Maintaining Multi-Modality through Mixture Tracking

Jaco Vermaak, Arnaud Doucet
Cambridge University Engineering Department
Cambridge, CB2 1PZ, UK

Patrick Pérez
Microsoft Research
Cambridge, CB3 0FB, UK

Abstract

In recent years particle filters have become a tremendously popular tool to perform tracking for non-linear and/or non-Gaussian models. This is due to their simplicity, generality and success over a wide range of challenging applications. Particle filters, and Monte Carlo methods in general, are however poor at consistently maintaining the multi-modality of the target distributions that may arise due to ambiguity or the presence of multiple objects. To address this shortcoming this paper proposes to model the target distribution as a non-parametric mixture model, and presents the general tracking recursion in this case. It is shown how a Monte Carlo implementation of the general recursion leads to a mixture of particle filters that interact only in the computation of the mixture weights, thus leading to an efficient numerical algorithm, where all the results pertaining to standard particle filters apply. The ability of the new method to maintain posterior multi-modality is illustrated on a synthetic example and a real world tracking problem involving the tracking of football players in a video sequence.

1. Introduction

Tracking involves the detection and recursive localisation of an object or objects of interest based on sequential data measurements. Typical examples include face tracking in video sequences [6], tracking aircraft using radar returns [2], localising a mobile robot using laser range measurements [12], and many more. In practical settings there are many factors that contribute towards the uncertainty in an object's exact location and configuration. These include measurement noise, inaccurate modelling, clutter (false positives), fluctuations in environmental conditions, *etc.* To adequately capture the uncertainty due to these factors a probabilistic framework is required.

Within a tracking context one particularly popular approach is Bayesian Sequential Estimation. This framework allows the recursive estimation of a time-evolving posterior distribution that describes the object state conditional on all the observations seen so far, commonly known as the filtering distribution. It requires the definition of a Markovian dynamic model that describes how the object state evolves, and a model to evaluate the likelihood of a hypothesised state giving rise to the observed data. This, in theory, is sufficient to allow recursive estimation of the filtering distribution. However, the likelihood models for tracking often lead to intractable inference, requiring approximation techniques.

In recent years Sequential Monte Carlo Estimation, otherwise known as Particle Filtering [4], has proved to be a popular approximation methodology. Its popularity stems from its simplicity, generality and success over a wide range of challenging applications. It represents the filtering distribution with a set of samples, or particles, and associated importance weights, which are then propagated through time to give approximations of the filtering distribution at subsequent time steps. It requires only the definition of a suitable proposal distribution from which new particles can be simulated, and the ability to evaluate the likelihood and dynamic models.

One important shortcoming of particle filters, and Monte Carlo methods in general, is that they are poor at consistently maintaining the multi-modality in the target distribution. Multiple modes arise if there is ambiguity about the object state due to insufficient measurements or clutter, or if the measurements come from multiple objects. In the first case it is desirable to track all the modes until the ambiguity can be naturally resolved, and in the second, it is often required to track all the objects present. In a practical particle filter implementation, however, it often happens that all the particles quickly migrate to one of the modes, subsequently discarding all other modes.

This paper introduces a strategy that is better able to maintain multi-modality. Working from the assumption that mixture models are inherently more effective at capturing multiple modes, the target distribution is formulated as a non-parametric mixture of filtering distributions. As is the case for Bayesian sequential estimation a general framework is derived in which the mixture filtering distribution

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 **IEEE**

can be computed recursively in two steps: a prediction step, followed by an update step when the new data becomes available. A Monte Carlo implementation of the general framework essentially leads to a mixture of particle filters that interact only in the computation of the mixture weights. This result is elegant in the sense that all the results that hold for standard particle filters transfer to the individual mixture components. Furthermore, the approach can be combined with any convenient strategy to obtain and update the mixture representation.

The remainder of the paper is organised as follows. Section 2 introduces the general mixture tracking framework. Section 3 shows how a Monte Carlo implementation of the general framework leads to a mixture of particle filters. Section 4 discusses some important issues regarding the initialisation and updating of the mixture representation. In Section 5 the proposed algorithm is compared to the standard particle filter on two problems. The first is a synthetic example where the multi-modality is due to ambiguity, whereas the second is a real world problem involving the tracking of multiple football players in a video sequence. The paper is summarised in Section 6.

