A Boosted Particle Filter: Multitarget Detection and Tracking

Kenji Okuma, Ali Taleghani, Nando de Freitas,
James J. Little, and David G. Lowe

University of British Columbia, Vancouver B.C V6T 1Z4, CANADA
okumak@cs.ubc.ca
http://www.cs.ubc.ca/~okumak

Abstract. The problem of tracking a varying number of non-rigid objects has two major difficulties. First, the observation models and target distributions can be highly non-linear and non-Gaussian. Second, the presence of a large, varying number of objects creates complex interactions with overlap and ambiguities. To surmount these difficulties, we introduce a vision system that is capable of learning, detecting and tracking the objects of interest. The system is demonstrated in the context of tracking hockey players using video sequences. Our approach combines the strengths of two successful algorithms: mixture particle filters and Adaboost. The mixture particle filter [17] is ideally suited to multi-target tracking as it assigns a mixture component to each player. The crucial design issues in mixture particle filters are the choice of the proposal distribution and the treatment of objects leaving and entering the scene. Here, we construct the proposal distribution using a mixture model that incorporates information from the dynamic models of each player and the detection hypotheses generated by Adaboost. The learned Adaboost proposal distribution allows us to quickly detect players entering the scene, while the filtering process enables us to keep track of the individual players. The result of interleaving Adaboost with mixture particle filters is a simple, yet powerful and fully automatic multiple object tracking system.

1 Introduction

Automated tracking of multiple objects is still an open problem in many settings, including car surveillance [10], sports [12,13] and smart rooms [6] among many others [5,7,11]. In general, the problem of tracking visual features in complex environments is fraught with uncertainty [6]. It is therefore essential to adopt principled probabilistic models with the capability of learning and detecting the objects of interest. In this work, we introduce such models to attack the problem of tracking a varying number of hockey players on a sequence of digitized video from TV.

Over the last few years, particle filters, also known as condensation or sequential Monte Carlo, have proved to be powerful tools for image tracking [3, 8,14,15]. The strength of these methods lies in their simplicity, flexibility, and systematic treatment of nonlinearity and non-Gaussianity.

Various researchers have attempted to extend particle filters to multi-target tracking. Among others, Hue *et. al* [5] developed a system for multitarget tracking by expanding the state dimension to include component information, assigned by a Gibbs sampler. They assumed a fixed number of objects. To manage a varying number of objects efficiently, it is important to have an automatic detection process. The Bayesian Multiple-BLob tracker (BraMBLe) [7] is an important step in this direction. BraMBLe has an automatic object detection system that relies on modeling a fixed background. It uses this model to identify foreground objects (targets). In this paper, we will relax this assumption of a fixed background in order to deal with realistic TV video sequences, where the background changes.

As pointed out in [17], particle filters may perform poorly when the posterior is multi-modal as the result of ambiguities or multiple targets. To circumvent this problem, Vermaak *et* al introduce a mixture particle filter (MPF), where each component (mode or, in our case, hockey player) is modelled with an individual particle filter that forms part of the mixture. The filters in the mixture interact only through the computation of the importance weights. By distributing the resampling step to individual filters, one avoids the well known problem of sample depletion, which is largely responsible for loss of track [17].

In this paper, we extend the approach of Vermaak *et* al. In particular, we use a cascaded Adaboost algorithm [18] to learn models of the hockey players. These detection models are used to guide the particle filter. The proposal distribution consists of a probabilistic mixture model that incorporates information from Adaboost and the dynamic models of the individual players. This enables us to quickly detect and track players in a dynamically changing background, despite the fact that the players enter and leave the scene frequently. We call the resulting algorithm the Boosted Particle Filter (BPF).

2 Statistical Model

In non-Gaussian state-space models, the state sequence $\{\mathbf{x}_t; t \in \mathbb{N}\}, \mathbf{x}_t \in \mathbb{R}^{n_{\mathbf{x}}}$, is assumed to be an unobserved (hidden) Markov process with initial distribution $p(\mathbf{x}_0)$ and transition distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, where $n_{\mathbf{x}}$ is the dimension of the state vector. In our tracking system, this transition model corresponds to a standard autoregressive dynamic model. The observations $\{\mathbf{y}_t; t \in \mathbb{N}^*\}, \mathbf{y}_t \in \mathbb{R}^{n_{\mathbf{y}}}$, are conditionally independent given the process $\{\mathbf{x}_t; t \in \mathbb{N}\}$ with marginal distribution $p(\mathbf{y}_t|\mathbf{x}_t)$, where $n_{\mathbf{y}}$ is the dimension of the observation vector.

