Robust Visual Tracking for Multiple Targets

Yizheng Cai, Nando de Freitas, and James J. Little

University of British Columbia, Vancouver, B.C., Canada, V6T 1Z4,
{yizhengc, nando, little}@cs.ubc.ca

Abstract. We address the problem of robust multi-target tracking within the application of hockey player tracking. The particle filter technique is adopted and modified to fit into the multi-target tracking framework. A rectification technique is employed to find the correspondence between the video frame coordinates and the standard hockey rink coordinates so that the system can compensate for camera motion and improve the dynamics of the players. A global nearest neighbor data association algorithm is introduced to assign boosting detections to the existing tracks for the proposal distribution in particle filters. The mean-shift algorithm is embedded into the particle filter framework to stabilize the trajectories of the targets for robust tracking during mutual occlusion. Experimental results show that our system is able to automatically and robustly track a variable number of targets and correctly maintain their identities regardless of background clutter, camera motion and frequent mutual occlusion between targets.

1 Introduction

Tracking multiple targets, although has its root in control theory, has been of broad interest in many computer vision applications for decades as well. A visual-based multi-target tracking system should be able to track a variable number of objects in a dynamic scene and maintain the correct identities of the targets regardless of occlusions and any other visual perturbations. As it is a very complicated and challenging problem, extensive research work has been done. In this work, we address the problem of robust multi-target tracking within the application of hockey player tracking.

Particle filtering was first introduced to visual tracking by Isard and Blake in [1]. Pérez et al. [2, 3] extended the particle filter framework to track multiple targets. Okuma et al. [4] further extended it [3] by incorporating a boosting detector [5] into the particle filter for automatic initialization of a variable number of targets. However, as their system did not have explicit mechanisms to model mutual occlusions between targets, it loses the identities of the targets after occlusions. On the other hand , various approaches have been taken to solve the occlusion problem in tracking. Kang et al. [6] tried to resolve the ambiguity of the locations of the targets by registering video frames from multiple cameras. Zhao et al. [7] also rectified video frames to the predefined ground plane and model the targets in the 3D space with a body shape model. A static camera was used and background subtraction was applied as well in their work. Explicit target shape modelling can help resolving the likelihood modelling and data association problems during occlusions. The approach is often used within static scenes [8–10]. However, in our application, camera motion makes it difficult to separate target motion or perform background subtraction. Players with drastic pose changes are difficult to be captured by any explicit shape models.

In order to build a tracking system that can correctly track multiple targets regardless of camera motion and mutual occlusion, we propose four improvements on the previous systems. Firstly, a rectification technique is employed to compensate for camera motions. Secondly, a second order autoregression model is adopted as the dynamics model. Thirdly, a global nearest neighbor data association technique is used to correctly associate boosting detections with the existing tracks. Finally, the mean-shift algorithm is embedded into the particle filter framework to stabilize the trajectories of the targets for reliable motion prediction. Although similar work [11] has been done on combining mean-shift with particle filtering, our work is the first one that describes in detail the theoretical formulation of embedding mean-shift seamlessly into the particle filter framework for multi-target tracking. Consequently, although our system performs comparably to the system in [4], it significantly improves upon that system when occlusions happen, which is the main focus of this work.

2 Filtering

Particle filtering has been a successful numerical approximation technique for Bayesian sequential estimation with non-linear, non-Gaussian models. In our application, the fast motion of hockey players and the color model we adopt [12, 13] is highly non-linear and non-Gaussian. Therefore, particle filtering is the ideal model to be the basic skeleton of our tracking system.

The basic Bayesian filtering is a recursive process in which each iteration consists of a prediction step and a filtering step.

$$\begin{array}{ll} \textbf{prediction step:} & p(x_t|y_{0:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{0:t-1})dx_{t-1} \\ \textbf{filtering step:} & p(x_t|y_{0:t}) = \frac{p(y_t|x_t)p(x_t|y_{0:t-1})}{\int p(y_t|x_t)p(x_t|y_{0:t-1})dx_t} \end{array} \tag{1}$$

where the process is initialized by the prior distribution $p(x_0|y_0) = p(x_0)$, $p(x_t|x_{t-1})$ is the target dynamics model, and $p(y_t|x_t)$ is the likelihood model. Particle filtering uses a set of weighted samples $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_s}$ to approximate the posterior distribution in the filtering. The sample set is propagated by sampling from a designed proposal distribution $q(x_t|x_{t-1}, y_{0:t})$, which is called importance sampling. The importance weights of the particles are updated in each iteration as follows

$$w_t^{(i)} \propto \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{t-1}^{(i)}, y_{0:t})} w_{t-1}^{(i)}, \sum_{i=1}^{N_s} w_t^{(i)} = 1. \tag{2}$$

Resampling of the particles is necessary from time to time in each iteration to avoid degeneracy of the importance weights.