Related Work

The weakness of particle methods to maintain multimodality has been acknowledged before. The authors in [9] introduce the idea of clustered particle filtering to guard against samples sets becoming prematurely impoverished in the context of mobile robot localisation in highly symmetric environments. The algorithm essentially groups the particles into clusters that are independently tracked. Each cluster is assigned a probability that is tracked using another Monte Carlo filter operating at a higher level. The cluster with the highest probability at any particular time step is deemed to correspond to the true robot configuration at that instant.

In the special context of multi-object tracking a vast body of literature exists. However, most algorithms broadly fall into one of two categories. The first builds multi-object trackers by multiple instantiations of single object tracking algorithms, *e.g.* [3, 11]. Strategies with various levels of sophistication have been developed to interpret the output of the resulting trackers in the case of occlusions and overlapping objects. The second category of multi-object trackers explicitly extends the state-space to include components for all the objects of interest, *e.g.* [7, 8]. A variable number of objects can be accommodated by either dynamically changing the dimension of the state-space, or by a corresponding set of indicator variables signifying whether an object is present or not.

2. Mixture Tracking

Let $\mathbf{x}_t$ denote the state of the object of interest, and $\mathbf{y}^t = (\mathbf{y}_1 \cdots \mathbf{y}_t)$ the observations up to time t. For tracking the distribution of interest is the filtering distribution $p(\mathbf{x}_t|\mathbf{y}^t)$. In Bayesian sequential estimation this distribution can be computed using the two step recursion:

$$\textbf{predict}: \quad p(\mathbf{x}_t|\mathbf{y}^{t-1}) = \int D(\mathbf{x}_t|\mathbf{x}_{t-1})p(d\mathbf{x}_{t-1}|\mathbf{y}^{t-1}) \tag{1}$$

$$\textbf{update}: \quad p(\mathbf{x}_t|\mathbf{y}^{t}) = \frac{L(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}^{t-1})}{\int L(\mathbf{y}_t|\mathbf{s}_t)p(d\mathbf{s}_t|\mathbf{y}^{t-1})}, \tag{2}$$

where the prediction distribution follows from marginalisation, and the new filtering distribution is a direct consequence of Bayes' rule. The recursion requires the specification of a dynamic model describing the state evolution $D(\mathbf{x}_t|\mathbf{x}_{t-1})$, and a model that gives the likelihood of any state in the light of the current observation $L(\mathbf{y}_t|\mathbf{x}_t)$. The recursion is initialised with some initial distribution $p(\mathbf{x}_0)$.

To capture multi-modality this paper formulates the filtering distribution as a M-component mixture model, *i.e.*

$$p(\mathbf{x}_t|\mathbf{y}^t) = \sum_{m=1}^{M} \pi_{m,t} p_m(\mathbf{x}_t|\mathbf{y}^t), \tag{3}$$

with $\sum_{m=1}^{M} \pi_{m,t} = 1$. Note that no parametric model is assumed for the individual mixture components. The remainder of this section shows how this non-parametric mixture representation can be updated recursively in the same fashion as the two step approach for standard Bayesian sequential estimation.

Assuming that the mixture filtering distribution $p(\mathbf{x}_{t-1}|\mathbf{y}^{t-1})$ is known, the new prediction distribution is obtained by substitution into (1), leading to

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}^{t-1}) &= \sum_{m=1}^{M} \pi_{m,t-1} \int D(\mathbf{x}_t|\mathbf{x}_{t-1}) p_m(d\mathbf{x}_{t-1}|\mathbf{y}^{t-1}) \\ &= \sum_{m=1}^{M} \pi_{m,t-1} p_m(\mathbf{x}_t|\mathbf{y}^{t-1}), \end{aligned}$$

with $p_m(\mathbf{x}_t|\mathbf{y}^{t-1}) = \int D(\mathbf{x}_t|\mathbf{x}_{t-1}) p_m(d\mathbf{x}_{t-1}|\mathbf{y}^{t-1})$ the prediction distribution for the m-th component. Thus the new prediction distribution is straightforwardly obtained by computing the prediction distribution for each of the components individually, and combining them in a mixture that retains the original component weights.

