

# Multiple Object Tracking with Kernel Particle Filter

Cheng Chang\*      Rashid Ansari      Ashfaq Khokhar  
ECE Dept., University of Illinois at Chicago  
{cchang, ansari, ashfaq}@ece.uic.edu, <http://www.ece.uic.edu/~cchang>

## Abstract

A new particle filter, Kernel Particle Filter (KPF), is proposed for visual tracking for multiple objects in image sequences. The KPF invokes kernels to form a continuous estimate of the posterior density function and allocates particles based on the gradient derived from the kernel density estimate. A data association technique is also proposed to resolve the motion correspondence ambiguities that arise when multiple objects are present. The data association technique introduces minimal amount of computation by making use of the intermediate results obtained in particle allocation. We show that KPF performs robust multiple object tracking with improved sampling efficiency.

## 1 Introduction

In computer vision, the superior performance of the particle filter (PF) over Kalman filter in tracking objects in heavy clutter has been reported in [11] and by many other research groups thereafter. Despite its success in various applications, it was observed that PF does not perform well when the dynamic system has a very small process noise, or if the observation noise has very small variance. In these cases, the particle set quickly collapses to one single point in the state space and the filter performance is severely affected.

Various approaches have been proposed to improve particle filter's performance in handling weak dynamic models and high dimensionality. Existing methods are based on the idea that samples should be allocated to the high probability mass areas [8], and a general approach is to introduce optimization procedures to re-allocate particles to peaks of the likelihood after the initial sampling [2, 8, 17]. To maintain particles as fair samples from the posterior, multiple Markov chains are used in [6] to explore the state space. Previously we have proposed a Kernel Particle Filter (KPF) [4, 3] that invokes mean shift [5] to allocate particles more efficiently and use importance sampling to maintain



**Figure 1. Three independent PF trackers tracking 3 tennis balls. The particle set of the leftmost ball collapses to a mode created by the nearby ball.**

fair samples from the posterior. In this paper we review KPF and employ it in a new formulation of the problem of tracking multiple objects.

In the context of multiple object tracking (MOT), establishing the correspondence between objects and observations, namely the *motion correspondence* problem [1, 15, 2, 22], is not a trivial task. Particle filtering is appealing in MOT because of its ability to carry multiple hypotheses. Direct application of PF on multiple object tracking, however, is not feasible because of the following facts: First, the standard particle filter does not define a way to identify individual modes (or hypotheses). Secondly, the particle set of a standard particle filter often quickly collapses to one mode and discards all other modes, as indicated in [10] and [20]. Fig. 1 shows this typical error of a particle filter. Thirdly, a particle filter working in the product space of individual object spaces, obtained by concatenating the state variables of individual objects, is computationally expensive [12, 13, 21]. Finally, particle filters tracking objects individually lack a consistent way to resolve the ambiguities that arise in associating objects with measurements.

This paper examines MOT under the assumption that the tracked objects are indistinguishable from each other in terms of the observation model used. The proposed work explores kernel particle filter's ability of tracking multiple objects in a single-object state space with a particle-based data association technique. The work differentiates itself from previous work by using an iterative mode-seeking technique that finds the modes in the *posterior* density, in-

\*Cheng Chang is now with Microsoft Corporation.

stead of the *likelihood* density. An efficient association technique introduces minimal amount of extra computation by making use of the intermediate results obtained during mode-seeking. We show that KPF performs robust single and multiple object tracking with improved sampling efficiency.

## 2 Kernel Particle Filter

The idea of kernel-based tracking was originally published in [7] where kernels are used for object representation and localization. Recently, mean shift is used with a particle filter to find the *likelihood* modes [9]. We have used mean shift as a mode-seeking procedure to locate the *posterior* modes [4, 3].

### 2.1 Kernel-based Posterior Estimation

Denote the target state and the observation at (discrete) time  $t$  as  $\mathbf{x}_t$  and  $\mathbf{y}_t$  respectively and  $\mathbf{x}_t \in \mathbb{R}^{n_x}$ ,  $\mathbf{y}_t \in \mathbb{R}^{n_y}$ . Let  $\mathbf{Y}_t = \{\mathbf{y}_0, \dots, \mathbf{y}_t\}$  be the history of observations up to time  $t$ . In recursive Bayesian estimation, the posterior PDF is estimated by propagating the PDF over time:

$$p(\mathbf{x}_t | \mathbf{Y}_t) \propto p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{Y}_{t-1}), \quad (1)$$

where

$$p(\mathbf{x}_t | \mathbf{Y}_{t-1}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{Y}_{t-1}) d\mathbf{x}_{t-1}. \quad (2)$$

