical sensor fusion has been applied to video conferencing, multi-modal interfaces and augmented reality, and tracking is a key component for these applications. The combination of video and audio sensors for object tracking

\*Current affiliation: University of California, Irvine, CA 92697

†Current affiliation: Google Inc., Mountain View, CA 94043

is described in [14, 18]. In these methods, particle filters are used both to fuse measurements and to perform tracking.

Sometimes, physical and logical sensors are used together for sensor fusion in the particle filter framework [4, 13]. In [13], generic importance sampling mechanisms for data fusion are introduced, and three cues (color, motion and sound) are modeled by an appropriate likelihood function. On the other hand, [4] proposes a combination of top-down and bottom-up approaches to fuse multiple sensing modalities (color, sound, and contour).

While particle filters provide a principled methodology for sensor fusion, they simply multiply the likelihoods from individual sensors to compute the measurement probability for each sample. Consequently, the multi-modal characteristics of different sensors can be lost and the performance improvement through sensor fusion is reduced. Recently, the mixture Bayesian filtering framework has been proposed to maintain multi-modality by modeling the posterior as a mixture of parametric [3] or non-parametric [19] density functions. This technique has been applied to multi-object tracking in a single camera setting [12, 19].

We adopt the concept of the mixture particle filter to propagate multi-modal density functions obtained from multiple sensors; the summary of our approach is described below.

## 1.2. Our Approach

Our sensor fusion framework is different from previous approaches in the following ways.

1. While the standard sensor fusion technique combines data from individual sensors in the measurement step of the particle filter, our method performs the fusion in the update step. The combined posterior is constructed by a mixture of individual posteriors, which are continuous functions — a mixture of Gaussians.
2. The individual posteriors have different weights to contribute to the combined posterior in proportion to the prior and the measurement confidence. By adopting a weighted mixture model for the posterior instead of a single probability density function, the posterior estimation is more accurate; it gives more weight to reliable sensors for robust state estimation. Therefore, tracker performance can be improved, especially in the presence of clutter and occlusion.
3. The other important difference is that all sensors may not be used for the measurement of each sample. Instead, the sensor for which the actual observation is made is determined probabilistically based on the expected likelihood for each sample. The proposal distribution is constructed from prior knowledge as well as partial observations from each sensor. The sensor selection provides a framework to assign an adaptive number of particles to each sensor based

on its reliability. This cannot easily be done in conventional particle filters since the probability for an arbitrary location in the state space is not available, so the expected likelihoods cannot be obtained; it is possible in kernel-based Bayesian filtering, where all the relevant density functions are represented with a mixture of Gaussians.

This approach is applied to visual tracking in both logical and physical sensor fusion frameworks; color, gradient, template and contour feature are employed for logical sensor fusion, and multiple cameras are used for physical sensor fusion. Also, tracking in the presence of sensor failures is tested; one of the cameras provides completely noisy signals, which is handled by dynamic particle allocation within the mixture kernel-based Bayesian filtering framework.

The rest of this paper is organized as follows. In Section 2, kernel-based Bayesian filtering is reviewed, and our sensor fusion technique is described in Section 3. The application to visual tracking and its performance are demonstrated in Section 4.

## 2. Background

In this section, we provide a brief summary of Kernel-based Bayesian Filtering (KBF) introduced in [7].

### 2.1. Overview

The state variable  $\mathbf{x}_t$  ( $t = 0, \dots, n$ ) in Bayesian filtering is characterized by its probability density function estimated from the sequence of measurements  $\mathbf{z}_t$  ( $t = 1, \dots, n$ ). The conditional density of the state variable given the measurements is propagated through prediction and update stages by a Bayesian framework

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{z}_{1:t-1}) &= \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1} \\ p(\mathbf{x}_t | \mathbf{z}_{1:t}) &= \frac{1}{k} p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}) \end{aligned} \quad (2)$$

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  $t$ ,  $p(\mathbf{x}_t | \mathbf{z}_{1:t})$ , is used as a prior in the next step.

When the prior is represented as a weighted mixture of Gaussians, the mixture representation can be preserved through the prediction and update stages and propagated to the next step.

### 2.2. Kernel-Based Bayesian Filtering

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

$$p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}) = \sum_{i=1}^{n_{t-1}} \omega_{t-1}^i \mathcal{N}(\mathbf{x}_{t-1}^i, \mathbf{P}_{t-1}^i) \quad (3)$$

where  $N(\mathbf{m}, \Sigma)$  represents a normal distribution with mean  $\mathbf{m}$  and covariance  $\Sigma$ .