2.1 Observation Model

Following [14], we adopt a multi-color observation model based on Hue-Saturation-Value (HSV) color histograms. Since HSV decouples the intensity (i.e., value) from color (i.e., hue and saturation), it is reasonably insensitive to illumination effects. An HSV histogram is composed of $N = N_h N_s + N_v$ bins and we denote $b_t(\mathbf{d}) \in \{1, \ldots N\}$ as the bin index associated with the
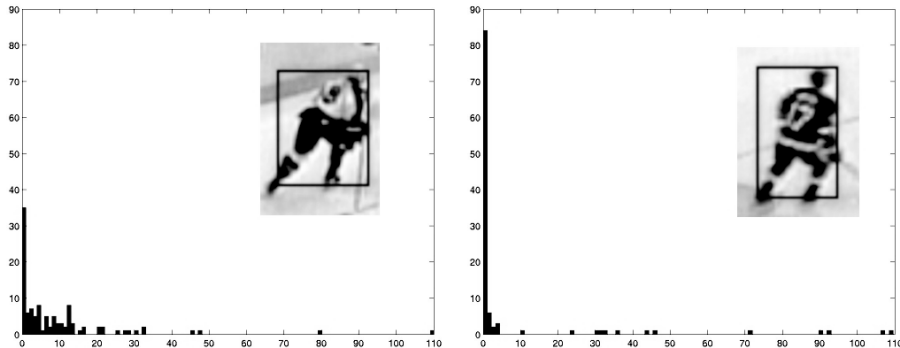
Color histogram of a player (LEFT: white uniform RIGHT: red uniform)

Fig. 1. Color histograms: This figure shows two color histograms of selected rectangular regions, each of which is from a different region of the image. The player on left has uniform whose color is the combination of dark blue and white and the player on right has a red uniform. One can clearly see concentrations of color bins due to limited number of colors. In (a) and (b), we set the number of bins, $N = 110$, where N_h, N_s, and N_v are set to 10

color vector $\mathbf{y}_t(\mathbf{k})$ at a pixel location $\mathbf{d}$ at time t. Figure 1 shows two instances of the color histogram. If we define the candidate region in which we formulate the HSV histogram as $R(\mathbf{x}_t) \triangleq \mathbf{l}_t + s_t W$, then a kernel density estimate $\mathbf{K}(\mathbf{x}_t) \triangleq \{k(n; \mathbf{x}_t)\}_{n=1,\dots,N}$ of the color distribution at time t is given by [1,14]:

$$k(n; \mathbf{x}_t) = \eta \sum_{\mathbf{d} \in R(\mathbf{x}_t)} \delta[b_t(\mathbf{d}) - n] \tag{1}$$

where δ is the delta function, η is a normalizing constant which ensures k to be a probability distribution, $\sum_{n=1}^{N} k(n; \mathbf{x}_t) = 1$, and a location $\mathbf{d}$ could be any pixel location within $R(\mathbf{x}_t)$. Eq. (1) defines $k(n; \mathbf{x}_t)$ as the probability of a color bin n at time t.

If we denote $\mathbf{K}^* = \{k^*(n; \mathbf{x}_0)\}_{n=1,\dots,N}$ as the reference color model and $\mathbf{K}(\mathbf{x}_t)$ as a candidate color model, then we need to measure the data likelihood (i.e., similarity) between $\mathbf{K}^*$ and $\mathbf{K}(\mathbf{x}_t)$. As in [1,14], we apply the Bhattacharyya similarity coefficient to define a distance ξ on HSV histograms. The mathematical formulation of this measure is given by [1]:

$$\xi[\mathbf{K}^*, \mathbf{K}(\mathbf{x}_t)] = \left[1 - \sum_{n=1}^{N} \sqrt{k^*(n; \mathbf{x}_0) k(n; \mathbf{x}_t)} \right]^{\frac{1}{2}} \tag{2}$$