Kernel density estimate (KDE) [16] is used in this work to form a continuous estimate of the posterior in order to facilitate gradient estimation. Given a particle set at time  $t$ :  $\mathbf{S}_t = \{\mathbf{s}_t^{(n)}\}_{n=1}^N$  and associated weights  $\{w_t^{(n)}\}_{n=1}^N$ , the kernel density estimation of the posterior with kernel  $K$  can be formulated as

$$\hat{p}(\mathbf{x}_t | \mathbf{Y}_t) = \sum_{n=1}^N K_\lambda(\mathbf{x}_t - \mathbf{s}_t^{(n)}) w_t^{(n)} \quad (3)$$

where  $K_\lambda$  is the kernel scaled by the kernel width  $\lambda$ :

$$K_\lambda(\mathbf{x}_t - \mathbf{s}_t^{(n)}) = \frac{1}{\lambda^{n_x}} K\left(\frac{\mathbf{x}_t - \mathbf{s}_t^{(n)}}{\lambda}\right). \quad (4)$$

When a Gaussian kernel is used and the posterior is also Gaussian with a unit covariance matrix, the optimal kernel width that minimizes the Mean Integrated Square Error (MISE) is given by [16]

$$\lambda_{opt} = \left(\frac{4}{(n_x + 2)N}\right)^{\frac{1}{n_x + 4}}. \quad (5)$$

Although (5) is optimal in the  $L^2$  sense only for equally weighted particles and Gaussian density, it can still be used in the general case to obtain a suboptimal filter [14]. In practice when densities are often multi-modal, we let  $\lambda = \lambda_0 = \frac{1}{2}\lambda_{opt}$  [14, 16].

### 2.2 Posterior Gradient Estimation

Given the posterior estimation (3), we now estimate its gradient and move particles along the gradient direction toward the modes of the posterior. This can be achieved using the mean shift procedure [5]. In this procedure, each particle is moved to its sample mean determined by

$$m(\mathbf{s}_t^{(n)}) = \frac{\sum_{l=1}^N H_\lambda(\mathbf{s}_t^{(n)} - \mathbf{s}_t^{(l)}) w_t^{(l)} \mathbf{s}_t^{(l)}}{\sum_{l=1}^N H_\lambda(\mathbf{s}_t^{(n)} - \mathbf{s}_t^{(l)}) w_t^{(l)}}. \quad (6)$$

It is shown that the mean shift vector  $m(\mathbf{x}) - \mathbf{x}$  using kernel  $H$  would be in the gradient direction of (3) if the kernel profiles satisfy  $h(r) = -ck'(r)$  for all  $r \in [0, \infty)$  and some  $c \in (0, \infty)$  [5]. It is also shown in [5] that mean shift is a steepest ascent procedure that will find *all* the local maxima of the KDE of the posterior.

Given  $\mathbf{S}_t$  and the associated weights, the empirical covariance matrix  $\hat{\mathbf{C}}_t$  of the particle set is computed first and a “whitening” step is performed on each particle in which  $\mathbf{s}_t^{(n)}$  is changed to  $\mathbf{A}_t^{-1}\mathbf{s}_t^{(n)}$ , where  $\mathbf{A}_t\mathbf{A}_t^T = \hat{\mathbf{C}}_t$ , in order to achieve a unit covariance matrix. A symmetric  $K_\lambda$  can then be used for mean shift and the results are multiplied by  $\mathbf{A}_t$ .

### 2.3 Particle Re-weighting

The mean shift can be applied repeatedly to a particle set. A problem arises when particles change their positions: the new particles do not follow the posterior distribution anymore. This is compensated in KPF by re-weighting the particles. Denote the particle set after the  $i$ -th mean shift procedure at time  $t$  as  $\{\mathbf{s}_{t,i}^{(n)}\}$ . After each mean shift procedure, the weight is re-computed as the posterior density evaluated at the new particle positions augmented with a particle density balancing factor

$$w_{t,i}^{(n)} = \frac{p(\mathbf{s}_{t,i}^{(n)} | \mathbf{Y}_t)}{q_{t,i}(\mathbf{s}_{t,i}^{(n)})}, \quad (7)$$

where the denominator is the new proposal density that captures the non-uniformity of the new particle set

$$q_{t,i}(\mathbf{x}_t) = \sum_{l=1}^N K_\lambda(\mathbf{x}_t - \mathbf{s}_{t,i}^{(l)}) \quad (8)$$