One of the critical issues in keeping particle filtering effective is the design of the proposal distribution. The proposal distribution should be able to shift the particles to the regions with high likelihood if there is a big gap between the mode of the prior distribution and the mode of likelihood distribution. The boosted particle filter (BPF) [4] used a mixture of Gaussians model that combines both the dynamics prior and the Adaboost detections [5]

$$q_B^*(x_t|x_{t-1}, y_{0:t}) = \alpha q_{ada}(x_t|y_t) + (1-\alpha)p(x_t|x_{t-1}), \tag{3}$$

where α is the parameter that is dynamically updated according to the overlap between the Gaussian distribution of boosting detection and the dynamics prior. The issue of data association arises here. Details about how to correctly assign boosting detections to the existing tracks will be discussed later. In addition, the original BPF work by Okuma et al. [4] is based on the mixture of particle filter structure (MPF) [3], which has a fixed number of particles for all the targets. As a result, new targets have to steal particles from existing tracks and reduce the accuracy of the approximation. The merge and split of particle clouds in the MPF structure also cause the loss of the correct identities of the targets during occlusions. Therefore, we adopt the boosted particle filter as the basic filtering framework in our application. However, instead of using the MPF structure, we use an independent particle set for each target to avoid the two inherent disadvantages of MPF.

3 Target Dynamics Modelling

In visual tracking systems, accurate modelling of the target dynamics can improve the prediction of the locations of the targets while visual support is insufficient due to occlusion. However, because of the camera motion in our application, the image coordinate system changes over time with respect to the hockey rink coordinates. Therefore, target motion modelling and prediction in the image coordinates are difficult. We adopt the approach by Okuma et al. [14] to map the locations of the targets from the image coordinates to the standard hockey rink coordinate system which is consistent over time. Therefore, according to the physical law of inertia, the motions of the players in hockey games are better predicted with a constant velocity autoregressive model.

3.1 Rectification

Homography is defined by Hartley and Zisserman in [15] as an invertible mapping h between two planes. Images recorded by cameras are 2D projections of the real world. For any plane in the world, its images from a camera, which can pan, tilt, zoom or even move, are exactly modelled by a homography as long as there is no noticeable lens distortion. As the hockey players are always moving in the plane formed by the hockey rink, their locations on the rink are in the same plane both in the real world and the image space. Therefore, it is possible to project their locations between the two planes.

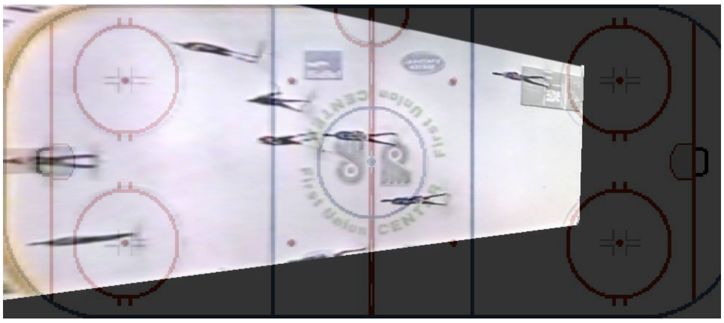


Fig. 1. This shows a projected video frame blended with the standard hockey rink.

The work by Okuma et al. [14] is able to automatically compute the homography between video frames and the hockey rink. Figure 1 shows how the video frames are mapped to the standard rink with the homography. With this homography, the hidden states of the targets are represented in the rink coordinates and particle filtering is performed in the rink coordinates as well. Hidden states will be mapped to the image coordinates when evaluating the likelihood of the observation.

3.2 Autoregressive Dynamics Model

An autoregressive process is a time series modelling strategy which takes into account the historical data to predict the current state value. In this model, the current state $\mathbf{x}_t$ only depends on the previous states with a deterministic mapping function and a stochastic disturbance.