Statistical properties of near optimality and scale invariance presented in [1] ensure that the Bhattacharyya coefficient is an appropriate choice of measuring similarity of color histograms. Once we obtain a distance ξ on the HSV color histograms, we use the following likelihood distribution given by [14]:

$$p(\mathbf{y}_t | \mathbf{x}_t) \propto e^{-\lambda \xi^2 [\mathbf{K}^*, \mathbf{K}(\mathbf{x}_t)]} \tag{3}$$

where $\lambda = 20$. λ is suggested in [14,17], and confirmed also on our experiments. Also, we set the size of bins N_h, N_s, and N_v as 10.

The HSV color histogram is a reliable approximation of the color density on the tracked region. However, a better approximation is obtained when we consider the spatial layout of the color distribution. If we define the tracked region as the sum of r sub-regions $R(\mathbf{x}_t) = \sum_{j=1}^{r} R_j(\mathbf{x}_t)$, then we apply the likelihood as the sum of the reference histograms $\{k_j^*\}_{j=1,\ldots,r}$ associated with each sub-region by [14]:

$$p(\mathbf{y}_t|\mathbf{x}_t) \propto e^{\sum_{j=1}^{r} -\lambda \xi^2 \left[\mathbf{K}_j^*, \mathbf{K}_j(\mathbf{x}_t)\right]} \tag{4}$$

Eq. (4) shows how the spatial layout of the color is incorporated into the data likelihood. In Figure 2, we divide up the tracked regions into two sub-regions in order to use spatial information of the color in the appearance of a hockey player.
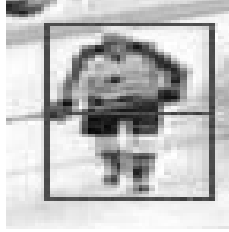
Fig. 2. Multi-part color likelihood model: This figure shows our multi-part color likelihood model. We divide our model into two sub-regions and take a color histogram from each sub-region so that we take into account the spatial layout of colors of two sub-regions

For hockey players, their uniforms usually have a different color on their jacket and their pants and the spatial relationship of different colors becomes important.

2.2   The Filtering Distribution

We denote the state vectors and observation vectors up to time t by $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0 \ldots \mathbf{x}_t\}$ and $\mathbf{y}_{0:t}$. Given the observation and transition models, the solution to the filtering problem is given by the following Bayesian recursion [3]:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{y}_{0:t}) &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{0:t-1})} \\ &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)\int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})d\mathbf{x}_{t-1}}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{0:t-1})\,d\mathbf{x}_t} \end{aligned} \tag{5}$$

To deal with multiple targets, we adopt the mixture approach of [17]. The posterior distribution, $p(\mathbf{x}_t|\mathbf{y}_{0:t})$, is modelled as an M-component non-parametric mixture model:

$$p(\mathbf{x}_t|\mathbf{y}_{0:t}) = \sum_{j=1}^{M} \Pi_{j,t} p_j(\mathbf{x}_t|\mathbf{y}_{0:t}) \tag{6}$$

where the mixture weights satisfy $\sum_{m=1}^{M} \Pi_{m,t} = 1$. Using the the filtering distribution, $p_j(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1})$, computed in the previous step, the predictive distribution becomes

$$p(\mathbf{x}_t|\mathbf{y}_{0:t-1}) = \sum_{j=1}^{M} \Pi_{j,t-1} p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \tag{7}$$

where $p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p_j(\mathbf{x}_{t-1}|\mathbf{y}_{0:t-1}) d\mathbf{x}_{t-1}$. Hence, the updated posterior mixture takes the form

$$\begin{aligned}
p(\mathbf{x}_t|\mathbf{y}_{0:t}) &= \frac{\sum_{j=1}^{M} \Pi_{j,t-1} p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{\sum_{k=1}^{M} \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \, d\mathbf{x}_t} \\
&= \sum_{j=1}^{M} \left[\frac{\Pi_{j,t-1} \int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \, d\mathbf{x}_t}{\sum_{k=1}^{M} \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \, d\mathbf{x}_t} \right] \\
&\quad \times \left[\frac{p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1})}{\int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \, d\mathbf{x}_t} \right] \\
&= \sum_{j=1}^{M} \Pi_{j,t} p_j(\mathbf{x}_t|\mathbf{y}_{0:t})
\end{aligned} \tag{8}$$