To obtain the new filtering distribution the prediction distribution is substituted into (2), leading to

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}^t) &= \frac{\sum_{m=1}^{M}\pi_{m,t-1}L(\mathbf{y}_t|\mathbf{x}_t)p_m(\mathbf{x}_t|\mathbf{y}^{t-1})}{\sum_{n=1}^{M}\pi_{n,t-1}\int L(\mathbf{y}_t|\mathbf{s}_t)p_n(d\mathbf{s}_t|\mathbf{y}^{t-1})} \\ &= \sum_{m=1}^{M}\Big[\frac{\pi_{m,t-1}\int L(\mathbf{y}_t|\mathbf{s}_t)p_m(d\mathbf{s}_t|\mathbf{y}^{t-1})}{\sum_{n=1}^{M}\pi_{n,t-1}\int L(\mathbf{y}_t|\mathbf{s}_t)p_n(d\mathbf{s}_t|\mathbf{y}^{t-1})}\Big] \\ &\quad\times\Big[\frac{L(\mathbf{y}_t|\mathbf{x}_t)p_m(\mathbf{x}_t|\mathbf{y}^{t-1})}{\int L(\mathbf{y}_t|\mathbf{s}_t)p_m(d\mathbf{s}_t|\mathbf{y}^{t-1})}\Big]. \end{aligned}$$

In the second line the second term in brackets is easily recognised as the new filtering distribution for the m-th component, *i.e.*

$$p_m(\mathbf{x}_t|\mathbf{y}^t) = \frac{L(\mathbf{y}_t|\mathbf{x}_t)p_m(\mathbf{x}_t|\mathbf{y}^{t-1})}{\int L(\mathbf{y}_t|\mathbf{s}_t)p_m(d\mathbf{s}_t|\mathbf{y}^{t-1})}.$$

The first term in brackets is independent of the state $\mathbf{x}_t$, and if this is taken to be the new weight, *i.e.*

$$\begin{aligned} \pi_{m,t} &= \frac{\pi_{m,t-1}\int L(\mathbf{y}_t|\mathbf{s}_t)p_m(d\mathbf{s}_t|\mathbf{y}^{t-1})}{\sum_{n=1}^{M}\pi_{n,t-1}\int L(\mathbf{y}_t|\mathbf{s}_t)p_n(d\mathbf{s}_t|\mathbf{y}^{t-1})} \\ &= \frac{\pi_{m,t-1}p_m(\mathbf{y}_t|\mathbf{y}^{t-1})}{\sum_{n=1}^{M}\pi_{n,t-1}p_n(\mathbf{y}_t|\mathbf{y}^{t-1})}, \end{aligned} \qquad (4)$$

the new filtering distribution is again a mixture of the individual component filtering distributions, as in (3). This is an elegant result indeed and means that the filtering recursion can be performed for each component individually. The correct target distribution is maintained as long as the mixture weights are updated according to (4). The new component weight is the normalised weighted likelihood for the component. This is the only part of the procedure where the components interact.

3. Particle Approximation

The general mixture tracking recursion introduced in the previous section yields closed-form expressions in only a small number of cases. A notable example occurs if both the dynamic and likelihood models are linear and Gaussian, resulting in a mixture of Kalman filters [1]. For models that are non-linear and/or non-Gaussian approximation techniques are required.

One popular approximation strategy is Sequential Monte Carlo methods [4], otherwise known as Particle Filters. These have gained tremendous popularity in recent years as a numerical approximation strategy for complex models. This is due to their simplicity, generality and modelling success over a wide range of challenging applications. The particle filter is a Monte Carlo method that represents the target distribution with a weighted set of samples that are propagated in such a manner so as to maintain a properly weighted sample from the target distribution at subsequent time steps. The remainder of this section presents the details of a particle approximation to the general mixture tracking recursion.

In what follows let $\mathcal{P}_t = \{N, M, \Pi_t, \mathcal{X}_t, \mathcal{W}_t, \mathcal{C}_t\}$ denote the particle representation of the mixture filtering distribution in (3), with N the number of particles, M the number of mixture components, $\Pi_t = \{\pi_{m,t}\}_{m=1}^{M}$ the mixture component weights, $\mathcal{X}_t = \{\mathbf{x}_t^{(i)}\}_{i=1}^{N}$ the particles, $\mathcal{W}_t = \{w_t^{(i)}\}_{i=1}^{N}$ the particle weights, and $\mathcal{C}_t = \{c_t^{(i)}\}_{i=1}^{N}$ the component indicators, *i.e.* $c_t^{(i)} \in \{1\cdots M\}$, with $c_t^{(i)} = m$ if particle i belongs to mixture component m. The particle representation implies a Monte Carlo approximation of the mixture filtering distribution of the form