As the particle filtering process is performed in the standard rink coordinates, the motions of the players on the rink are separated from the camera motion. Thus, the modelling is much easier. In hockey games, because of the effect of inertia, a constant velocity model is suitable to model the motion of the players. It is best described by the following second order autoregressive model

$$\mathbf{x}_t = A\mathbf{x}_{t-1} + B\mathbf{x}_{t-2} + C\mathcal{N}(0, \Sigma) \tag{4}$$

where $\{A, B, C\}$ are the autoregression coefficients, $\mathcal{N}(0, \Sigma)$ is a Gaussian noise with zero mean and standard deviation of 1.

4 Data Association

In a standard Bayesian filtering framework, data association is performed to pair the observations and tracks for the evaluation of the likelihood function $p(y_t^m|x_t^n)$. With proper estimation of segmentation and shape of the targets [10], the observation can be assigned to tracks in a globally optimal way. However, as we do not have an explicit shape model for the targets, the particle filter framework in our application handles this level of data association locally in an implicit way. Because the boosting detections are used to improve the proposal distribution in particle filters as in shown in Equation 3, we perform data association at this level to assign boosting detections to the existing tracks.

4.1 Linear Optimization

The assignment problem can be best represented by an assignment matrix shown in Table 1. Each entry in the table is the cost or gain of pairing the corresponding track and observation. In our application, the values of all the entries in the assignment matrix are defined to be the distance between the observations and the tracks in the rink coordinates. Assignments that are forbidden by gating are denoted by × in the corresponding entries. Observations that are forbidden by the gating to be associated to any track are considered as a new track in our application.

Table 1. Example of the assignment matrix for the assignment problem.

	Observations			
Tracks	O1	O2	O3	O4
T1	a_{11}	a_{12}	×	×
T2	a_{21}	×	×	a_{24}
T3	a_{31}	×	×	a_{34}

Such assignment problems stem from economic theory and auction theory as well. The objective is to minimize the cost or maximize the gain subject to a set of constraints. Given the assignment matrix shown in Table 1, the objective is to find a set $X = \{x_{ij}\}$, which are binary indicators, that maximizes or minimizes the objective function $C = \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}x_{ij}$ subject to some linear constrains. Linear programming was initially used to solve this problem. Later on, it was found that the auction algorithm [16] is the most efficient method so far to reach the optimal solution or sub-optimal one without any practical difference.

The extended auction algorithm [17] is able to solve the rectangular matrix problems with the constraint that one observation can only be assigned to one target while a target can have at least one observations. However, in our application, it is very likely that some tracks do not have any observation due to the mis-detection of the boosting detector. Therefore, even if there are some observations within the gate of that track, it is still possible that none of the observations belongs to the track. Hence, the constraints are formalized as

$$\begin{array}{c} \sum_{i=1}^{n} x_{ij} = 1, \forall j \\ \sum_{j=1}^{m} x_{ij} \geq 0, \forall i \end{array} \tag{5}$$

and the solution is

$$x_{i'j} = \begin{cases} 1 \text{ if } i' = \arg_i \min a_{ij} \\ 0 \text{ otherwise} \end{cases} \forall j \tag{6}$$

5 Mean-Shift Embedded Particle Filter

The motivation of embedding the mean-shift algorithm into the particle filter framework of our tracking system is to stabilize the tracking result. It is important for the dynamics model because stabilizing trajectories improves the accuracy of the computed velocity of targets, which is critical for improving the prediction of the location of the targets. It is also important for the likelihood model because accurate prediction leads sampling to more promising areas so that the influence from background clutter and mutual occlusion will be reduced.

5.1 Color Model

We adopted the color model in [13, 4] in our application because it is successful in tracking non-rigid objects with partial occlusion. The model is originally introduced by Comaniciu et al. [18] for the mean-shift based object tracking. The observation of the target is represented by an N-bin color histogram extracted from the region $R(\mathbf{x}_t)$ centered at the location $\mathbf{x}_t$. It is denoted as $Q(\mathbf{x}_t) = \{q(n; \mathbf{x}_t)\}_{n=1,\ldots,N}$, where

$$q(n; \mathbf{x}_t) = C \textstyle\sum_{k \in R(\mathbf{x}_t)} \delta[b(k) - n] \tag{7}$$

where δ is the Kronecker delta function, C is a normalization constant, k is any pixel within the region $R(\mathbf{x}_t)$. By normalizing the color histogram, $Q(\mathbf{x}_t)$ becomes a discrete probabilistic distribution.