where the new weights (independent of $\mathbf{x}_t$) are given by:

$$\Pi_{j,t} = \left[\frac{\Pi_{j,t-1} \int p_j(\mathbf{y}_t|\mathbf{x}_t) p_j(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \, d\mathbf{x}_t}{\sum_{k=1}^{M} \Pi_{k,t-1} \int p_k(\mathbf{y}_t|\mathbf{x}_t) p_k(\mathbf{x}_t|\mathbf{y}_{0:t-1}) \, d\mathbf{x}_t} \right]$$

Unlike a mixture particle filter by [17], we have M different likelihood distributions, $\{p_j(\mathbf{y}_t|\mathbf{x}_t)\}_{j=1...M}$. When one or more new objects appear in the scene, they are detected by Adaboost and automatically initialized with an observation model. Using a different color-based observation model allows us to track different colored objects.

3   Particle Filtering

In standard particle filtering, we approximate the posterior $p(\mathbf{x}_t|\mathbf{y}_{0:t})$ with a Dirac measure using a finite set of N particles $\{\mathbf{x}_t^i\}_{i=1...N}$. To accomplish this, we sample candidate particles from an appropriate proposal distribution $\tilde{\mathbf{x}}_t^i \sim q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t})$ (In the simplest scenario, it is set as $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{0:t}) =$

$p(\mathbf{x}_t|\mathbf{x}_{t-1})$, yielding the bootstrap filter [3]) and weight these particles according to the following importance ratio:

$$w_t^i = w_{t-1}^i \frac{p(\mathbf{y}_t|\tilde{\mathbf{x}}_t^i)p(\tilde{\mathbf{x}}_t^i|\mathbf{x}_{t-1}^i)}{q(\tilde{\mathbf{x}}_t^i|\mathbf{x}_{0:t-1}^i, \mathbf{y}_{0:t})} \qquad (9)$$

We resample the particles using their importance weights to generate an unweighted approximation of $p(\mathbf{x}_t|\mathbf{y}_{0:t})$. In the mixture approach of [17], the particles are used to obtain the following approximation of the posterior distribution:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{j=1}^{M} \Pi_{j,t} \sum_{i \in \mathcal{I}_j} w_t^i \delta_{\mathbf{x}_t^i}(\mathbf{x}_t)$$

where $\mathcal{I}_j$ is the set of indices of the particles belonging to the j-th mixture component. As with many particle filters, the algorithm simply proceeds by sampling from the transition priors (note that this proposal does not use the information in the data) and updating the particles using importance weights derived from equation (8); see Section 3 of [17] for details.

In [17], the mixture representation is obtained and maintained using a simple K-means spatial reclustering algorithm. In the following section, we argue that boosting provides a more satisfactory solution to this problem.

4   Boosted Particle Filter

The boosted particle filter introduces two important extensions of the MPF. First, it uses Adaboost in the construction of the proposal distribution. This improves the robustness of the algorithm substantially. It is widely accepted that proposal distributions that incorporate the recent observations (in our case, through the Adaboost detections) outperform naive transition prior proposals considerably [15,16]. Second, Adaboost provides a mechanism for obtaining and maintaining the mixture representation. This approach is again more powerful than the naive K-means clustering scheme used for this purpose in [17]. In particular, it allows us to detect objects leaving and entering the scene efficiently.