$$\overline{p}(\mathbf{x}_t|\mathbf{y}^t) = \sum_{m=1}^{M}\pi_{m,t}\sum_{i\in\mathcal{I}_m} w_t^{(i)}\delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t),$$

where $\delta_a(\cdot)$ is the Dirac delta measure with mass at a, and $\mathcal{I}_m = \{i \in \{1\cdots N\} : c_t^{(i)} = m\}$ is the set of indices of the particles belonging to the m-th mixture component. Note that the mixture component weights and the particle weights for each mixture component sum to one, *i.e.* $\sum_{m=1}^{M}\pi_{m,t} = 1$ and $\sum_{i\in\mathcal{I}_m} w_t^{(i)} = 1, m = 1\cdots M$.

Given a particle set $\mathcal{P}_{t-1}$ that is approximately distributed according to $p(\mathbf{x}_{t-1}|\mathbf{y}^{t-1})$, the objective is to compute the new particle set $\mathcal{P}_t$ such that it is a sample set from $p(\mathbf{x}_t|\mathbf{y}^t)$. Recall that in the general mixture tracking recursion each mixture component evolves independently, and that the mixture components interact only in the computation of the mixture weights. In the same way the particle representations for the mixture components also evolve independently. Considering the m-th component, the samples $\{\mathbf{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i\in\mathcal{I}_m}$ is a properly weighted sample set from $p_m(\mathbf{x}_{t-1}|\mathbf{y}^{t-1})$. New samples are generated from a suitably chosen proposal distribution, which may depend on the old state and the new measurement, *i.e.* $\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)$, $i \in \mathcal{I}_m$. To maintain a properly weighted sample set the new particle weights are set to

$$w_t^{(i)} = \frac{\widetilde{w}_t^{(i)}}{\sum_{j\in\mathcal{I}_m}\widetilde{w}_t^{(j)}}, \quad \widetilde{w}_t^{(i)} = \frac{w_{t-1}^{(i)}L(\mathbf{y}_t|\mathbf{x}_t^{(i)})D(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)}.$$

The new sample set $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i\in\mathcal{I}_m}$ is then approximately distributed to $p_m(\mathbf{x}_t|\mathbf{y}^t)$.

To obtain the new mixture weights it is necessary to compute the component likelihoods $p_m(\mathbf{y}_t|\mathbf{y}^{t-1})$, $m = 1\cdots M$. Using the particles a Monte Carlo approximation to the m-th component likelihood can be obtained as

$$p_m(\mathbf{y}_t|\mathbf{y}^{t-1}) = \iint L(\mathbf{y}_t|\mathbf{x}_t)D(d\mathbf{x}_t|\mathbf{x}_{t-1})p_m(d\mathbf{x}_{t-1}|\mathbf{y}^{t-1})$$

$$\approx \sum_{i\in\mathcal{I}_m} w_{t-1}^{(i)} \frac{L(\mathbf{y}_t|\mathbf{x}_t^{(i)})D(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)},\mathbf{y}_t)} = \sum_{i\in\mathcal{I}_m} \widetilde{w}_t^{(i)}.$$

Substituting this result into (4) leads to an approximation for the new mixture weights given by

$$\pi_{m,t} \approx \frac{\pi_{m,t-1}\widetilde{w}_{m,t}}{\sum_{n=1}^{M}\pi_{n,t-1}\widetilde{w}_{n,t}}, \quad \widetilde{w}_{m,t} = \sum_{i\in\mathcal{I}_m}\widetilde{w}_t^{(i)}.$$

From time to time it is necessary to resample the particles to avoid degeneracy of the weights (see [5] for more details on degeneracy and resampling procedures). Standard resampling, however, is insensitive to the locations of the particles, and may lead to a loss of support in the target distribution. A very important point to note here is that the mixture modelling approach allows independent resampling of each of the mixture components according to the component particle weights, thus naturally preserving the support of the posterior distribution. Following such a procedure the new particle weights become $w_t^{(i)} = 1/|\mathcal{I}_m|$, $i \in \mathcal{I}_m$, where $|\cdot|$ denotes the set size operator.

4. Mixture Computation

The discussion up to this point showed how a mixture representation for the filtering distribution can be propagated, once such a representation is available. So far nothing has been said about how to obtain and maintain the mixture representation. In the ideal case there would be one mixture component for each of the modes in the target distribution. In practice, however, the number of modes in the target distribution is rarely known beforehand. Furthermore, the number of modes is unlikely to remain fixed, but may fluctuate as ambiguities arise and are resolved, or objects appear and disappear.