The similarity between the current observation $Q(\mathbf{x}_t)$ and the reference model Q^*, which is constructed at the initialization step, is evaluated based on the Bhattacharyya coefficient

$$d(\mathbf{x}_t, \mathbf{x}_0) = \sqrt{1 - \rho[Q(\mathbf{x}_t), Q^*]}, \rho[Q(\mathbf{x}_t), Q^*] = \sum_{n=1}^{N} \sqrt{q(n; \mathbf{x}_t) q^*(n; \mathbf{x}_0)} \tag{8}$$

In order to encode the spatial information of the observation, a multi-part color model [13, 4] is employed, which splits the targets vertically into two parts. The color histogram of the two parts are constructed separately and concatenated in parallel as a new histogram. The likelihood is then evaluated as

$$p(\mathbf{y}_t | \mathbf{x}_t) \propto e^{-\lambda d^2(\mathbf{x}_t, \mathbf{x}_0)} \tag{9}$$

5.2 Mean-Shift

Mean-shift is a nonparametric statistical method that seeks the mode of a density distribution in an iterative procedure. It was first generalized and analyzed by Cheng in [19] and later developed by Comaniciu et al. in [20]. The objective of the mean-shift algorithm is to iteratively shift the current location $\mathbf{x}$ to a new location $\mathbf{x}'$ according to the following relation

$$\mathbf{x}' = \frac{\sum_{i=1}^{M} a_i w(a_i) k \left(\| \frac{a_i - \mathbf{x}}{h} \|^2 \right)}{\sum_{i=1}^{M} w(a_i) k \left(\| \frac{a_i - \mathbf{x}}{h} \|^2 \right)} \tag{10}$$

where $\{a_i\}_{i=1}^{M}$ are normalized points within the region $R(\mathbf{x})$ around the current location $\mathbf{x}$, $w(a_i)$ is the weight associated to each pixel a_i, and $k(x)$ is a kernel profile of kernel K that can be written in terms of a profile function $k : [0, \infty) \rightarrow R$ such that $K(\mathbf{x}) = k(\|\mathbf{x}\|^2)$. According to [19], the kernel profile $k(x)$ should be nonnegative, nonincreasing, piecewise continuous, and $\int_0^\infty k(r)dr < \infty$.

The theory guarantees that the mean-shift offset at location $\mathbf{x}$ is in the opposite direction of the gradient direction of the convolution surface

$$C(x) = \sum_{i=1}^{M} G(a_i - \mathbf{x})w(a_i) \tag{11}$$

where kernel G is called the shadow of kernel K and profile $k(x)$ is proportional to the derivative of profile $g(x)$

In order to utilize mean-shift to analyze a discrete density distribution, i.e., the color histogram, an isotropic kernel G with a convex and monotonically decreasing kernel profile $g(\mathbf{x})$ is superimposed onto the candidate region $R(\mathbf{x}_t)$ to construct such a convolution surface. Therefore, the new color model can be rewritten as

$$q(n; \mathbf{x}_t) = C_h \sum_{i=1}^{M_h} g\left(\left\|\frac{a_i - \mathbf{x}_t}{h}\right\|^2\right) \delta[b(a_i) - n] \tag{12}$$

where C_h is also a normalization constant that depends on h, and h is the bandwidth that determines the scale of the target candidate. It should be noted that in our application, scale of the targets is separated from the state space of the targets and smoothly updated, on per particle basis, using the adaptive scaling strategy in [12]. The weight in the mean-shift update for the color feature is shown below.

$$w(a_i) = \sum_{n=1}^{N} \sqrt{\frac{q^*(n; \mathbf{x}_0)}{q(n; \mathbf{x})}} \delta[b(a_i) - n]. \tag{13}$$

The Epanechnikov profile [12] is chosen to be the kernel profile of kernel G in our application. Because it is linear, the kernel K becomes a constant and the kernel term in Equation 13 can be omitted.