4.1   Adaboost Detection

We adopt the cascaded Adaboost algorithm of Viola and Jones [18], originally developed for detecting faces. In our experiments, a 23 layer cascaded classifier is trained to detect hockey players. In order to train the detector, a total of 6000 figures of hockey players are used. These figures are scaled to have a resolution of 10×24 pixels. In order to generate such data in a limited amount of time, we speed the selection process by using a program to extract small regions of the image that most likely contain hockey players. Simply, the program finds regions that are centered on low intensities (i.e., hockey players) and surrounded by high intensities (i.e., rink surface). However, it is important to note that the

Fig. 3. Training set of images for hockey players: This figure shows a part of the training data. A total of 6000 different figures of hockey players are used for the training

(a)	(b)	(c)
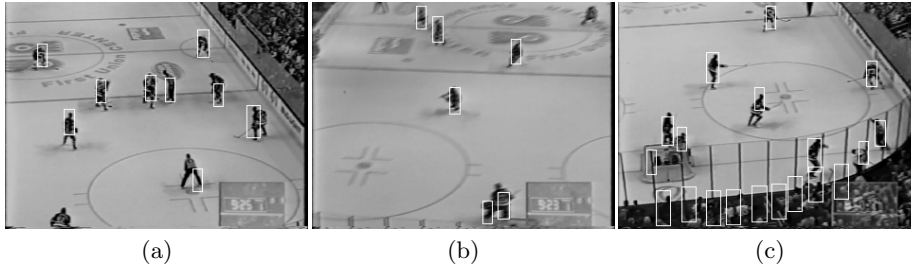
Fig. 4. Hockey player detection result: This figure shows results of the Adaboost hockey detector. (a) and (b) shows mostly accurate detections. In (c), there are a set of false positives detected on audience by the rink

data generated by such a simple script is not ideal for training Adaboost, as shown in Figure 3. As a result, our trained Adaboost produces false positives alongside the edge of the rink shown in (c) of Figure 4. More human intervention with a larger training set would lead to better Adaboost results, although failures would still be expected in regions of clutter and overlap. The non-hockey-player subwindows used to train the detector are generated from 100 images manually chosen to contain nothing but the hockey rink and audience. Since our tracker is implemented for tracking hockey scenes, there is no need to include training images from outside the hockey domain.

The results of using Adaboost in our dataset are shown in Figure 4. Adaboost performs well at detecting the players but often gets confused and leads to many false positives.

4.2   Incorporating Adaboost in the Proposal Distribution

It is clear from the Adaboost results that they could be improved if we considered the motion models of the players. In particular, by considering plausible motions, the number of false positives could be reduced. For this reason, we in-
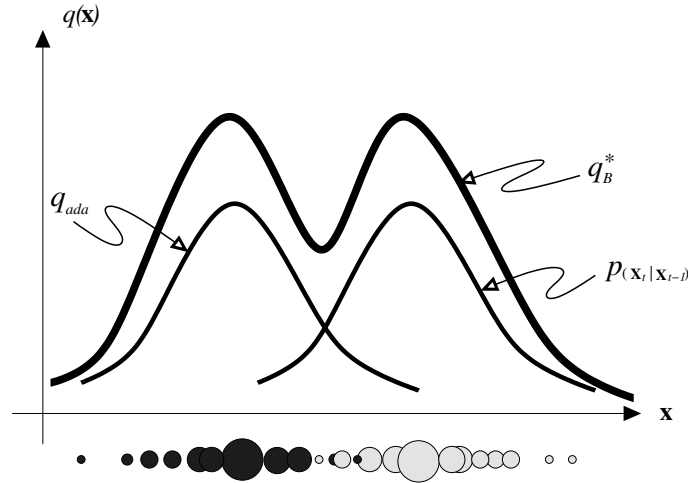
Fig. 5. Mixture of Gaussians for the proposal distribution

corporate Adaboost in the proposal mechanism of our MPF. The expression for the proposal distribution is given by the following mixture.

$$q_B^*(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}) = \alpha q_{ada}(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) + (1 - \alpha)p(\mathbf{x}_t|\mathbf{x}_{t-1}) \qquad (10)$$

where q_{ada} is a Gaussian distribution that we discuss in the subsequent paragraph (See Figure 5). The parameter α can be set dynamically without affecting the convergence of the particle filter (it is only a parameter of the proposal distribution and therefore its influence is corrected in the calculation of the importance weights). When $\alpha = 0$, our algorithm reduces to the MPF of [17]. By increasing α we place more importance on the Adaboost detections. We can adapt the value of α depending on tracking situations, including cross overs, collisions and occlusions.