Thus, from time to time it is necessary to recompute the mixture representation to take account of these fluctuations. For example, it may be desirable to merge components that have a significant degree of overlap, and split components that have become too diffuse. Fortunately this is easy to achieve using the particle representation.

Denote by $(\mathcal{C}'_t, M') = \mathcal{F}(\mathcal{X}_t, \mathcal{C}_t, M)$ a spatial reclustering procedure. It takes as inputs the particles and the current mixture representation (component indicators and number of components), and computes a new mixture representation that may or may not have the same number of components as the original representation. Such a function encapsulates any mixture computation operation of interest, including merging, splitting, reclustering *etc.* What remains is to compute the new mixture and particle weights, Π'_t and $\mathcal{W}'_t$, so that the new mixture approximation $\mathcal{P}'_t = \{N, M', \Pi'_t, \mathcal{X}_t, \mathcal{W}'_t, \mathcal{C}'_t\}$ is equal in distribution to $\mathcal{P}_t$. These are straightforwardly obtained by developing the mixture representation for $\mathcal{P}_t$ as follows:

$$\overline{p}(\mathbf{x}_t|\mathbf{y}^t) = \sum_{m=1}^{M}\pi_{m,t}\sum_{i\in\mathcal{I}_m} w_t^{(i)}\delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t)$$

$$= \sum_{i=1}^{N}\pi_{c_t^{(i)},t}w_t^{(i)}\delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t) = \sum_{m=1}^{M'}\sum_{i\in\mathcal{I}'_m}\pi_{c_t^{(i)},t}w_t^{(i)}\delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t)$$

$$= \sum_{m=1}^{M'}\pi'_{m,t}\sum_{i\in\mathcal{I}'_m} w_t'^{(i)}\delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t),$$

where the new mixture and particle weights are given by

$$\pi'_{m,t} = \sum_{i\in\mathcal{I}'_m}\pi_{c_t^{(i)},t}w_t^{(i)}, \quad w_t'^{(i)} = \frac{\pi_{c_t^{(i)},t}w_t^{(i)}}{\pi'_{c_t'^{(i)},t}}.$$

With these weights the recomputed mixture $\mathcal{P}'_t$ represents exactly the same distribution as $\mathcal{P}_t$, and can be substituted for $\mathcal{P}_t$ without affecting the convergence properties of the particle filter. Note that the particles $\mathcal{X}_t$ are not affected in the new representation.

The reclustering function $\mathcal{F}$ can be implemented in any convenient way. For the applications in this paper the initial mixture representation is obtained by K-means clustering of the initial sample set, which is simulated from some initial state distribution $p(\mathbf{x}_0)$. At each iteration the mixture representation is recomputed by merging clusters with significant overlap, and splitting clusters that have become too diffuse. Following this the new mixture components are refined by a run of the K-means algorithm, initialised with the components obtained after the merging and splitting procedures. This simple strategy that allows both the mixture composition and the number of mixture components to vary was found to work well in the applications considered here, as illustrated in the following section.

5. Experiments and Results

This section compares the performance of the proposed mixture particle filter with that of the standard particle filter on two multi-modal tracking problems. The first is a synthetic example where the multimodality is due to ambiguity. The second is a real world problem involving the tracking of multiple football players in a video sequence.

5.1. Synthetic Example

The purpose of this example is to establish a baseline performance comparison between the standard and mixture particle filters on a problem where the ground truth is known. The model considered is scalar, and the governing equations are given by

$$D(x_t|x_{t-1}) = \mathcal{N}(x_t|x_{t-1}, \sigma_x^2)$$
$$L(y_t|x_t) = \mathcal{N}(y_t|x_t^2, \sigma_y^2),$$

where $\mathcal{N}(\cdot|\mu, \sigma^2)$ denotes the univariate Gaussian distribution with mean μ and variance σ^2. In the results reported here the parameters were set to $\sigma_x = \sigma_y = 0.1$. The symmetry of the Gaussian random walk dynamics and the quadratic term in the likelihood means that the filtering distribution has two modes of equal mass[1] at $\pm x_t^*$, with x_t^* the true state. An exception occurs when $|x_t^*|$ is close to zero, in which case the two modes merge, resulting in a single mode at zero.