**Table 1. The Kernel Particle Filter algorithm**


---

Given  $\{\mathbf{s}_{t-1}^{(\cdot)}, w_{t-1}^{(\cdot)}\}$  :  
 $\mathbf{s}_{t,0}^{(\cdot)} = \text{Propagate}(\{\mathbf{s}_{t-1}^{(\cdot)}, w_{t-1}^{(\cdot)}\})$ ;  
 Calculate the empirical covariance matrix  $\hat{\mathbf{C}}_t$  of  $\{\mathbf{s}_{t,0}^{(\cdot)}\}$ ;  
 Compute  $\mathbf{A}_t$  such that  $\mathbf{A}_t \mathbf{A}_t^T = \hat{\mathbf{C}}_t$ ;  
 Draw  $\mathbf{e}^{(\cdot)} \sim N(\mathbf{0}, \mathbf{I}_{n_x})$ ;  
 $\mathbf{s}_{t,1}^{(\cdot)} = \mathbf{s}_{t,0}^{(\cdot)} + \lambda_0 \cdot \mathbf{A}_t \cdot \mathbf{e}^{(\cdot)}$ ;  
 $w_{t,1}^{(\cdot)} = \text{Weight}(\mathbf{s}_{t,1}^{(\cdot)})$ ;  
 //Gradient ascent on the posterior :  
 For  $i = 1 : I - 1$   
 $\mathbf{s}_{t,i+1}^{(\cdot)} = \text{Meanshift}(\{\mathbf{s}_{t,i}^{(\cdot)}, w_{t,i}^{(\cdot)}\})$ ; //Equation (6)  
 Draw  $\mathbf{e}^{(\cdot)} \sim N(\mathbf{0}, \mathbf{I}_{n_x})$ ;  
 $\mathbf{s}_{t,i+1}^{(\cdot)} = \mathbf{s}_{t,i+1}^{(\cdot)} + \lambda_i \cdot \mathbf{A}_t \cdot \mathbf{e}^{(\cdot)}$ ;  
 $w_{t,i+1}^{(\cdot)} = \text{Reweight}(\mathbf{s}_{t,i+1}^{(\cdot)})$ ; //Equation (7)  
 end  
 $\hat{\mathbf{x}}_t = \sum_{n=1}^N \mathbf{s}_{t,I}^{(n)} w_{t,I}^{(n)}$ ;  
 $\mathbf{s}_t^{(\cdot)} = \mathbf{s}_{t,J}^{(\cdot)}; w_t^{(\cdot)} = w_{t,J}^{(\cdot)}; (0 \leq J \leq I)$

---

and the posterior density is given by (1):

$$p(\mathbf{s}_t^{(n)} | \mathbf{Y}_t) \propto p(\mathbf{y}_t | \mathbf{s}_t^{(n)}) \cdot \sum_{l=1}^N p(\mathbf{s}_t^{(n)} | \mathbf{s}_{t-1}^{(l)}) w_{t-1}^{(l)}. \quad (9)$$

The second term on the right-hand side of (9) is a sample-based approximation of the prior density. When mean shift is performed on  $\mathbf{S}_t$ , the next generation of particles will concentrate more on the approximated density modes, but the weight will contain a factor that offsets this effect.

## 2.4 Implementation

A pseudo-code of the KPF algorithm at one time-step is shown in Table 1. Note that to prevent local plateaus in the density from stopping the gradient ascent too early, a small perturbation is added to the particles at each iteration. Rather than using a constant  $\lambda$ , it is more effective to use a wide kernel at first to smooth out certain weak modes in the PDF and gradually scale it down to move particles to the most dominant modes. In our experiments we set  $\lambda_i = \alpha^i \lambda_0$  where  $\alpha$  is a number empirically chosen to be 0.8.

The iterative mode-seeking produces a set of particles that are clustered around posterior modes. When multiple objects are present in a cluttered background, it is necessary to associate each mode with the correct object. In the next section we present a framework for tracking multiple objects using KPF.

## 3 Multiple Object Tracking

To differentiate a KPF tracking a single object and a KPF that tracks multiple modalities, we will call the latter MM-KPF, although both filters can share the same code. The

MM-KPF tracker starts with the same mode-seeking procedure used in KPF. The posterior modes are subsequently detected through a clustering algorithm. Since mean shift produces naturally clustered particles around posterior modes<sup>1</sup>, we choose an efficient basic sequential algorithmic scheme (BSAS) [19], in which each incoming data point either joins an existing cluster closest to it or forms a new cluster, depending on a distance threshold  $d_C$ . Fig. 2(a) shows two clustering results using BSAS. Once posterior modes are detected, a hypotheses generation and evaluation module is initiated to assign modes to objects.

### 3.1 Hypotheses Generation

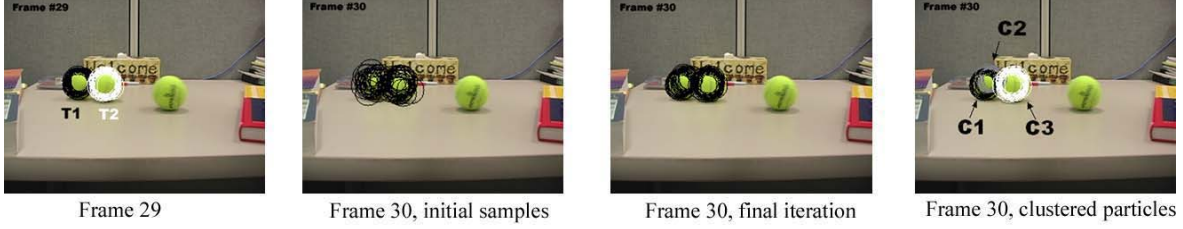
Suppose  $N_m$  posterior modes are detected at the current frame. We call the sequence of modes associated with one object a *track*. To associate  $N_m$  objects with  $N_o$  tracks, the following assumptions are made: (A1) Each object in the group generates one and only one mode in the posterior density. (A2) Each mode in the posterior originates from either an object or background clutter. It then follows  $N_m \geq N_o$ . Note that assumption (A1) may not be true in the case of severe occlusion. However removing (A1) will significantly increase the number of hypotheses. We choose to handle the case of  $N_m < N_o$  through other measures, as will be discussed in 3.3.