5.3 Mean-Shift Embedded Particle Filter

Applying the mean-shift algorithm directly to the tracking output only gives one deterministic offset at each step. It might not be able to capture the true location of the targets due to background clutter or mutual occlusion between targets in the image. Embedding it into the particle filter framework brings uncertainty to the deterministic method so that the statistical property can improve the robustness of the algorithm. In our application, the mean-shift operation biases all the particles right after the sampling from the mixture of Gaussians proposal distribution and before the resampling step in the particle filter framework. Although similar work [11] has been done for tracking, it was only for single target and the proper way of updating the particle weights after the mean-shift bias was not addressed clearly.

However, embedding the mean-shift algorithm seamlessly into the particle filter framework without introducing bias is non-trivial. Directly biasing sampled particles from the old proposal distribution will change the overall posterior distribution. This makes updating the weight of the particles without bias extremely difficult. Although the mean-shift bias is a deterministic mapping so that it can be seen as a change of variable, it is not applicable in practice. On one hand, because the mean-shift bias is a multiple to one mapping, it is not invertible. On the other hand, because it is difficult to write the mean-shift bias in an analytical expression for differentiation even in a piecewise manner, it is difficult to compute the Jacobian matrix in the variable change.

We take an alternative approach in our application. Mean-shift biases the samples $\{\hat{\mathbf{x}}_t^{(i)}\}_{i=1,...,N}$ that are propagated by the old proposal distribution to a new particle set $\{\tilde{\mathbf{x}}_t^{(i)}\}_{i=1,...,N}$. We denote mean-shift searching with function $\varphi(\cdot)$ such that $\tilde{\mathbf{x}}_t = \varphi(\hat{\mathbf{x}}_t)$. Finally, a Gaussian distribution is superimposed on the biased particles to sample new particles. Therefore, the mean-shift bias with a superimposed Gaussian distribution combined with the old proposal distribution can be considered as a new proposal distribution $\breve{q}(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{y}_t)$. For the new proposal distribution, the weight is updated as follows:

$$\breve{w}_t^{(i)} \propto \frac{p(\mathbf{y}_t|\breve{\mathbf{x}}_t^{(i)})p(\breve{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\breve{q}(\breve{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)},\mathbf{y}_t)} w_{t-1}^{(i)} \tag{14}$$

where $\breve{q}(\breve{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)},\mathbf{y}_t) = \mathcal{N}(\breve{\mathbf{x}}_t^{(i)}|\tilde{\mathbf{x}}_t^{(i)},\mathit{\Sigma})$. Here, $\mathit{\Sigma}$ is a diagonal 2×2 matrix and the value of the two entries are chosen to be the same, which is 0.3, in our application. Note that we use a sample $\breve{\mathbf{x}}_t^{(i)}$ instead of the biased particle $\tilde{\mathbf{x}}_t^{(i)}$. This ensures that the sequential importance sampler remains unbiased and valid.

The following pseudo-code depicts the overall structure of our tracking system, which includes all the contributions in our work.