Figure 1 shows some synthetic data for 100 time steps. Note that the true state was not simulated from the model, but deterministically generated to be sinusoidal. The ability of the standard and mixture particle filters to maintain the ambiguity was tested on this data set. In both cases the initial particles were uniformly generated, and the particle proposal was taken to be the dynamics, so that the new particle weights become proportional to the old weights multiplied by the corresponding likelihoods. The mixture particle filter was constrained to have a maximum of two components.

The comparative results for a typical run with 100 particles are given in Figure 1. The standard particle filter loses track of the true mode after time step 60. This behaviour is common in particle filters, and Monte Carlo methods in general. The mixture particle filter, however, is able to successfully track both modes throughout.

To determine the generality of this result the experiment was repeated 20 times for different numbers of particles. For each run the performance score was defined as the fraction of time steps during which both modes were represented, thus ranging from zero (only one mode tracked throughout) to one (both modes tracked throughout). The results are given in Figure 2. As expected the performance of the standard particle filter increases with an increase in the number of particles, up to 500 particles, from which point it is able to consistently track both modes. Even with a small number of particles the mixture particle filter never fails to track both modes. It is interesting to note the behaviour of the first mixture component weight, also depicted in Figure 2. Since both modes are equally strong the mean weight is approximately 0.5 over all the time steps, regardless of the number of particles. The variability in the weight behaviour, however, decreases with an increase in the number of particles.

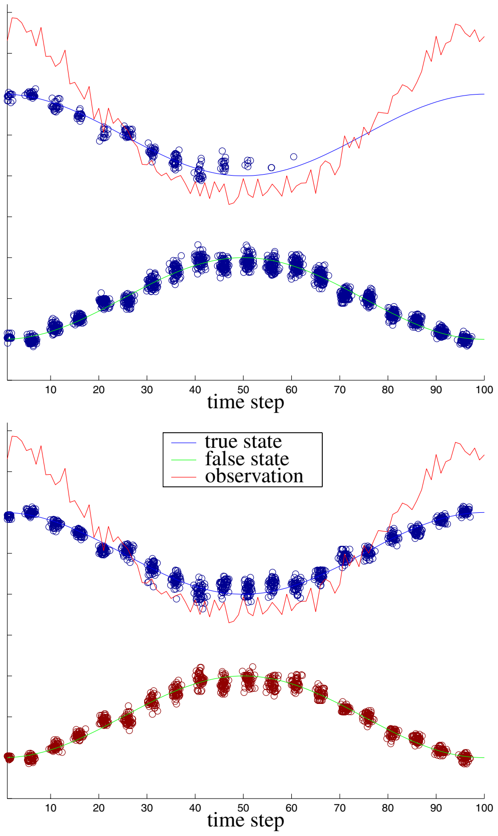

Figure 1. Simulated particles. (Top) Standard particle filter with 100 particles. (Bottom) Mixture particle filter with a maximum of two mixture components and 50 particles per component. The standard particle filter loses track of the true mode after time step 60, whereas the mixture particle filter tracks both modes throughout.

5.2. Visual Tracking

This section considers the problem of tracking football players in a video sequence. In this setting the multimodality is due to the presence of multiple objects (football players), and to some extent, clutter.

More precisely, the object of interest is represented by its bounding box. However, more general object models can easily be accommodated. The reference bounding box to be tracked is specified by the user, and parameterised as[2] $B_{ref} = (x_{ref}, y_{ref}, l_x, l_y)$, where (x_{ref}, y_{ref}) is the centre of the bounding box, and l_x and l_y are the bounding box width and height, respectively. For the tracking the state of the bounding box is taken to be $\mathbf{x} = (x, y, s_x, s_y)$, so

[1] Assuming that the initial state distribution is also symmetric.

[2] In what follows the time subscript is suppressed for the sake of brevity.