Denote the detected modes as  $\{C_k\}_{k=1}^{N_m}$  and tracks as  $\{T_j\}_{j=1}^{N_o}$ . MM-KPF generates multiple hypotheses by enumerating all possible associations for the current frame. Let  $\Theta$  denote a hypothesis containing a set of assignments, and let  $\theta^{kj}$  denote an assignment rule that associates the  $k$ -th mode with the  $j$ -th track. We have  $\Theta = \{\theta^{kj}\}$  with  $k \in [1, N_m]$  and  $j \in [0, N_o]$ , in which  $\theta^{k0}$  means the mode is associated with background clutter. A hypothesis can be represented by a hypothesis matrix in which detected posterior modes are represented by the rows and the known tracks by the columns. A nonzero element at matrix position  $(k, j)$  denotes that  $C_k$  is associated with track  $T_j$ . An additional column  $T_0$  is appended to the matrix denoting false alarms. The two assumptions (A1) and (A2) can be imposed by restricting an hypothesis matrix to have only a single nonzero value in any row or column, except for the first column. Fig. 2(b) shows a scenario where a tracker tracking two tennis balls detected 3 modes in the posterior. In this case, there are  $P_3^2 = 6$  possible hypothesis matrices.

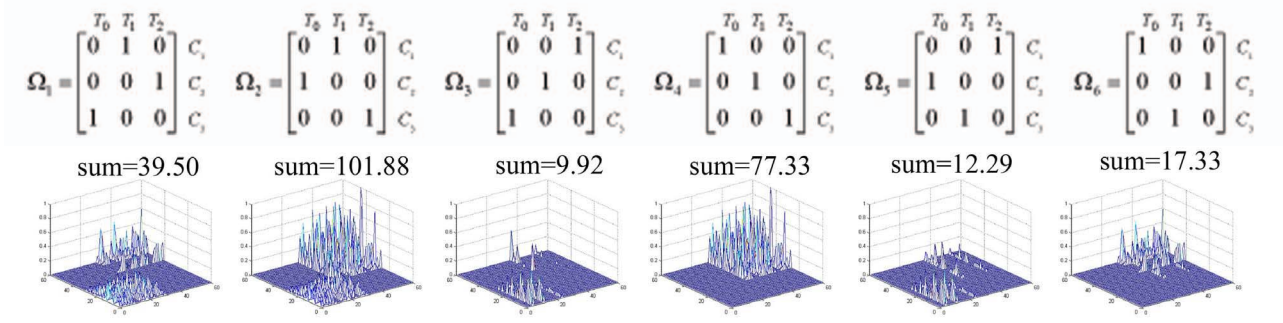
### 3.2 Hypotheses Evaluation

The fitness of  $\theta^{kj}$  can be computed by measuring the motion coherence between the  $k$ -th cluster at time  $t$  with its

<sup>1</sup>It was shown in [5] that iteration of mean shift gives rise to natural clustering algorithms in which sample points will merge into cluster centers.



(a) Particles in frame #29 and #30.



(b) The 6 hypothesis matrices and their corresponding MC matrices shown as  $60 \times 60 \times 1$  plots.

**Figure 2. Particle-based motion correspondence for TENNISBALLS sequence. In (b), the sum value on top of each plot is the summation of all elements of the corresponding MC matrix as defined in (15). In this example, the second hypothesis is the winning hypothesis.**

assumed source cluster at time  $t - 1$ . Based on this idea, we formulate a Maximum a Posteriori estimate of  $\Theta$ . Assuming

$$p(\Theta | \mathbf{Y}_t) = \prod_{\theta^{kj} \in \Theta} p(\theta^{kj} | \mathbf{Y}_t), \quad (10)$$

we have

$$\begin{aligned} \hat{\Theta}_{MAP} &= \arg \max_{\Theta} \prod_{\theta^{kj} \in \Theta} p(\theta^{kj} | \mathbf{Y}_t) \\ &= \arg \max_{\Theta} \prod_{\theta^{kj} \in \Theta} \int p(\theta^{kj}, \mathbf{x}_t | \mathbf{Y}_t) d\mathbf{x}_t \end{aligned} \quad (11)$$

The above equation can be approximated by replacing integral with summation:

$$\hat{\Theta}_{MAP} \approx \arg \max_{\Theta} \prod_{\theta^{kj} \in \Theta} \sum_n p(\theta^{kj}, \mathbf{s}_t^{(n)} | \mathbf{Y}_t). \quad (12)$$

When  $j = 0$ ,  $p(\theta^{kj}, \mathbf{s}_t^{(n)} | \mathbf{Y}_t)$  is the probability that the particle  $\mathbf{s}_t^{(n)}$  originates from clutter. In this work we simply let the probability be a constant for  $j = 0$ . This essentially means we have no prior information to believe that one hypothesis is better than another<sup>2</sup>. When  $j \neq 0$ ,

<sup>2</sup>When association is regarding low-level features such as edges and points, a prior distribution of  $\Theta$  may be derived given the clutter level of the background [11].