Note that the Adaboost proposal mechanism depends on the current observation $\mathbf{y}_t$. It is, therefore, robust to peaked likelihoods. Still, there is another critical issue to be discussed: determining how close two different proposal distributions need to be for creating their mixture proposal. We can always apply the mixture proposal when q_{ada} is overlaid on a transition distribution modeled by autoregressive state dynamics. However, if these two different distributions are not overlapped, there is a distance between the mean of these distributions.

If a Monte Carlo estimation of a mixture component by a mixture particle filter overlaps with the nearest cluster given by the Adaboost detection algorithm, we sample from the mixture proposal distribution. If there is no overlap between the Monte Carlo estimation of a mixture particle filter for each mixture component and clusters given by the Adaboost detection, then we set $\alpha = 0$ so that our proposal distribution takes only a transition distribution of a mixture particle filter.
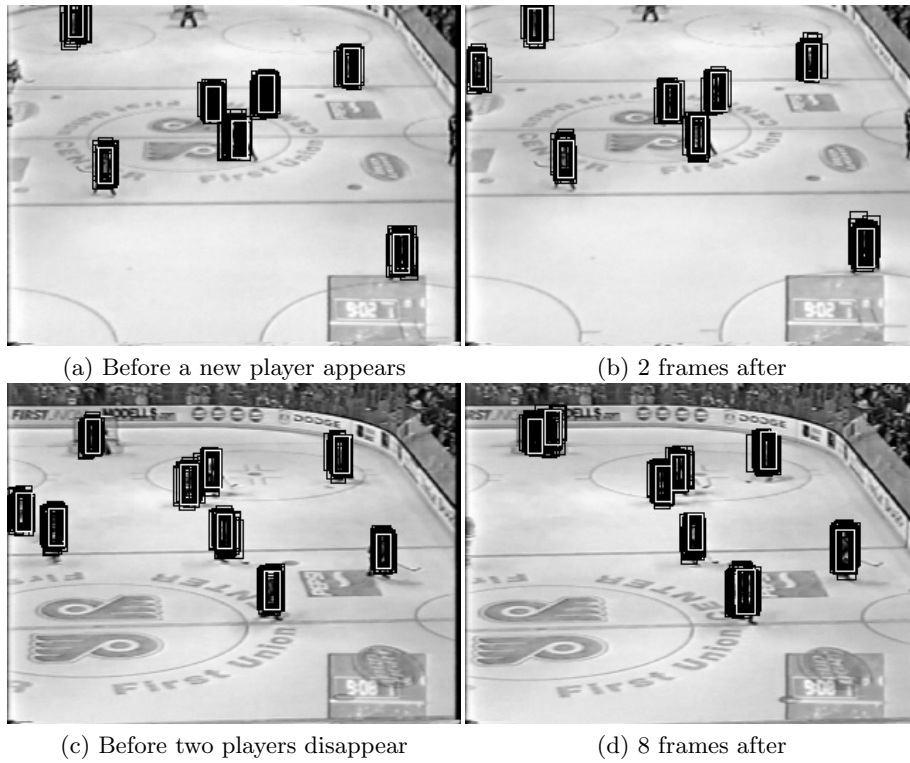
5   Experiments

This section shows BPF tracking results on hockey players in a digitized video sequence from broadcast television. The experiments are performed using our non-optimized implementation in C on a 2.8 GHz Pentium IV.

Figure 6 shows scenes where objects are coming in and out. In this figure, it is important to note that no matter how many objects are in the scene, the mixture representation of BPF is not affected and successfully adapts to the change. For objects coming into the scene, in (a) of the figure, Adaboost quickly detects a new object in the scene within a short time sequence of only two frames. Then BPF immediately assigns particles to an object and starts tracking it.

Figure 7 shows the Adaboost detection result in the left column, a frame number in the middle, and BPF tracking results on the right. In Frame 32 and 33, a small crowd of three players in the middle of the center circle on the rink are not well detected by Adaboost. This is an example of the case

(a) Before a new player appears   (b) 2 frames after

(c) Before two players disappear   (d) 8 frames after

Fig. 6. Objects appearing and disappearing from the scene: This is a demonstration of how well BPF handles objects randomly coming in and out of the scene. (a) and (b) show that a new player that appears in the top left corner of the image is successfully detected and starts to get tracked. (c) and (d) show that two players are disappearing from the scene to left