In the prediction step, the Unscented Transformation (UT) [9, 10] is applied to each mode in the prior so that non-linear process models can be handled. Using the unscented transformation, the prior is transformed to another mixture of Gaussian as follows.

$$p(\mathbf{x}_t | \mathbf{z}_{1:t-1}) = \sum_{i=1}^{n_{t-1}} \hat{\omega}_t^i N(\hat{\mathbf{x}}_t^i, \hat{\mathbf{P}}_t^i) \quad (4)$$

where  $\hat{\omega}_t^i = \omega_{t-1}^i$ , and  $\hat{\mathbf{x}}_t^i$  and  $\hat{\mathbf{P}}_t^i$  are the transformed mean and covariance, respectively. This non-linear transformation is accurate up to second order.

Density interpolation based on the Non-Negative Least Square (NNLS) method is incorporated to parameterize the measurement density with a Gaussian mixture, and the measurement function with  $m_t$  Gaussians at time  $t$  is given by

$$p(\mathbf{z}_t | \mathbf{x}_t) = \sum_{i=1}^{m_t} \tau_t^i N(\mathbf{x}_t^i, \mathbf{R}_t^i) \quad (5)$$

where  $\tau_t^i$  is the weight and  $\mathbf{R}_t^i$  is the covariance associated with the mean  $\mathbf{x}_t^i$  ( $i = 1, \dots, m_t$ ).

In the update step, the posterior is obtained by the products of the Gaussian pairs between prediction and measurement density. Even though the derived density function is also a weighted Gaussian mixture, the exponential increase of Gaussian components in the mixture during the propagation makes the whole procedure intractable. In order to avoid this problem, a density approximation technique is applied. It allows us to maintain a compact and accurate density representation even after many stages of density propagation. After the update step, the final posterior distribution is given by

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) = \sum_{i=1}^{n_t} \omega_t^i N(\mathbf{x}_t^i, \mathbf{P}_t^i) \quad (6)$$

where  $n_t$  is the number of modes at time step  $t$  and the sum of  $\omega_t^i$  is equal to 1.

### 2.3. Discussion of Kernel-Based Bayesian Filtering

Kernel-based Bayesian filtering has an advantage over conventional particle filters. It is generally known that a continuous proposal distribution can improve the quality of sampling [6], so the natural filtering algorithm based on continuous density functions ameliorates *degeneracy* or the *loss of diversity* problem. Therefore, the kernel-based Bayesian filtering is more efficient since the number of samples can be reduced.

Also, there is an important characteristic of kernel-based Bayesian filtering. Unlike the particle filters based on dis-

crete probability density functions, the probability at an arbitrary location in the state space can be computed analytically in the kernel-based Bayesian filter. This property plays an important role in computing the expected likelihood of each sample before the “real” observation, and will be utilized in our sensor fusion framework.

In the next section, we explain how the kernel-based Bayesian filtering framework is employed for sensor fusion.

## 3. Fusion Tracking by Mixture KBF

Suppose that we have  $K$  sensors and try to fuse data from those sensors. If the mixture weight of each sensor is given by  $\pi_{t-1}^i$  ( $i = 1, \dots, K$ ) at time  $t - 1$ , the posterior at time step  $t - 1$  — also the prior at time  $t$  — is defined by

$$p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}) = \sum_{k=1}^K \pi_{t-1}^k p_k(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}) \quad (7)$$

where  $p_k(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1})$  is the posterior of an individual sensor at time  $t - 1$ , which is a mixture of Gaussians.

Our purpose is to preserve this representation through the iterations of Bayesian filtering. The overall procedure for an individual Bayesian filter is similar to the description in Section 2, and we will explain how to combine the information from multiple sensors and how sensors interact with each other.

### 3.1. Prediction Step and Proposal Distribution

We make a prediction for an individual Bayesian filter independently by the unscented transformation as described in 2.2, and the predicted density function is given by

$$p(\mathbf{x}_t | \mathbf{z}_{1:t-1}) = \sum_{k=1}^K \pi_{t-1}^k p_k(\mathbf{x}_t | \mathbf{z}_{1:t-1}) \quad (8)$$

Since our method selects a sensor for observation probabilistically, the proposal distribution is very important to overall performance. In the particle filter framework, there are several techniques to improve the proposal distribution such as the use of an auxiliary tracker with different features [8], unscented particle filter [10, 14], and multi-stage sampling [7, 12].