$p(\theta^{kj}, \mathbf{s}_t^{(n)} | \mathbf{Y}_t)$  is the probability that  $\mathbf{s}_t^{(n)}$  originates from cluster  $\mathbf{C}_t^k$  which is associated with track  $T_j$ . Let us assume the clusters at time  $t - 1$  have been properly re-labeled such that  $\mathbf{C}_{t-1}^j$  is associated with  $T_j$ . In this case the posterior can be evaluated as:

$$p(\theta_t^{kj}, \mathbf{s}_t^{(n)} | \mathbf{Y}_t) = \sum_l p(\mathbf{s}_t^{(n)} | \mathbf{s}_{t-1}^{(l)}) w_t^{(n)} w_{t-1}^{(l)} \cdot \delta_{t-1}^{jl} \cdot \delta_t^{kn}, \quad (13)$$

where  $j \neq 0$  and  $\delta_t^{kn}$  is the Kronecker  $\delta$  defined as

$$\delta_t^{kn} = \begin{cases} 1 & \mathbf{s}_t^{(n)} \text{ belongs to cluster } \mathbf{C}_t^k, \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

Equation (13) is similar to (9) except that it is non-zero only for particles that belongs to  $\mathbf{C}_t^k$  and the prior is only evaluated over particles that belongs to  $\mathbf{C}_{t-1}^j$ . This computation involves evaluation of the weight  $w_t^{(n)}$  and the transition kernel  $p(\mathbf{s}_t^{(n)} | \mathbf{s}_{t-1}^{(l)})$  for each particle, whose values have all been acquired in the re-weighting step of equation (7). The computation hence can be conveniently implemented by constructing a  $N \times N$  matrix  $\mathbf{P}$  where  $N$  is the number of particles, such that the element at position  $(n, l)$  is given by (9). Equation (13) can be conveniently implemented by masking and summing up elements of  $\mathbf{P}$ .

It is illuminating to look at how masked matrix  $\mathbf{P}$  reflects the ‘‘fitness’’ of each hypothesis. To do so, define for each

assignment an  $N \times N$  mask matrix  $\mathbf{M}(\theta^{kj})$  such that the element at position  $(n, l)$  is 1 if  $\mathbf{s}_t^{(n)} \in \mathbf{C}_t^k$  and  $\mathbf{s}_{t-1}^{(l)} \in \mathbf{C}_{t-1}^j$  and 0 otherwise. Let  $\tilde{\mathbf{P}}(\theta^{kj})$  denote the element-wise product of  $\mathbf{P}$  and  $\mathbf{M}(\theta^{kj})$ . We define for each hypothesis a *Motion Correspondence (MC) Matrix*  $\Psi(\Theta)$  as:

$$\Psi(\Theta) = \sum_{\theta \in \Theta} \tilde{\mathbf{P}}(\theta), \quad (15)$$

where for simplicity the superscripts of  $\theta^{kj}$  are omitted. The MC matrix can serve as a rough visual indicator of the “fitness” of each hypothesis. In Fig. 2(b), the 6 hypothesis matrices are shown for the TENNISBALLS sequence where 3 clusters are to be assigned to two tracks. The 6 MC matrices are plotted at the bottom of the figure, along with the sum of their elements. In this example, the winning hypothesis is the second hypothesis.

### 3.3 Implementation

An object interaction mechanism is used to detect if two objects are *close* and to adaptively switch between independent KPF trackers and a MM-KPF tracker: MM-KPF is only invoked when one object’s particle set reaches the  $\beta$ -cut covariance ellipse of other objects. Given object sizes and their dynamic covariance matrices, a distance threshold  $d_T$  can be calculated for each pair of objects. Let  $d$  denote the distance between a pair of objects. If  $d < d_T$  is true, we say the two objects are *close*. A *group*  $\mathcal{G}$  is a set of objects where for any  $\mathcal{O} \in \mathcal{G}$ :  $\mathcal{O}$  is close to at least one other object and any object close to  $\mathcal{O}$  is in  $\mathcal{G}$ . For each group, particles from its members will be combined into a single particle set and MM-KPF is invoked to perform the iterations outlined in TABLE 1. Before processing the next frame each group will be split back into individual particle sets by re-sampling for each object its associated cluster and its closest cluster that represents a false mode, if there is any. A total of  $N$  particles will be generated for each objects. There are several benefits of doing this. First, by propagating clusters individually each object is allowed to have its own motion model. Secondly, a cluster with only a few particles but associated with an object can have more than a few “children”, reducing the risk of losing one object mode. Finally, propagating by cluster ensures that when groups are formed, most particles generated for one object are initially stored next to each other in computer memory. This reduces possible errors made by the BSAS clustering algorithm where the order in which data is presented may lead to different clustering results. Note that the threshold  $d_T$  also provides a clue for automatically deciding the clustering threshold  $d_C$ . It is found that in general  $\frac{1}{3}d_T \leq d_C \leq \frac{1}{2}d_T$  is a good choice.