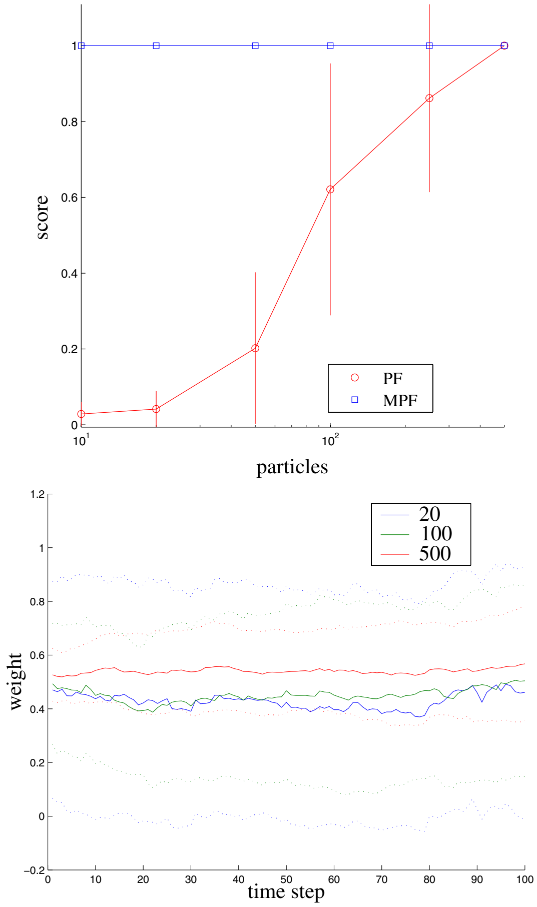

Figure 2. Multiple runs. (Top) Score curves. Performance score and error bars as a function of the number of particles. (Bottom) Mixture component one weight behaviour for an increasing number of particles. The mean weight is indicated by a solid line, and is approximately 0.5, since both modes are equally strong. The dashed one standard deviation lines indicate a decrease in the weight variability as the number of particles increases.

that the corresponding hypothesised bounding box becomes $B_\mathbf{x} = (x, y, s_x l_x, s_y l_y)$. The variables s_x and s_y thus act as scale factors. The components of the state are assumed to follow independent Gaussian random walk models with variances $(\sigma_x^2, \sigma_y^2, \sigma_{s_x}^2, \sigma_{s_y}^2)$. The measurements are taken to be the normalised histograms of the pixel colour components within the bounding box, *i.e.* $\mathbf{y} = (\mathbf{h}^R_{B_\mathbf{x}}, \mathbf{h}^G_{B_\mathbf{x}}, \mathbf{h}^B_{B_\mathbf{x}})$. Note that the measurements depend on the object state. The likelihood for a hypothesised state is defined as

$$L(\mathbf{y}|\mathbf{x}, B_{ref}) \propto \exp\Big[-\big(B^2(\mathbf{h}^R_{B_\mathbf{x}}, \mathbf{h}^R_{B_{ref}}) + B^2(\mathbf{h}^G_{B_\mathbf{x}}, \mathbf{h}^G_{B_{ref}}) + B^2(\mathbf{h}^B_{B_\mathbf{x}}, \mathbf{h}^B_{B_{ref}})\big)/2\sigma^2\Big],$$

where $B(\mathbf{h}_1, \mathbf{h}_2)$ is the Bhattacharyya distance between the normalised N_b bin histograms $\mathbf{h}_1$ and $\mathbf{h}_2$, defined as

$$B(\mathbf{h}_1, \mathbf{h}_2) = \Big[1 - \sum_{b=1}^{N_b} \sqrt{h_{b,1} h_{b,2}}\Big]^{1/2} \in [0, 1].$$

Thus the closer the colour histograms in the hypothesised bounding box are to the corresponding colour histograms in the reference bounding box, the higher the likelihood for the hypothesis. The width of the likelihood is controlled by the variance parameter σ^2. This likelihood is highly non-linear due to the mapping from the state to the measurements. A similar model was employed in the context of object tracking before in [10].

This model with $N_b = 30$, $\sigma = 0.15$, and $(\sigma_x^2, \sigma_y^2, \sigma_{s_x}^2, \sigma_{s_y}^2) = (2.5, 1.5, 0.05, 0.05)$, was used to track the football players in red in the video sequence for which a number of keyframes appear in Figure 3. For both the standard and the mixture particle filters the proposal was taken to be the dynamics, and the initial particles were generated around the red football players in the first frame. For the standard particle filter 200 particles were used, whereas the mixture particle filter was constrained to have a maximum of 10 components, with 20 particles for each component alive. Thus the total number of particles for the mixture particle filter is always equal or less than that for the standard particle filter.

A typical tracking result for both algorithms is depicted in Figure 3. Both algorithms are initialised in exactly the same manner around the four players to be tracked. Even with a large number of particles the standard particle filter is unable to maintain the multi-modality for more than a few frames. The mixture particle filter, however, quickly discovers the four main modes, and successfully track them throughout the video sequence.