We combine the prior and partial observation distribution from each individual filter through a 2-stage sampling scheme to construct the proposal distribution, which improves the effectiveness of particles. Since the posterior in the previous step in Eq. (7) is the combined information obtained from individual sensors, it should be more reliable than the individual posterior. So, the initial proposal distribution, denoted by  $q^1(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t})$ , is common for every sensor and is equal to the predicted distribution in Eq. (8).

$$q^1(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_{1:t}) = p(\mathbf{x}_t | \mathbf{z}_{1:t-1}) \quad (9)$$

In the second stage, the proposal distribution for each sensor,  $q_k^2(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t})$ , is determined by the combination of the initial proposal distribution and the partial observation from each sensor as follows:

$$q_k^2(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t}) = (1 - \alpha)q^1(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t}) + \alpha p_k^1(\mathbf{z}_t|\mathbf{x}_t), \quad (10)$$

where  $p_i^1(\mathbf{z}_t|\mathbf{x}_t)$  is the initial measurement density and  $\alpha$  is a constant. The combined proposal distribution is given by

$$q^2(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t}) = \sum_{k=1}^K \pi_{t-1}^k q_k^2(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t}). \quad (11)$$

This 2-stage sampling strategy improves the sampling quality and possibly reduces the number of samples required, since the proposal distribution combines the priors of all sensors and the partial observations in the current step.

### 3.2. Measurement Step

The measurement step is also composed of two stages in accordance with the 2-stage sampling. The main purpose of the 2-stage sampling is to improve the proposal distribution in a progressive manner. By assigning a fixed number of particles to each sensor in the first stage, the degeneracy problem — the situation that no particle is drawn from one or more sensors and the measurement density does not become available — can be avoided. This situation may happen when only a couple of sensors dominate the posterior due to their strong measurement likelihoods and the rest of the sensors have negligible mixture weights.

In the first stage, the samples are drawn from the common proposal distribution  $q^1(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t})$ , and the observations are made in all sensors for the same locations in the state space. Then, the initial observation results in every sensor  $p_k^1(\mathbf{z}_t|\mathbf{x}_t)$  ( $k = 1, \dots, K$ ) is reflected in the proposal distribution for the next stage, as shown in Eq. (11), from which samples are drawn. In the second stage, each sample is assigned to only one sensor probabilistically for observation by considering the prior and likelihood expectation. The probability that the  $k$ -th sensor is selected is given by

$$p(\text{sel}(i) = k) = \frac{\pi_{t-1}^k (\beta p_k(\mathbf{x}_t^{(i)}|\mathbf{z}_{1:t-1}) + (1 - \beta)p_k^1(\mathbf{z}_t|\mathbf{x}_t^{(i)}))}{\sum_{j=1}^K \pi_{t-1}^j (\beta p_j(\mathbf{x}_t^{(i)}|\mathbf{z}_{1:t-1}) + (1 - \beta)p_j^1(\mathbf{z}_t|\mathbf{x}_t^{(i)}))}. \quad (12)$$

where  $\text{sel}(i)$  is the selected sensor number for the  $i$ -th sample,  $\beta$  is a constant,  $p_k(\mathbf{x}_t^{(i)}|\mathbf{z}_{1:t-1})$  is the prediction probability of the  $i$ -th particle in the  $k$ -th sensor, and  $p_k^1(\mathbf{z}_t|\mathbf{x}_t^{(i)})$  is the probability of the  $i$ -th sample given the initial measurement density. The sensor selection for the  $i$ -th sample is given by

$$\text{sel}(i) = \arg \min_s \left( \sum_{k=1}^s p(\text{sel}(i) = k) > r_i \right) \quad (13)$$

where  $r_i$  is a random number from a uniform distribution in  $[0, 1)$ .

This procedure is similar to the E-step of the EM algorithm, and cannot be done in conventional particle filters based on discrete density functions since it is difficult to obtain probabilities at arbitrary locations. The sensor expected to produce the highest likelihood is prioritized for observation, and is given more samples to improve the robustness of the measurement density. The sampling and measurement procedure in the second stage is illustrated in Figure 1.

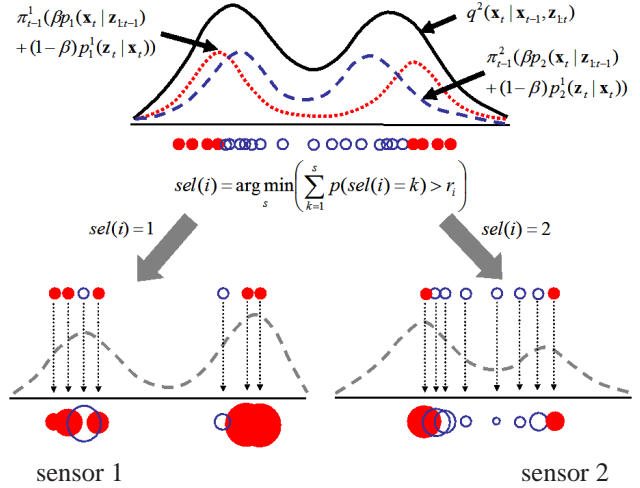


Figure 1. An example of sampling and measurement procedure in the second stage. The proposal distribution  $q_k^2(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t})$  is constructed from the prior and the partial measurement density function of the  $k$ -th sensor, and  $q^2(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_{1:t})$  is the mixture of  $q_k^2$ , ( $k = 1, 2$ ). **(Top)** The samples such that  $p(\text{sel}(i) = 1) \geq p(\text{sel}(i) = 2)$  are represented with red (shaded) circles, and the rest are represented with blue circles. The sensor selection for each sample is performed by Eq. (13). **(Bottom)** Because the sensor selection for each particle is probabilistic, red and blue particles are mixed in each sensor. Based on the measurements of each sensor, the final measurement density functions are constructed by density interpolation.