The MM-KPF does not try to resolve the occlusion order during occlusion, but instead focuses on how to resolve

the correspondence ambiguity *after* occlusion. To do so it needs to be aware of two exceptions caused by severe occlusion: (1) When  $N_m < N_o$ , it usually means severe occlusion has occurred. (2) It is possible that occlusion has occurred, but the tracker detects  $N_m \geq N_o$ . A winning hypothesis with a very small score (e.g., out of the  $3.5\sigma$  region of previous scores) usually accompanies the occurrence of the latter case. The principle employed in MM-KPF is that the tracker should avoid blindly trusting the clustering result in occlusion and should keep looking for the objects according to their motion model. Therefore, when MM-KPF detects the two exceptions, the tracker will temporarily use particles before the first iteration ( $i = 0$ ) for estimation and will propagate these particles with equal probability, until it detects neither situation is true. This makes the tracker actively look for the occluded object and latch onto it once the object reemerges. We have found that this works effectively for most video sequences with short full occlusion where objects do not change their motion pattern dramatically during occlusion.

## 4 Experiments

Comparison of KPF with the standard PF and Regularised PF [14] using simulated data on single object tracking is reported in [4]. In this paper we report experimental results on real video for both single and multiple object tracking. The observation model used in the experiments is determined by several factors based on color, gradient, Principle Component Analysis (PCA), and motion. For the color module  $p_c$ , a cosine similarity is computed between each hypothesis histogram and a model histogram obtained prior to tracking. The gradient factor  $p_g$  is determined by the average pixel values of the gradient map over the contour of the model. The PCA factor  $p_p$  employs PCA analysis in feature space and is used to track multiple faces. In the motion factor  $p_m$ , a background subtraction algorithm [18] is used when tracking objects in distance. The KPF algorithm is applied to track moving objects in various videos containing both indoor and outdoor scenes. We now present results obtained using some of the test videos as listed in Table 2 (Test videos are available on the author’s website).

**Table 2. Test videos**

Videos	Single Object		Multiple Objects	
	1FACE	FIGURESKATE	3FACES	PLAZA
Frames	797	120	654	458
Size	180x120	176x120	200x150	320x240
FPS	29	29	25	30
$p(\mathbf{x}_t   \mathbf{Y}_t)$	$p_c \cdot p_g$	$p_c \cdot p_g$	$p_c \cdot p_p$	$p_m$
Particles	30	30	30	50
Iteration( $I$ )	3	3	3	3



Figure 3. Tracking a skater in FIGURESKATE sequence.

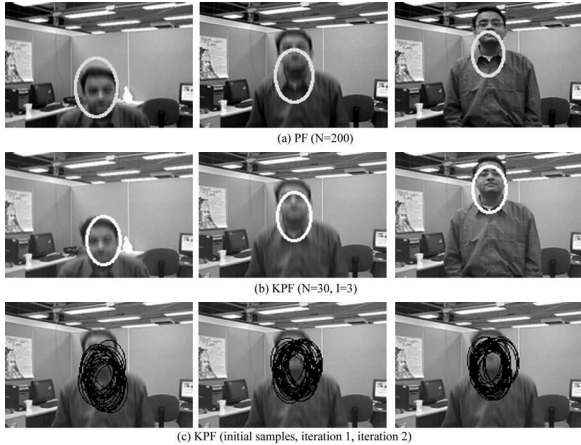


Figure 4. Results of frame #24, #27, and #31 from sequence 1FACE.

#### 4.1 Single Object Tracking

KPF is applied to head tracking to test its ability in tracking with a weak dynamic model. The test videos involve various motions such as sudden acceleration, rotation, abrupt changes of direction, jump, and out-of-plane rotation. The first test video sequence, 1FACE, consists of 797 frames of a human face moving in a typical laboratory environment. The face is modeled as an ellipse with a vertical major axis and a fixed aspect ratio of 1.4. Trackers are initialized manually. A few frames of the tracking results using PF and KPF are shown in Fig. 4. The PF tracker with the same dynamic model tends to lag behind the object and eventually loses the head at the #373rd frame. A PF tracker is able to succeed after doubling the dynamic noise and uses 250 particles to saturate the search region. On the other hand, KPF with 30 particles and 3 iterations, despite being occasionally distracted by the background clutter, is able to track the face throughout the sequence. Fig. 4(c) shows the intermediate samples of KPF in processing frame #27. Starting from a poor prediction, the KPF is able to move particles toward the correct direction through mean shift iteration and gives a better estimate. In the FIGURESKATE sequence shown in Fig. 3, the KPF successfully tracks the female skater's head as she performs out-of-plane rotation

followed by a sudden acceleration.

#### 4.2 Multiple Object Tracking

The first test sequence for multiple object is the TENNISBALLS sequence containing 63 frames of size  $320 \times 240$ . The sequence includes full occlusion of two balls by the third ball. Two balls then stay *close* from frame 26 to 63, a duration long enough to distract independent PF trackers. An MM-KPF is configured to track all three balls in a single-object state space with a constant velocity motion model. The result is compared with that of using 3 independent PF trackers with the same motion model. It is found that PF had difficulty in robustly recovering from occlusion. Even when PF did survive occlusion, it eventually failed when two balls were close, as seen in Fig. 1. On the other hand, the MM-KPF tracker was able to robustly handle occlusion and kept three modes from frame 26 to frame 63. Clustered samples from a few frames are shown in Fig. 5.

