

Multi-camera track-before-detect

Murtaza Taj and Andrea Cavallaro

Queen Mary University of London

School of Electronic Engineering and Computer Science

Mile End Road, London, E1 4NS, UK

Email: {murtaza.taj, andrea.cavallaro}@elec.qmul.ac.uk

Abstract—We present a novel multi-camera multi-target fusion and tracking algorithm for noisy data. Information fusion is an important step towards robust multi-camera tracking and allows us to reduce the effect of projection and parallax errors as well as of the sensor noise. Input data from each camera view are projected on a top-view through multi-level homographic transformations. These projected planes are then collapsed onto the top-view to generate a detection volume. To increase track consistency with the generated noisy data we propose to use a track-before-detect particle filter (TBD-PF) on a 5D state-space. TBD-PF is a Bayesian method which extends the target state with the signal intensity and evaluates each image segment against the motion model. This results in filtering components belonging to noise only and enables tracking without the need of hard thresholding the signal. We demonstrate and evaluate the proposed approach on real multi-camera data from a basketball match.

I. INTRODUCTION

Tracking is an important processing step for many single and multi-camera applications such as sports video analysis, traffic monitoring, behavior identification and event detection. The target tracking task is usually performed in two steps, first the detection of objects of interest (targets) and then their temporal linkage from frame to frame. Target detection can be performed on image changes [1] or learned target models [2]. The major drawback of detection-based tracking algorithms is that they rely on hard thresholding the input data and therefore are only applicable on inputs with high signal-to-noise ratios (SNR).

In many single and multi-sensor applications the SNR of the input or pre-processed signal is relatively low. Examples of such signals are the far-field of infrared (IR) images, bearing frequency distributions (sonar) and range-Doppler maps (radar). Examples of sensors whose signals are fused include cameras [3], [4], microphones [5] and radars [6]. The fusion involves triangulation of noisy information that can result in much larger number of solutions than desired. To address this type of data, simultaneous detection and tracking can be performed via track-before-detect (TBD). In TBD the entire sensor image is considered as a measurement. This measurement is a highly non-linear function of the target state and can be solved either by discretization of the state [7] or by employing non-linear state estimation techniques such as particle filtering [8], which are computationally less expensive.

This work was supported in part by the EU, under the FP7 project APIDIS (ICT-216023).

A. Target tracking: related work

We briefly review here the related state of the art on track-after-detect and track-before-detect. *Track-after-detect* can be based on adaptive multi-feature tracking [9] using color and orientation information under a particle filtering (PF) framework. Similarly, color and edge features are used in [10] to track single targets, such as faces and hands, using a trust-region method. Multi-target tracking algorithms include Mixture Particle filters (MPF) [11], where individual interacting PFs perform distributed resampling to avoid track loss due to sample depletion. Similarly, Boosted Particle filtering (BPF) [12] uses proposal distributions with a mixture model that contains contributions from a detector and the target dynamic model. In [13], the target state is augmented with an existence variable to model the number of targets in the Bayesian estimation. This leads to a hybrid estimation problem solved using a jump Markov model [14], as one component of the state vector is discrete valued, while the rest are continuous valued.

The above-mentioned tracking algorithms cannot be applied for targets with low observability. In such cases, a *track-before-detect* (TBD) algorithm can be used. A recursive Bayesian single target TBD is proposed in [15] using particle filtering. This method assumes a point target and extends the target state with the signal intensity based on the assumption that the return intensity from the target is unknown. Similar to multi-target PF, multi-target TBD-PF approaches are also based on extending the target state with an existence variable and solved with a jump Markov model [16]. An approach based on dynamic programming is used in [17] to track airplanes through TBD. In this context, traditional change-based detection cannot be applied because targets are very small and the presence of clouds makes them dimmer. In [5] multiple microphones were used to track multiple speakers using TBD on steered beam forming results. In this approach a conditional probability density is used that characterizes uncertainty in both target state and target number, given the measurements. The polar Hough transform is used in the fusion between multiple radar signals [6]. As the coordinate measurement errors (range, azimuth) degrade the accumulation of a signal in each cell of the Hough space (i.e., reduce the output SNR and the output signal peak, while increasing the output side lobes peak), TBD is applied for target tracking.

Most TBD algorithms are demonstrated on simulated

data [6], [15], [16], [18]. Two exceptions are [5], [17]: [5] is a multi-target multi-sensor tracking algorithm applied on audio sensors and [17] is applied to IR sequences from a single sensor only. To the best of our knowledge, the work we propose in this paper is the first adaptation of the TBD concepts to multi-camera tracking.

B. Contribution

We present a multi-camera multi-target track-before-detect particle filter that uses mean-shift clustering and we demonstrate it on real data. The information from multiple cameras is first fused to obtain a *detection volume* using a multi-layer homography [4]. To track multiple objects in the detection volume, unlike traditional tracking algorithms that incorporate the measurement at the likelihood level, TBD uses the signal intensity in the state representation. Moreover, as different targets can have different signal intensities on the detection volume, we account for this variation in the weight update strategy. Finally, unlike traditional methods that use K-means or Mixture of Gaussian (MoG) clustering on the detections, the proposed approach does not require manual initialization of the targets nor the prior knowledge of the number of clusters as we use mean-shift on the particles, after the update step.