The multi-stage measurements performed in the individual filter is identical to kernel-based Bayesian filter [7], where a density interpolation technique based on the non-negative least square method is used to obtain measurement density functions. The individual measurement density function of the  $k$ -th sensor at time step  $t$ ,  $p_k(\mathbf{z}_t|\mathbf{x}_t)$ , is given by

$$p_k(\mathbf{z}_t|\mathbf{x}_t) = \sum_{i=1}^{m_{t,k}} \kappa_{t,k}^i N(\mathbf{x}_t^i, \mathbf{R}_{t,k}^i) \quad (14)$$

where  $m_{t,k}$  is the number of components,  $\kappa_{t,k}^i$  is an unnormalized weight of each Gaussian component, and  $\mathbf{x}_t^i$  and  $\mathbf{R}_t^i$  are the mean and covariance in the  $k$ -th measurement density, respectively.

### 3.3. Update Step

In the update step, the prior and the measurement information are combined to construct the posterior for each sensor, and the individual posteriors are combined to derive the overall posterior probability density function. Define the combined observation density function as

$$p(\mathbf{z}_t|\mathbf{x}_t) = \frac{\sum_{k=1}^K \pi_{t-1}^k p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1}) p_k(\mathbf{z}_t|\mathbf{x}_t)}{\sum_{k=1}^K \pi_{t-1}^k p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1})} \quad (15)$$

Then the combined posterior distribution is given by

$$\begin{aligned} & p(\mathbf{x}_t|\mathbf{z}_{1:t}) \\ &= \frac{p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{\int p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t} \\ &= \frac{p(\mathbf{z}_t|\mathbf{x}_t)\sum_{k=1}^K \pi_{t-1}^k p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{\int p(\mathbf{z}_t|\mathbf{x}_t)\sum_{k=1}^K \pi_{t-1}^k p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t} \\ &= \frac{\sum_{k=1}^K \pi_{t-1}^k p_k(\mathbf{z}_t|\mathbf{x}_t)p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{\sum_{k=1}^K \pi_{t-1}^k \int p_k(\mathbf{z}_t|\mathbf{x}_t)p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t} \\ &= \sum_{k=1}^K \left( \frac{\pi_{t-1}^k \int p_k(\mathbf{z}_t|\mathbf{x}_t)p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t}{\sum_{j=1}^K \pi_{t-1}^j \int p_j(\mathbf{z}_t|\mathbf{x}_t)p_j(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t} \right) p_k(\mathbf{x}_t|\mathbf{z}_{1:t}) \\ &= \sum_{k=1}^K \pi_t^k p_k(\mathbf{x}_t|\mathbf{z}_{1:t}) \end{aligned} \quad (16)$$

where the posterior for each sensor is given by

$$p_k(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{p_k(\mathbf{z}_t|\mathbf{x}_t)p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{\int p_k(\mathbf{z}_t|\mathbf{x}_t)p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t} \quad (17)$$

and the new weight for the  $k$ -th filter,  $\pi_t^k$ , is

$$\begin{aligned} \pi_t^k &= \frac{\pi_{t-1}^k \int p_k(\mathbf{z}_t|\mathbf{x}_t)p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t}{\sum_{j=1}^K \pi_{t-1}^j \int p_j(\mathbf{z}_t|\mathbf{x}_t)p_j(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t} \\ &= \frac{\pi_{t-1}^k p_k(\mathbf{z}_t|\mathbf{z}_{1:t-1})}{\sum_{j=1}^K \pi_{t-1}^j p_j(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \end{aligned} \quad (18)$$

Note that  $p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1})$  and  $p_k(\mathbf{z}_t|\mathbf{x}_t)$  are already known and that the computation of the individual posterior,  $p_k(\mathbf{x}_t|\mathbf{z}_{1:t})$ , is straightforward using KBF.