(a) Adaboost detection result   (b) BPF tracking
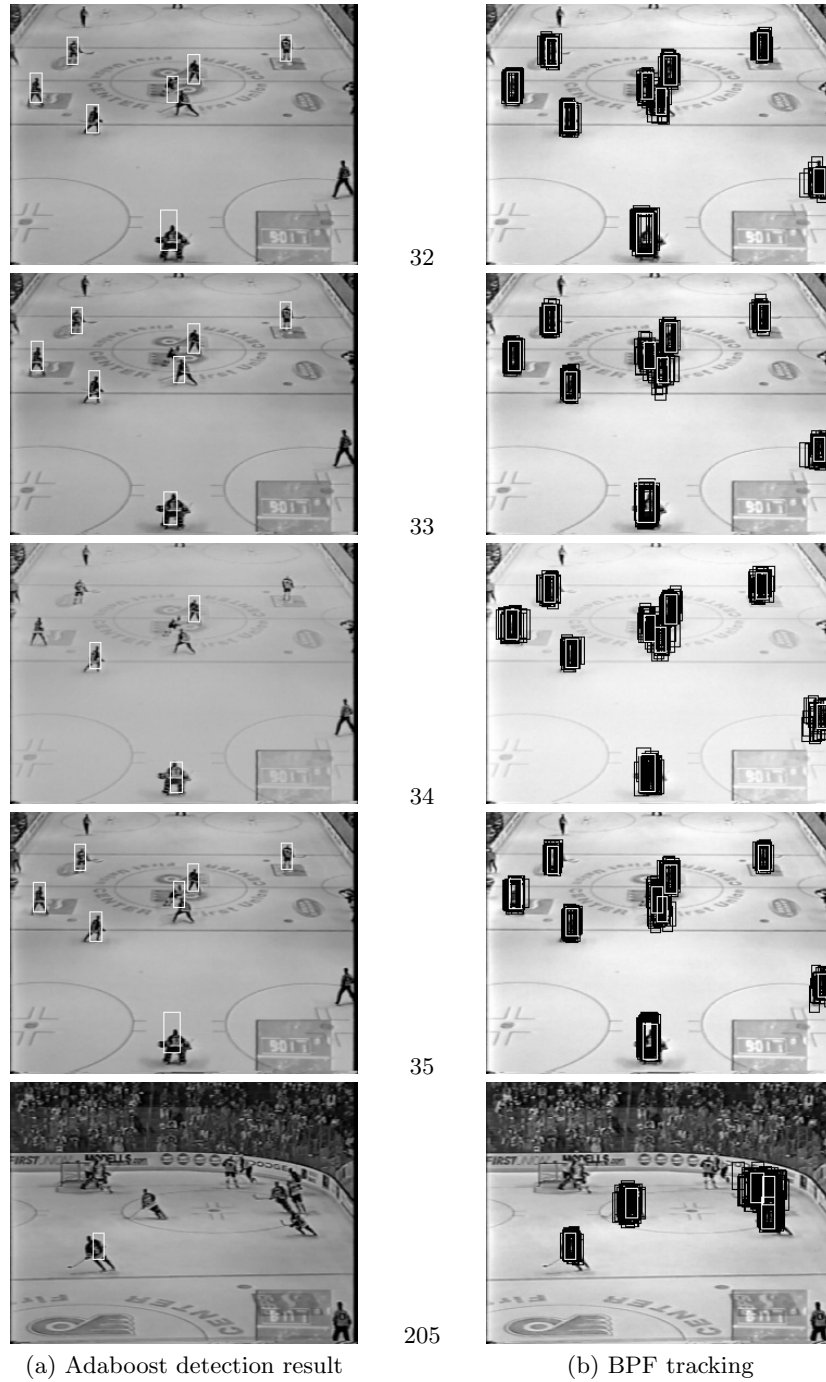
Fig. 7. BPF tracking result: The results of Adaboost detection are shown on the left, with the corresponding boosted particle filter results on the right

in which Adaboost does not work well in clutter. There are only two windows detected by Adaboost. However, BPF successfully tracks three modes in the cluttered environment. Frames 34 and 35 show another case when Adaboost fails. Since Adaboost features are based on different configurations of intensity regions, it is highly sensitive to a drastic increase/decrease of the intensity. The average intensity of the image is clearly changed by a camera flash in Frame 34. The number of Adaboost detections is much smaller in a comparison with the other two consecutive frames. However, even in the case of an Adaboost failure, mixture components are well maintained by BPF, which is also clearly shown in Frame 205 in the figure.