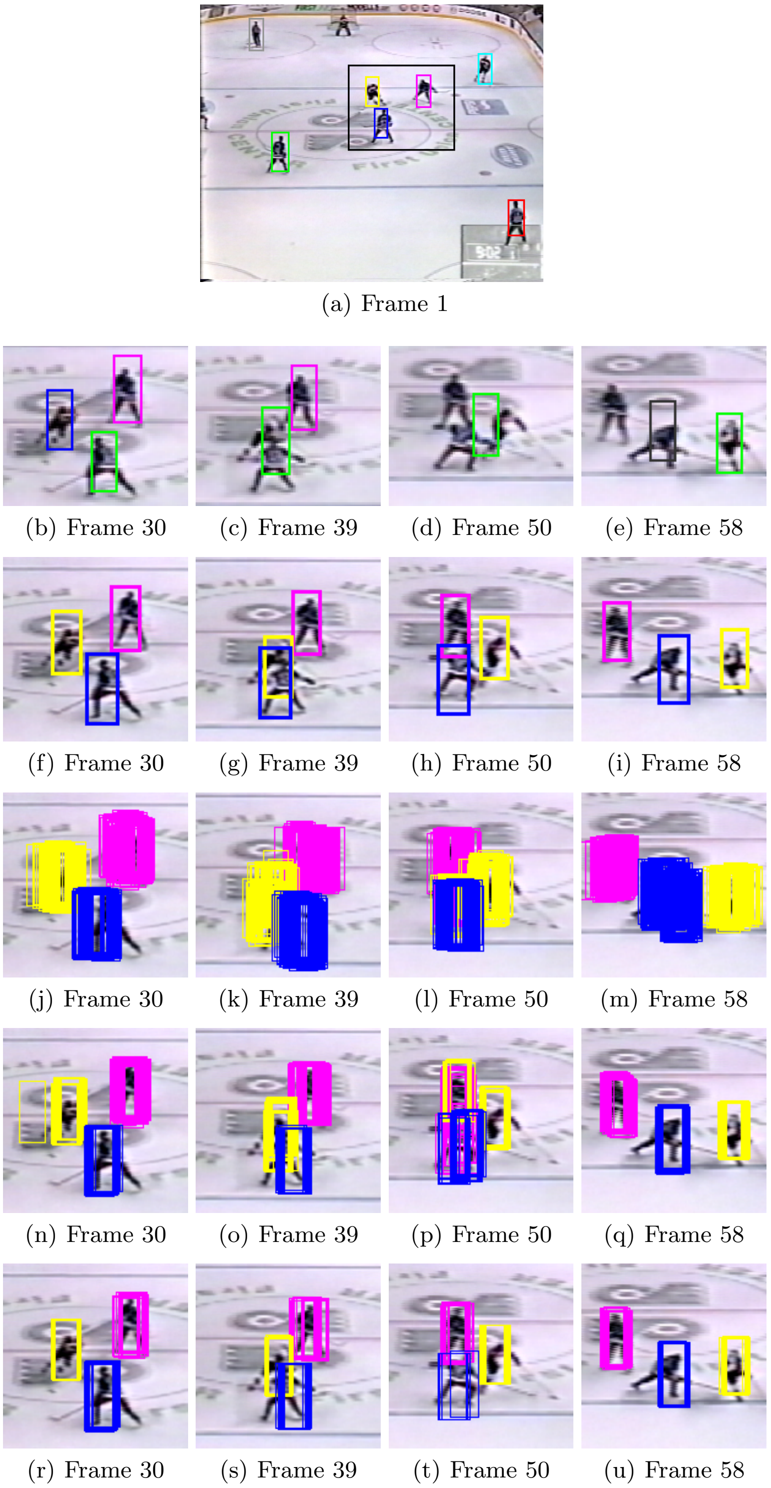
```
– Initialization: $t = 0$
    • Map boosting detections to the rink coordinates to get $\{\mathbf{x}_{k,0}\}_{k=1,...,M_0}$.
    • Create particle set $\{\mathbf{x}_{k,0}^{(i)}, \frac{1}{N}\}_{i=1}^{N}$ by sampling from $p(\mathbf{x}_{k,0})$.
– For $t = 1, ..., T$,
    1. Targets addition and removal
        • Remove targets with large particle set variance.
        • Map boosting detections from the video frame to the rink.
        • Data association
            ∗ Create a particle set for each new target.
            ∗ Associate boosting detections to the existing tracks to construct
              Gaussian mixture proposal distribution $q(\mathbf{x}_{k,t}|\mathbf{x}_{k,t-1},\mathbf{z}_{k,t})$, where
              $\mathbf{z}_{k,t}$ is boosting detection.
    2. For all particles in each track
        • Importance sampling
            * For all particles in each track, sample $\hat{\mathbf{x}}_{k,t}^{(i)} \sim q(\mathbf{x}_{k,t}|\mathbf{x}_{k,t-1}^{(i)}, \mathbf{z}_{k,t})$.
        • Mean-shift biasing
            * Bias the particles as $\tilde{\mathbf{x}}_{k,t}^{(i)} = \varphi(\hat{\mathbf{x}}_{k,t}^{(i)})$.
            * Sample $\breve{\mathbf{x}}_{k,t}^{(i)} \sim \breve{q}(\mathbf{x}_{k,t}|\tilde{\mathbf{x}}_{k,t}^{(i)})$
        • Weight update
            * Update weights $\breve{w}_{k,t}^{(i)}$ according to Equation 14 and normalize.
    3. Deterministic resampling
        • For each track, resample particles to get new sample set $\{\mathbf{x}_{k,t}^{(i)}, \frac{1}{N}\}_{i=1}^{N}$.
    4. Output
        • For each track, $E(\mathbf{x}_{k,t}) = \sum_{i=1}^{N} w_{k,t}^{(i)} \mathbf{x}_{k,t}^{(i)}$
```