The remaining issue is the calculation of the mixture weight  $\pi_t^k$ . To obtain the mixture weight for each filter, it is necessary to compute the component likelihoods  $p_k(\mathbf{z}_t|\mathbf{z}_{1:t-1})$  ( $k = 1, \dots, K$ ) as shown in Eq. (19).

$$\begin{aligned} p_k(\mathbf{z}_t|\mathbf{z}_{1:t-1}) &= \int p_k(\mathbf{z}_t|\mathbf{x}_t)p_k(\mathbf{x}_t|\mathbf{z}_{1:t-1})d\mathbf{x}_t \\ &\approx \int \sum_{i=1}^{m_{t,k}} \kappa_{t,k}^i N(\mathbf{x}_{t,k}^i, \mathbf{R}_{t,k}^i) d\mathbf{x}_t \\ &= \sum_{i=1}^{m_{t,k}} \kappa_{t,k}^i \end{aligned} \quad (19)$$

Therefore, the mixture weight at time step  $t$  is given by

$$\pi_t^k = \frac{\pi_{t-1}^k \sum_{i=1}^{m_{t,k}} \kappa_{t,k}^i}{\sum_{j=1}^K \pi_{t-1}^j \sum_{i=1}^{m_{t,j}} \kappa_{t,j}^i}. \quad (20)$$

In other words, the updated mixture weights depend on the confidence of new observations in each filter as well as the prior knowledge.

## 4. Experiments

The proposed sensor fusion technique is applied to visual tracking problem, where several logical and physical sensors are integrated and the weighted mixture density function is propagated in the framework of mixture KBF to estimate target state. Also, we tested the performance of our sensor fusion algorithm under the condition that some sensors fails temporarily and are completely unreliable.

### 4.1. Logical Sensor Fusion

We first apply the fusion technique to object tracking using multiple logical sensors; the features (sensors) used are (1) color, (2) gradient, (3) template, and (4) contour. For the color and the gradient sensor, the target appearance is modeled using histograms and the Bhattacharyya distance is used to compute likelihoods. The template sensor measures the mean squared difference of the color pixels in a smoothed image template. Finally, the contour sensor uses the magnitude of the gradients along the normal direction around the perimeter of an ellipse. For each sensor, tracking is performed independently by KBF, but all the sensors interact and compete for dynamic particle allocation as explained in Section 3. The object is tracked in a 4D state space consisting of image location  $(x, y)$ , in-plane rotation, and scale, and the random walk is chosen as the process model.

Figure 2 present the results of tracking with four different sensors by the mixture KBF. Our algorithm tracked a target successfully under significant pose variations and severe appearance changes for the entire 500 frames. The number of samples drawn is 90 altogether — 10 in the first stage and 80 in the second stage, so the total number of observations in all sensors combined is 120. This result is presented in Figure 2, where the images for template and contour features are omitted for lack of space. Note that the gradients in the  $x$  and  $y$  direction are mapped to R and G space in the gradient images, respectively.

### 4.2. Physical Sensor Fusion

There has been a significant amount of prior work on tracking using multiple cameras [5, 11], but few attempts have been made to control the degree of contribution from each of the cameras.

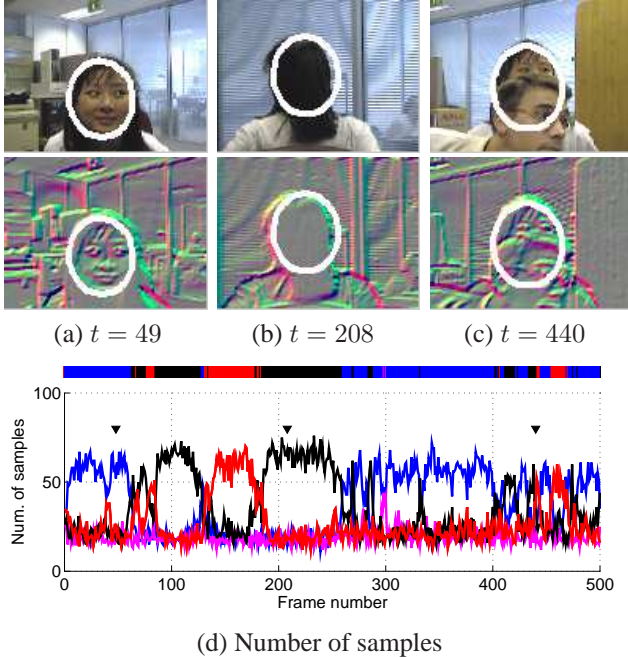


Figure 2. Tracking results by logical sensor fusion. (a)(b)(c) (Row1) Color (Row2) Gradient (d) (Blue) Color, (Red) Gradient (Magenta) Template (Black) Contour (The color bar indicates the dominating sensor in each frame and triangle marks correspond to the frames shown in this figure.)

We assume objects are moving on a ground plane and all cameras have some common field of view of those objects. The common state space is defined as the 2D location  $(x, y)$  in the canonical top view, and the state vector is transformed into each view for an observation using the ground plane homography. Even though the cameras are static, no background subtraction information is used for tracking.