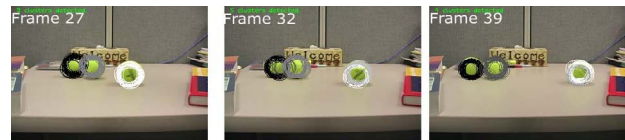
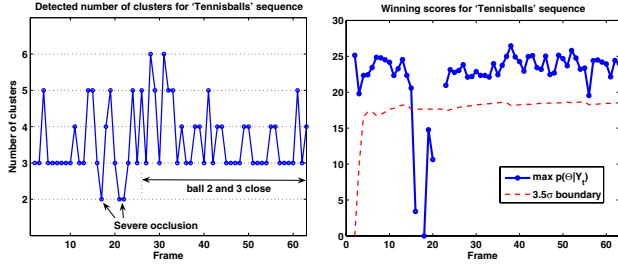


Figure 5. Clustered particles in MM-KPF for TENNISBALLS.

A few typical clustering results during occlusion are shown in Fig. 6. Fig. 7 shows the number of detected clusters in KPF. By detecting severe occlusions, the KPF tracker is able to handle occlusion more robustly.



Figure 6. A few typical cluster results in occlusion (zoomed in). (a)  $N_m < N_o$ . (b)  $N_m \geq N_o$ . (c)  $N_m \geq N_o$  with a low score.



**Figure 7. The detected number of clusters and winning hypothesis scores for the TENNISBALLS sequence.**

Fig. 8 shows frames from the 3FACES sequence and the PLAZA sequence. The 3FACES sequence contains 3 human faces moving in a laboratory. The PLAZA sequence contains multiple pedestrians walking in a campus environment. In both sequences objects are modeled as a rectangle with a variable width and a fixed aspect ratio. Object interaction reasoning is used to switch between independent KPF trackers with random walk models and MM-KPF trackers assuming constant acceleration. Two objects detected to be close are connected by a line in Fig. 8. The tracker is able to dynamically update the object relationship and perform correct data association. The bottom row of Fig. 8 shows the tracker tracking 9 pedestrians simultaneously. Note how two of the pedestrians (marked by an arrow) are tracked as a group for as long as 7 seconds as they form and remove group relationship with other pedestrians.

### 4.3 Computational Cost and Accuracy

KPF improves filtering performance at the cost of introducing extra computation. Equations (6), (8) and (9) all require  $O(N^2)$  function evaluations. However, it is possible to greatly reduce the computation complexity to  $O(N)$  by seeking suboptimal results. Note that the kernel evaluations in (6) and (8) only need to be computed once for every iteration if we use a normal kernel such that  $H_\lambda = K_\lambda$  (the derivative of a normal profile remains normal). Also, the summation of  $N$  terms in the two equations may be reduced to fewer terms by summing over samples with the most significant weights. In our simulations it was possible to reduce the summation to 10 terms from  $N = 200$  without affecting the results significantly. A similar reduction can be made to (9). Computations can be further reduced when accurate initialization is available and the observation model is robust, in which case the motion model may be relaxed by skipping the re-weighting step and mean shift will find likelihood modes. On the other hand, since a MM-KPF tracker tracking two objects does not expand the state space,

**Table 3. Normalized computation time of the TENNISBALLS and the 1FACE sequences**

	Meanshift	Reweight	MR	Weight <sup>a</sup>
TENNISBALLS	$\frac{1}{10}$	$\frac{1}{5}$	$\frac{1}{3.3}$	1
1FACE	$\frac{1}{15}$	$\frac{1}{9}$	$\frac{1}{5.6}$	1

<sup>a</sup>‘MR’ is the summation of ‘Meanshift’ and ‘Reweight’. All shown numbers are obtained by performing the corresponding operation on all particles once.

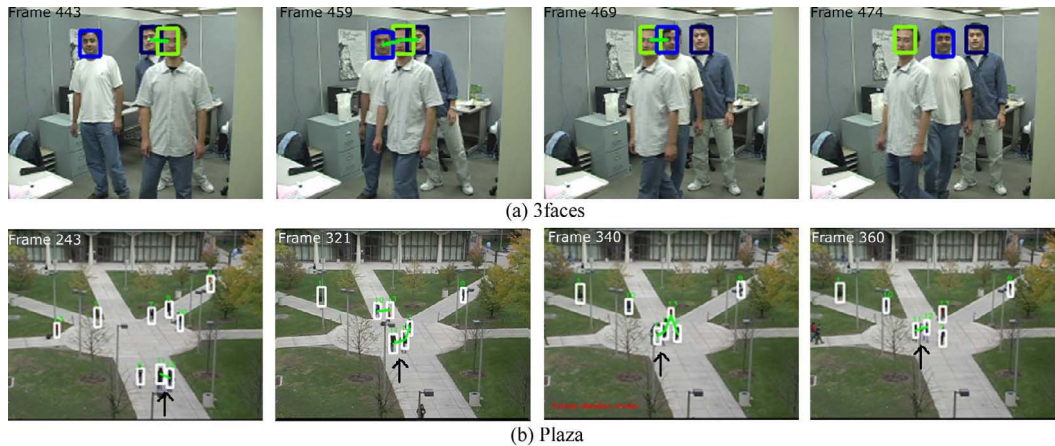
it is possible to reduce the total number of particles by up to 1/2 when joining multiple particle sets into one.