6 Experimental Results

Figure 2 shows the comparison between the tracking results of the system in [4] and our system. Subfigure (a) is the key frame in the same tracking sequence that shows the overall view of the tracking results. Subfigures (b-e) and (f-i) are the close-up views of the rectangular region labelled in (a). Each player has a unique color box assigned to it. The color of the same player may not necessarily the same across results of the two systems. According to the results, we can see from Subfigures (b-e) that the trackers merge together when they get close and a new track is created when they split. Meanwhile, our system can maintain correct identities during occlusion.

Subfigures (j-u) in Figure 2 shows the particle representation of the tracking results of our system. In the pseudo-code in Section 5.3, the evolution of particle sets in each iteration of propagation can be divided into three steps: before the mean-shift bias, after the bias, and after the deterministic resampling. The last three rows in the figure compare the difference between the particle sets after each step. Generally, the mean-shift algorithm moves particles from different locations around the target to locations in the neighborhood that are most similar to the reference model in the color space. Therefore, particle sets appear more condensed after the mean-shift bias. The difference between Subfigure (p) and (q) in Figure 2 indicates that mean-shift might move particles to some other targets because of the similarity between the two targets in the color space. However, such particles will be assigned low weights because of the regularization of the dynamics model. As a result, those particles will have much fewer or no children after the resampling stage. For the same reason, particles that are biased to regions without any target, as are shown in Subfigure (n) and (o), will be penalized as well. In summary, both the mean-shift algorithm and the dynamics model penalize erroneous particle hypotheses and improve the robustness of the overall tracking system.

Figure 3 shows more tracking results from three different sequences. All of them are able to correctly maintain the identities of the players regardless of partial of complete occlusions.



(a) Frame 1

(b) Frame 30 (c) Frame 39 (d) Frame 50 (e) Frame 58

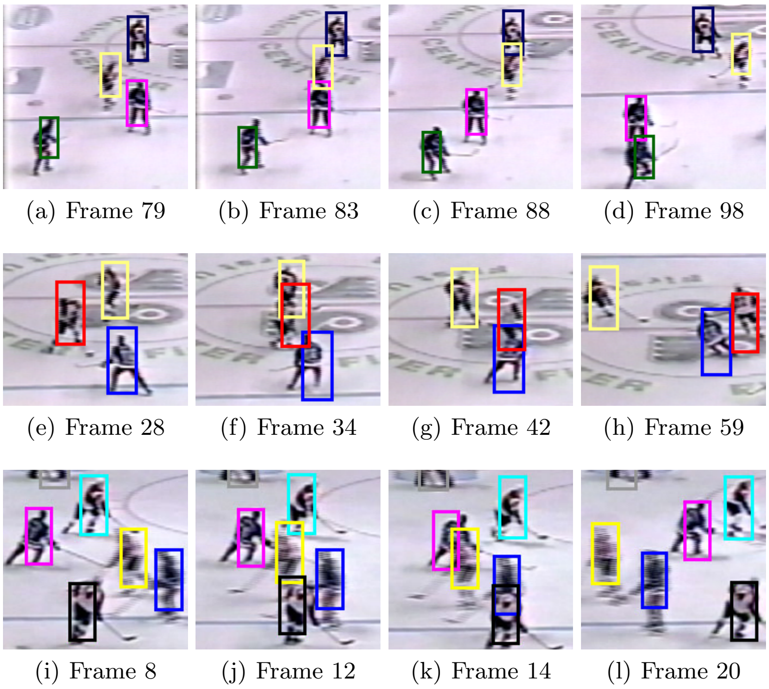
(f) Frame 30 (g) Frame 39 (h) Frame 50 (i) Frame 58

(j) Frame 30 (k) Frame 39 (l) Frame 50 (m) Frame 58

(n) Frame 30 (o) Frame 39 (p) Frame 50 (q) Frame 58

(r) Frame 30 (s) Frame 39 (t) Frame 50 (u) Frame 58

Fig. 2. Each row is a close-up view of the rectangular region in (a). Subfigures (b-e) show the tracking results of the system in [4]. Subfigures (f-i) show the tracking results of our system. Subfigures (j-u) show the particle representation of each target during the tracking process. Different targets are labelled with rectangles of different colors.