The process model is also the random walk, and the likelihood of each sample is based on the similarity of the RGB histogram between the target and the candidates.

We tested our method on a sequence captured by two cameras in which a walking person is tracked. The appearance model is constructed based on two separate histograms — one for the upper and the other for the lower body, and the Bhattacharyya distance is used to compute the joint likelihood.

Figure 3 illustrate the result of tracking a person using two cameras in an indoor environment. In this example, 60 samples are used — 10 for each camera in the first stage and 40 in the second stage. Even with frequent occlusion and clutter, the person is successfully tracked throughout the sequence by the adaptive collaboration of two cameras. Also, the mixture weights and the number of particles assigned to each camera are updated at each frame depending on visibility and the distinctiveness of the target in each view, which is illustrated in Figure 4. As observed in Figure 4, the mix-

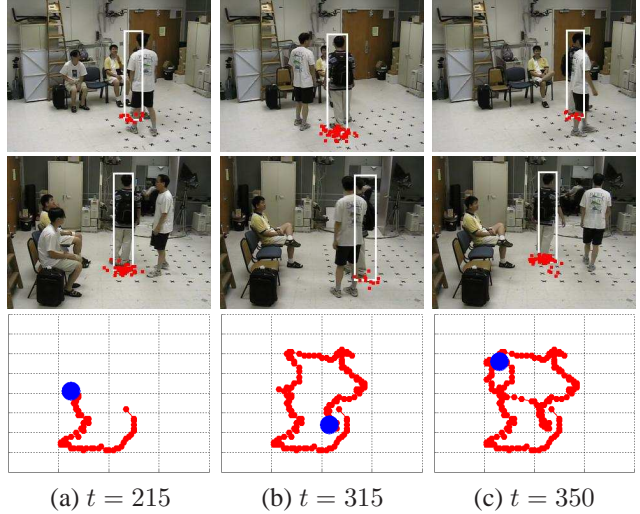


Figure 3. Single object tracking example. (Top, Middle) Results in camera 1 and camera 2. The red dots represent sample locations in each image plane. (Bottom) Current location (blue circle) and trajectory (red circles and lines) in top view for each frame

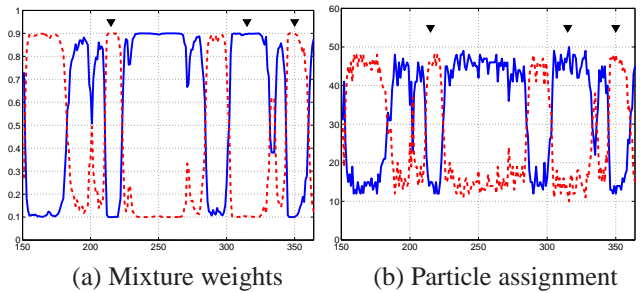
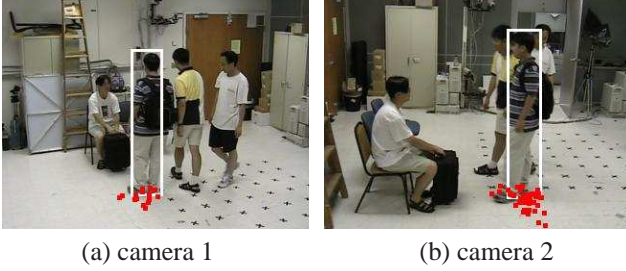


Figure 4. Mixture weights and particle assignments for each sensor in each frame. Blue solid lines and red dotted lines are for camera 1 and camera 2, respectively. The frames shown in Figure 3 are marked with triangles in each figure.

ture model preserves a component with a small weight and allows it to make a significant contribution later.

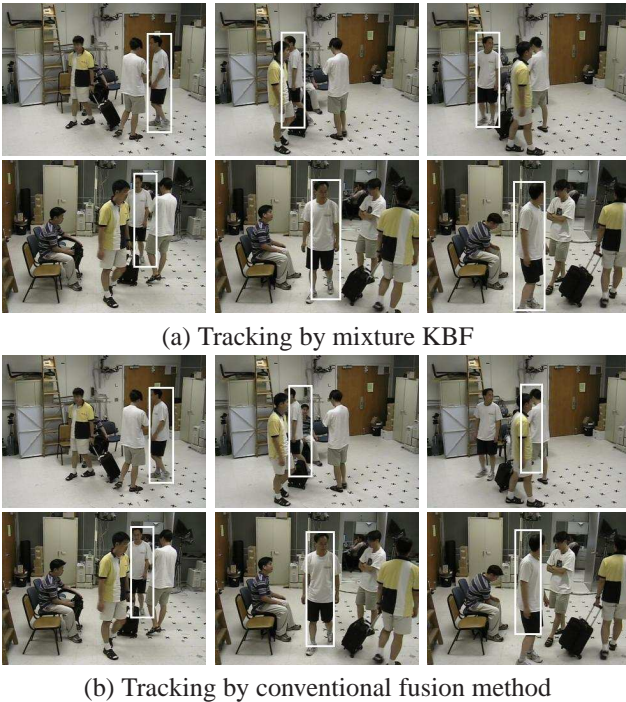
Occlusion and clutter can significantly affect the likelihood; but experiments indicate that appearance change is also a very important factor. Consequently, a very high mixture weight is sometimes given to one camera due to pose and lighting variations as illustrated in Figure 5, even though the target can be seen clearly in both cameras.