Table 3 shows the time allocation in tracking the TENNISBALLS and 1FACE sequence by KPF. The time spent on hypotheses generation and evaluation only takes a fraction of the total computation time (usually less than 1%). Over 50% of the computation time is spent on feature extraction and sample weighting. Therefore in visual tracking, KPF can actually improve the filter efficiency by reducing weight computation. Let us assume  $N_{pf}$  particles are used in PF and  $N_{kpf}$  in each iteration of KPF. If each sample weighting takes  $T_w$  seconds and each mean shift and re-weighting takes  $T_{mr}$ , then  $N_{pf} \cdot T_w$  seconds are used by PF for each frame and  $N_{kpf} I \cdot T_w + (I - 1) N_{kpf} \cdot T_{mr}$  by a KPF with  $I$  iterations. Suppose on average a tracker switching between KPF and MM-KPF requires  $T_{mr} \approx T_w/4$ . In this case, KPF will be less time-consuming than PF as long as  $N_{kpf} < \frac{1}{3.5} N_{pf}$ , assuming  $I = 3$ .

## 5 Discussion

A kernel-based particle filter (KPF) is introduced in this paper. The new particle filter invokes kernels to form a continuous estimate of the posterior density function. This representation allows direct estimation of the posterior gradient and efficient allocation of particles. A data association approach employing intermediate results in mode-seeking enables the KPF to simultaneously track multiple objects with partial occlusion and short full occlusion. Experiments show KPF perform robust visual tracking with improved sampling efficiency.

Future work may address situations when the motion pattern of objects within one group changes dramatically, especially during long and continuous full occlusion, in which case the proposed association technique may fail. It is possible to improve the tracker by forcing a larger dynamic noise and keeping all detected modes during occlusion. However, we believe the problem is inherent to any purely motion-based association techniques and a more robust solution would be to employ both motion continuity and appearance



**Figure 8. Results on the 3FACES and PLAZA using KPF with object interaction reasoning. Two objects detected to be close to each other are marked by a line connecting the objects.**

difference between objects. But then, when different observation models can be used for different objects, these objects may not be tracked in a single-object state space by a single particle filter.

## Acknowledgments

The work is supported in part by the National Science Foundation under the grant BCS-9980054 and CCR 0196365, and by UIC 2004 Provost Graduate Research Award. The authors would like to thank Ahmet Murat Bagci at UIC for providing routines for background subtraction.

## References

- [1] Y. Bar-Shalom and A. G. Jaffer. *Tracking and Data Association*. Academic Press, San Diego, CA, 1988.
- [2] T. Cham and J. M. Rehg. A multiple hypothesis approach to figure tracking. In *IEEE CVPR*, June 1999.
- [3] C. Chang and R. Ansari. Kernel particle filter: iterative sampling for efficient visual tracking. In *IEEE ICIP*, 2003.
- [4] C. Chang and R. Ansari. Kernel particle filter for visual tracking. *IEEE Signal Processing Lett.*, 12(3):242 – 245, March 2005.
- [5] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. PAMI*, 17(8):790–799, 1995.
- [6] K. Choo and D. J. Fleet. People tracking using hybrid monte carlo filtering. In *IEEE ICCV*, 2001.
- [7] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. PAMI*, 25(5):564–577, May 2003.
- [8] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE CVPR*, 2000.
- [9] B. Han, D. Comaniciu, Y. Zhu, and L. Davis. Incremental density approximation and kernel-based bayesian filtering for object tracking. In *IEEE CVPR*, Jun. 2004.
- [10] C. Hue, J. L. Cadre, and P. Prez. Sequential monte carlo methods for multiple target tracking and data fusion. *IEEE Trans. Signal Processing*, 50(2), Feb. 2002.
- [11] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *ECCV*, 1996.
- [12] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *IEEE ICCV*, 1999.
- [13] J. MacCormick and M. Isard. Bramble: A bayesian multiple-blob tracker. In *IEEE ICCV*, 2001.
- [14] C. Musso, N. Oudjane, and F. LeGland. Improving regularised particle filters. In A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.
- [15] C. Rasmussen and G. D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE Trans. PAMI*, 23(6):560–576, Jun 2001.
- [16] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.
- [17] C. Sminchisescu and B. Triggs. Hyperdynamics importance sampling. In *ECCV*, volume 1, pages 769–783, 2002.
- [18] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE CVPR*, 1999.
- [19] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 1999.
- [20] J. Vermaak, A. Doucet, and P. Pérez. Maintaining multimodality through mixture tracking. In *IEEE ICCV*, 2003.
- [21] Y. Wu, T. Yu, and G. Hua. Tracking appearances with occlusions. In *IEEE CVPR*, Madison, Wisconsin, June 2003.
- [22] T. Yu and Y. Wu. Collaborative tracking of multiple targets. In *IEEE CVPR*, Washington, D.C., June 2004.