6. Conclusions

This paper proposed to model the filtering distribution as a mixture model to better cope with the multi-modality that may arise due to ambiguity or the presence of multiple objects. The general tracking recursion for the mixture model was shown to comprise a prediction step and an update step, similar to the standard Bayesian recursion for a single component model. It was also shown how a Monte Carlo implementation of the general recursion leads to a mixture of particle filters that interact only in the computation of the

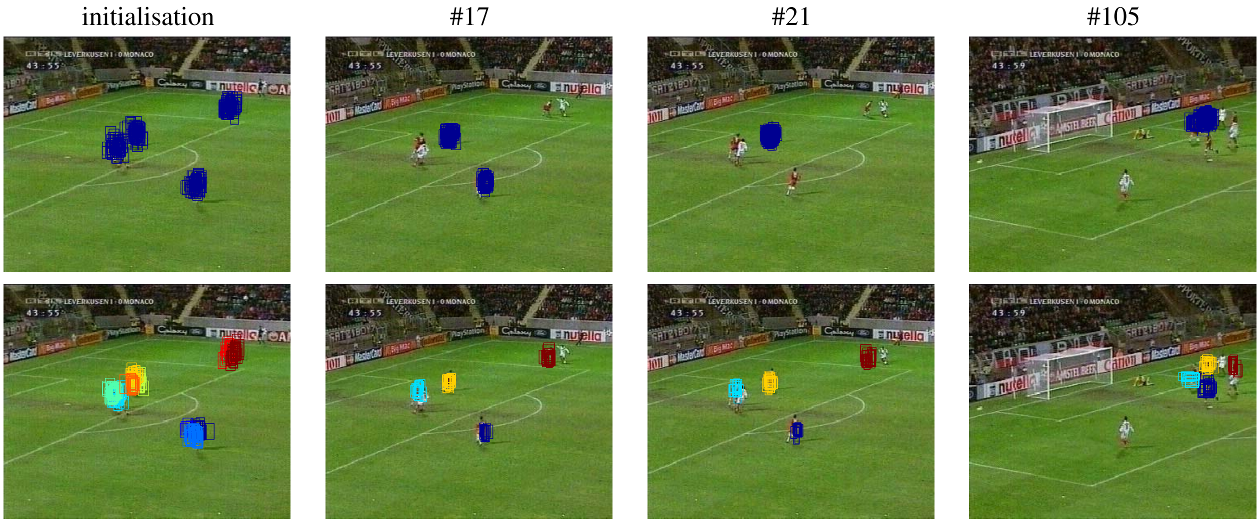

Figure 3. Visual tracking results. (Top) Standard particle filter with 200 particles. (Bottom) Mixture particle filter with a maximum of 10 mixture components and 20 particles per component. All the particles for the standard particle filter quickly migrate to one of the modes. The mixture particle filter rapidly discovers the four main modes and successfully track them throughout the video sequence.

mixture weights. This mixture particle filter is able to maintain the multi-modality inherent in tracking problems where the standard particle filter fails, as was illustrated on a synthetic and a real world tracking problem.

References

[1] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, 1979.

[2] Y. Bar-Shalom, L. Xiao-Rong, and T. Kirubarajan. *Estimation, Tracking and Navigation: Theory, Algorithms and Software*. John Wiley and Sons, 2001.

[3] S. L. Dockstader and A. M. Tekalp. Tracking multiple objects in the presence of articulated and occluded motion. In *Proceedings of the IEEE Workshop on Human Motion*, pages 88–95, 2000.

[4] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[5] A. Doucet, S. J. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[6] S. Gokturk, J. Bouguet, and R. Grzeszczuk. A data-driven model for monocular face tracking. In *Proc. Int. Conf. Computer Vision*, pages 701–708, 2001.

[7] C. Hue, J.-P. Le Cadre, and P. Pérez. Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):791–812, 2002.

[8] M. Isard and J. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *Proc. Int. Conf. Computer Vision*, pages II: 34–41, 2001.

[9] A. Milstein, J. N. Sánchez, and E. T. Williamson. Robust global localization using clustered particle filtering. In *Proceedings of AAAI/IAAI*, pages 581–586, 2002.

[10] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Proc. Europ. Conf. Computer Vision*, pages I: 661–675, 2002.

[11] C. Rasmussen. Joint likelihood methods for mitigating visual tracking disturbances. In *Proceedings of the IEEE Workshop on Multi-Object Tracking*, pages 69–76, 2001.

[12] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):00–141, 2000.