We also compared the tracking performance of our method and a conventional fusion by particle filtering, which is presented in Figure 6. The sequence for this experiment is similar to the one used for Figure 3, but there are many dynamic occlusions between two people whose appearances are very similar because both are wearing white T-shirts. The same number of measurements (50 altogether) is performed for both methods; in the case of our method, 5 samples are drawn at the first stage of measurement step, and 40 samples are then dynamically allocated to both cam-



(a) camera 1 (b) camera 2

Figure 5. Severe difference of mixture weights due to appearance change ( $t = 170$ ). In this sequence, the initial target appearance model is constructed based on the frontal view of the person, and the lighting condition varies in each frame. (**Camera 1**) mixture weight = 0.1210, number of particles = 15, (**Camera 2**) mixture weight = 0.8790, number of particles = 45. Note that the particles drawn in the first stage are counted twice.

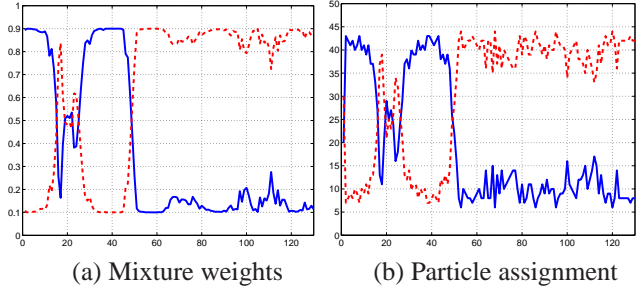


(a) Tracking by mixture KBF (b) Tracking by conventional fusion method

Figure 6. Comparison between mixture KBF and conventional fusion by particle filtering. The results at time  $t = 18, 54, 67$  are presented for each case (a) and (b) where the first and second row represent results in camera 1 and camera 2.

eras. After the first occlusion, both tracking algorithms recovered from short-term failures but the conventional fusion method based on particle filtering lost the target after the second occlusion. On the other hand, our method succeeded in tracking the target even after the second occlusion.

In our method, the number of observations in camera 2 between  $t = 54$  and  $t = 67$  is consistently much more than camera 1 as illustrated in Figure 7. It suggests that tracking by mixture KBF is successful because the utilization of the more reliable sensor (camera 2) is maximized by the



(a) Mixture weights (b) Particle assignment

Figure 7. Mixture weights and particle assignments for each sensor in each frame. Blue solid lines and red dotted lines are for camera 1 and camera 2, respectively.

dynamic sample allocation.

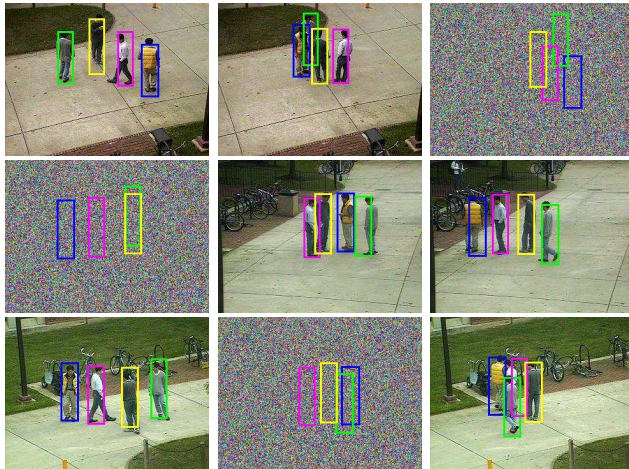
The comparison with mixture particle filter [12, 19] is not straightforward since the “competing” mechanism among sensors is hard to implement in particle filters based on discrete density functions. However, the comparison between KBF and conventional particle filter in [7] suggests the potential advantage of mixture KBF.