(a) Frame 79 (b) Frame 83 (c) Frame 88 (d) Frame 98

(e) Frame 28 (f) Frame 34 (g) Frame 42 (h) Frame 59

(i) Frame 8 (j) Frame 12 (k) Frame 14 (l) Frame 20

Fig. 3. Each row in the figure shows the tracking results of three different sequences where the top one is the same sequence as the one shown in Figure 2.

7 Conclusions

In this paper, we devote our endeavors to building a tracking system that is able to robustly track multiple targets and correctly maintain their identities regardless of background clutter, camera motions and mutual occlusion between targets.

The new particle filter framework is more suitable for tracking a variable number of targets. The rectification technique compensates for the camera motion and make the motion of targets easier to predict by the second order autoregression model. The linear optimization algorithm achieves the global optimal solution to correctly assign boosting detections to the existing tracks. Finally, the mean-shift embedded particle filter is able to stabilize the trajectory of the targets and improve the dynamics model prediction. It biases particles to new locations with high likelihood so that the variance of particle sets decreases significantly.

Acknowledgements

This work has been supported by grants from NSERC, the GEOIDE Network of Centres of Excellence and Honeywell Video Systems.

References

1. Isard, M., Blake, A.: CONDENSATION-Conditional Density Propagation for Visual Tracking. Internatinal Journal on Computer Vision **29**(1) (1998) 5–28
2. Hue, C., Le Cadre, J., Pèrez, P.: Tracking Multiple Objects with Particle Filtering. In: IEEE Transactions on Aerospace and Electronic Systems. Volume 38. (2003) 313–318
3. Vermaak, J., Doucet, A., Pèrez, P.: Maintaining Multi-modality through Mixture Tracking. In: Internatinal Conference on Computer Vision. Volume II. (2003) 1110–1116
4. Okuma, K., Taleghani, A., de Freitas, J., Little, J., Lowe, D.: A Boosted Particle Filter: Multitarget Detection and Tracking. In: European Conference on Computer Vision. Volume I. (2004) 28–39
5. Viola, P., Jones, M.: Robust Real-Time Face Detection. Internatinal Journal on Computer Vision **57**(2) (2004) 137–154
6. Kang, J., Cohen, I., Medioni, G.: Soccer Player Tracking across Uncalibrated Camera Streams. In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS) In Conjunction with ICCV. (2003) 172–179
7. Zhao, T., Nevatia, R.: Tracking Multiple Humans in Complex Situations. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(9) (2004) 1208–1221
8. MacCormick, J., Blake, A.: A Probabilistic Exclusion Principle for Tracking Multiple Objects. Internatinal Journal on Computer Vision **39**(1) (2000) 57–71
9. Isard, M., MacCormick, J.: BraMBLe: A Bayesian Multiple-Blob Tracker. In: Internatinal Conference on Computer Vision. Volume II. (2001) 34–41
10. Rittscher, J., Tu, P., Krahnstoever, N.: Simultaneous estimation of segmentation and shape. In: CVPR05. Volume II. (2005) 486–493
11. Shan, C., Wei, Y., Tan, T., Ojardias, F.: Real Time Hand Tracking by Combining Particle Filtering and Mean Shift. In: International Conference on Automatic Face and Gesture Recognition. (2004) 669–674
12. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based Object Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(5) (2003) 564–577
13. Pèrez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. In: European Conference on Computer Vision. Volume I. (2002) 661–675
14. Okuma, K., Little, J., Lowe, D.: Automatic Rectification of Long Image Sequences. In: Asian Conference on Computer Vision. (2004)
15. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
16. Blackman, S., Popoli, R.: Design and Analysis of Modern Tracking Systems. Artech House, Norwood (1999)
17. Bertsekas, D.: Linear Network Optimization: Algorithms and Codes. The MIT Press, Cambridge (1991)
18. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: International Conference on Computer Vision and Pattern Recognition. (2000) 2142–2149
19. Cheng, Y.: Mean Shift, Mode Seeking, and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence **17**(8) (1995) 790–799
20. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(5) (2002) 603–619