6   Conclusions

We have described an approach to combining the strengths of Adaboost for object detection with those of mixture particle filters for multiple-object tracking. The combination is achieved by forming the proposal distribution for the particle filter from a mixture of the Adaboost detections in the current frame and the dynamic model predicted from the previous time step. The combination of the two approaches leads to fewer failures than either one on its own, as well as addressing both detection and consistent track formation in the same framework.

We have experimented with this boosted particle filter in the context of tracking hockey players in video from broadcast television. The results show that most players are successfully detected and tracked, even as players move in and out of the scene. We believe results can be further improved by including more examples in our Adaboost training data for players that are seen against non-uniform backgrounds. Further improvements could be obtained by dynamic adjustment of the weighting parameter selecting between the Adaboost and dynamic model components of the proposal distribution. Adopting a probabilistic model for target exclusion [11] may also improve our BPF tracking.

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Real-Time Tracking of Non-Rigid Objects using Mean Shift. IEEE Conference on Computer Vision and Pattern Recognition, pp. 142-151 (2000)
2. Deutscher, J., Blake, A., Ried, I.: Articulated body motion capture by annealed particle filtering. IEEE Conference on Computer Vision and Pattern Recognition, (2000)
3. Doucet, A., de Freitas, J. F. G., N. J. Gordon, editors: Sequential Monte Carlo Methods in Practice. Springer-Verlag, New York (2001)
4. Freund, Y., Schapire, R. E.: A decision-theoretic generalization of on-line learning and an application to boosting. Computational Learning Theory, pp. 23-37, Springer-Verlag, (1995)
5. Hue, C., Le Cadre, J.-P., Pérez, P.: Tracking Multiple Objects with Particle Filtering. IEEE Transactions on Aerospace and Electronic Systems, 38(3):791-812 (2002)
6. Intille, S. S., Davis, J. W., Bobick, A.F.: Real-Time Closed-World Tracking. IEEE Conference on Computer Vision and Pattern Recognition, pp. 697-703 (1997)
7. Isard, M., MacCormick, J.: BraMBLe: A Bayesian multiple-blob tracker. International Conference on Computer Vision, pp. 34-41(2001)
8. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. International Journal on Computer Vision, 28(1):5-28 (1998)
9. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems Transactions of the ASME–Journal of Basic Engineering, vol.82 Series D pp.35-45 (1960)
10. Koller, D., Weber, J., Malik, J.: Robust Multiple Car Tracking with Occlusion Reasoning. European Conference on Computer Vision, pp. 186-196, LNCS 800, Springer-Verlag (1994)
11. MacCormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects. International Conference on Computer Vision, pp. 572-578 (1999)
12. Misu, T., Naemura, M., Wentao Zheng, Izumi, Y., Fukui, K.: Robust Tracking of Soccer Players Based on Data Fusion IEEE 16th International Conference on Pattern Recognition, pp. 556-561 vol.1 (2002)
13. Needham, C. J., Boyle, R. D.: Tracking multiple sports players through occlusion, congestion and scale. British Machine Vision Conference, vol. 1, pp. 93-102 BMVA (2001)
14. Pérez, P., Hue. C, Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. European Conference on Computer Vision, (2002)
15. Rui, Y., Chen, Y.: Better Proposal Distributions: Object Tracking Using Unscented Particle Filter. IEEE Conference on Computer Vision and Pattern Recognition, pp. 786-793 (2001)
16. van der Merwe, R., Doucet, A., de Freitas, J. F. G., Wan, E: The Unscented Particle Filter. Advances in Neural Information Processing Systems, vol. 8 pp 351-357 (2000)
17. Vermaak, J., Doucet, A., Pérez, P.: Maintaining Multi-Modality through Mixture Tracking. International Conference on Computer Vision (2003)
18. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE Conference on Computer Vision and Pattern Recognition (2001)