### 4.3. Fusion in the Presence of Sensor Failures

In many vision systems, it is common that some sensor data is temporarily missing or is totally unreliable due to sensor noise, external blockages, hardware/software errors, etc. The performance of our sensor fusion algorithm is also tested in the presence of sensor failures and compared with the conventional fusion method using particle filtering. Figure 8 illustrates the result of multi-object tracking using three cameras in an outdoor scene, where one of the cameras fails temporarily (50 ~ 100 frames). 5 particles are used at the first stage of measurement and 60 particles are distributed to 3 sensors, resulting in a total of 75 observations. In spite of frequent occlusions amongst the group of people and temporary sensor failures, tracking is successful for the entire 900 frames. The number of total observations made corresponds to allocating only 25 samples for each sensor in the conventional fusion method; according to our experiments, tracking by the conventional method is less stable than the proposed method. In our method, only around 10 samples are allocated to every person in the failed sensors, which indicates that our fusion technique minimizes the contribution of unreliable sensors effectively.

## 5. Conclusion

We presented a probabilistic sensor fusion technique based on mixture kernel-based Bayesian filtering. This framework provides a principled methodology to select sensors probabilistically for measurements. By assigning particles to a sensor based on its reliability, the observation becomes more robust and the effectiveness of particles is improved. We applied our algorithm to various sensor fusion



(a) Tracking by mixture KBF



(b) Tracking by conventional fusion method

Figure 8. Tracking people in existence with temporal sensor failures. (Left)  $t = 140$  (Middle)  $t = 356$  (Right)  $t = 558$  (Row1) camera1 (Row2) camera2 (Row3) camera3

scenarios, and presented tracking results in the presence of severe occlusion, clutter, and sensor failures.

## Acknowledgement

This research was funded in part by the U.S. Government's VACE program.

## References

- [1] Y. Azoz, L. Devi, and R. Sharma. Reliable tracking of human arm dynamics by multiple cue integration and constraint fusion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998. 1
- [2] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Santa Barbara, CA, pages 232–237, 1998. 1
- [3] R. Chen and J. Liu. Mixture kalman filters. *J. Roy. Statist. Soc. B.*, 62:493–508, 2000. 2

- [4] Y. Chen and Y. Rui. Real-time speaker tracking using particle filter sensor fusion. *Proceedings of IEEE*, 92(3):485–494, 2004. 2
- [5] S. L. Dockstader and A. M. Tekalp. Multiple camera tracking of interacting and occluded human motion. *Proceedings of the IEEE*, 89(10):1441–1455, 2001. 5
- [6] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001. 1, 3
- [7] B. Han, Y. Zhu, D. Comaniciu, and L. Davis. Kernel-based Bayesian filtering for object tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, 2005. 1, 2, 3, 4, 7
- [8] M. Isard and A. Blake. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. *Proc. European Conf. on Computer Vision*, Freiburg, Germany, 1406:893–908, 1998. 1, 3
- [9] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proceedings SPIE*, volume 3068, pages 182–193, 1997. 3
- [10] R. Merwe, A. Doucet, N. Freitas, and E. Wan. The unscented particle filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000. 3
- [11] A. Mittal and L. S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *Intl. J. of Computer Vision*, 51(3):189–203, 2003. 5
- [12] K. Okuma, A. Taleghani, N. Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proc. European Conf. on Computer Vision*, Prague, Czech Republic, May 2004. 2, 3, 7
- [13] P. Perez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particle filter. *Proceedings of IEEE*, 92(3):495–513, 2004. 2
- [14] 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, pages 786–793, 2001. 2, 3
- [15] J. Sherrah and S. Gong. Continuous global evidence-based bayesian modality fusion for simultaneous tracking of multiple objects. In *Proc. 8th Intl. Conf. on Computer Vision*, Vancouver, Canada, 2001. 1
- [16] N. T. Siebel and S. J. Maybank. Fusion of multiple tracking algorithms for robust people tracking. In *Proc. European Conf. on Computer Vision*, Copenhagen, Denmark, volume IV, pages 373–387, 2002. 1
- [17] J. Triesch and C. von der Malsburg. Democratic integration: Self-organized integration of adaptive cues. *Neural Computation*, 13(9):2049–2074, 2001. 1
- [18] J. Vermaak, A. Blake, M. Gangnet, and P. Perez. Sequential monte carlo fusion of sound and vision for speaker tracking. In *Proc. 8th Intl. Conf. on Computer Vision*, Vancouver, Canada, volume I, pages 714–746, 2001. 2
- [19] J. Vermaak, A. Doucet, and P. Perez. Maintaining multi-modality through mixture tracking. In *Proc. 9th Intl. Conf. on Computer Vision*, Nice, France, volume II, 2003. 1, 2, 7
- [20] Y. Wu and T. Huang. A co-inference approach to robust visual tracking. In *Proc. 8th Intl. Conf. on Computer Vision*, Vancouver, Canada, volume II, pages 26–33, 2001. 1
- [21] X. Zhong, J. Xue, and N. Zheng. Graphical model based cue integration strategy for head tracking. In *Proc. British Machine Vision Conference*, Edinburgh, UK, 2006. 1