The rest of the paper is organized as follows. Section II describes the proposed algorithm. Experimental results are presented in Section III. Finally, conclusions are drawn in Section IV.

II. MULTI-TARGET TRACK-BEFORE-DETECT PARTICLE FILTERING

In this section, we first introduce the single target track-before-detect formulation based on particle filtering. Next we discuss the framework for multi-sensor data fusion and the proposed multi-target track-before-detect particle filtering (MT-TBD-PF). Finally, we describe the mean-shift clustering and identity propagation approach within MT-TBD-PF.

A. Single target track-before-detect

Let \mathbf{x}_k be the target state vector at time k , using a discrete time model with a fixed sampling period T , the state can be defined as

$$\mathbf{x}_k = (x_k, \dot{x}_k, y_k, \dot{y}_k, I_k)^T, \quad (1)$$

where (x_k, y_k) are the position components, (\dot{x}_k, \dot{y}_k) are the velocity components and I_k is the target signal strength (intensity) at time k . The state evolution can be modeled as

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \nu_k), \quad (2)$$

where f is the state-transition function and ν_k is the process noise. For a linear stochastic process, the state evolution can be expressed as

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + \nu_k, \quad (3)$$

where F is the state transition matrix defined as

$$F = \begin{bmatrix} B & 0_{2 \times 2} & 0_{2 \times 1} \\ 0_{2 \times 2} & B & 0_{2 \times 1} \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}. \quad (4)$$

The process noise ν_k is generally modeled as Gaussian random variable with covariance Q , defined as

$$Q = \begin{bmatrix} D & 0_{2 \times 2} & 0_{2 \times 1} \\ 0_{2 \times 2} & D & 0_{2 \times 1} \\ 0 & 0 & q_2 T \end{bmatrix}, D = \begin{bmatrix} \frac{q_1}{3} T^3 & \frac{q_1}{3} T^2 \\ \frac{q_1}{2} T^2 & q_1 T \end{bmatrix}, \quad (5)$$

where q_1 is the variance of the acceleration noise.

Let z_k be the measurement, at each time k , encoded in a $W \times H$ resolution image. At each pixel position the measurement intensity $z_k(i, j)$ is either due to the presence of the target or due to measurement noise w , that is

$$z_k(i, j) = \begin{cases} h_k(i, j)(\mathbf{x}_k) + w_k(i, j) & \text{if target is present} \\ w_k(i, j) & \text{if target is not present} \end{cases}, \quad (6)$$

where $h_k(i, j)(\cdot)$ is the contribution of the target intensity in the pixel position (i, j) . In case of a point target, the distribution of the target intensity over the surrounding pixels will be only due to the sensor point spread function and can be approximated as

$$h_k(i, j)(\mathbf{x}_k) \approx \frac{\Delta_x \Delta_y I_k}{2\pi A^2} \exp\left(-\frac{(i\Delta_x - x_k)^2 + (j\Delta_y - y_k)^2}{2A^2}\right), \quad (7)$$

where A models the amount of blurring introduced by the sensor and $\Delta_x \times \Delta_y$ is the resolution of the segment centering at $(i\Delta_x, j\Delta_y)$.

Given the set of measurements $Z_k = \{z_m | m = 1, \dots, k\}$ up to time k , the objective is to recursively quantify some degree of belief in the state \mathbf{x}_k taking different values, i.e., to estimate the posterior *pdf* $p(\mathbf{x}_k | Z_k)$. Using the Bayesian recursion, the posterior *pdf* $p(\mathbf{x}_k | Z_k)$ can be computed in two steps: *prediction* and *update*. In the *prediction* step, the prior density of the state at time k is obtained using Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k | Z_{k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | Z_{k-1}) d\mathbf{x}_{k-1}, \quad (8)$$

where $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ is the transition density defined by the target model (Eq. 2) and $p(\mathbf{x}_{k-1} | Z_{k-1})$ is the posterior at time $k-1$. The *update* step is carried out using the measurement at time k by applying Bayes' rule:

$$p(\mathbf{x}_k | Z_k) = \frac{p(z_k | \mathbf{x}_k, Z_{k-1}) p(\mathbf{x}_k | Z_{k-1})}{\int p(z_k | \mathbf{x}_k, Z_{k-1}) p(\mathbf{x}_k | Z_{k-1}) d\mathbf{x}_k}, \quad (9)$$

where $p(z_k | \mathbf{x}_k)$ is the likelihood function.

The above algorithm is implemented using Sampling Importance Resampling (SIR) particle filter where a posterior density is represented by a set of particles each with associated weight $\{\omega_k^n, \mathbf{x}_k^n\}$.

In the *prediction* step, we draw two sets of particles to estimate the predicted density. L_{k-1} particles are generated from the proposal density $q_k(\mathbf{x}_k | \mathbf{x}_{k-1}, Z_k)$ based on target dynamic model (Eq. 3) and another set of J_k particles from another distribution $p(\mathbf{x}_k | Z_k)$ based on current measurement

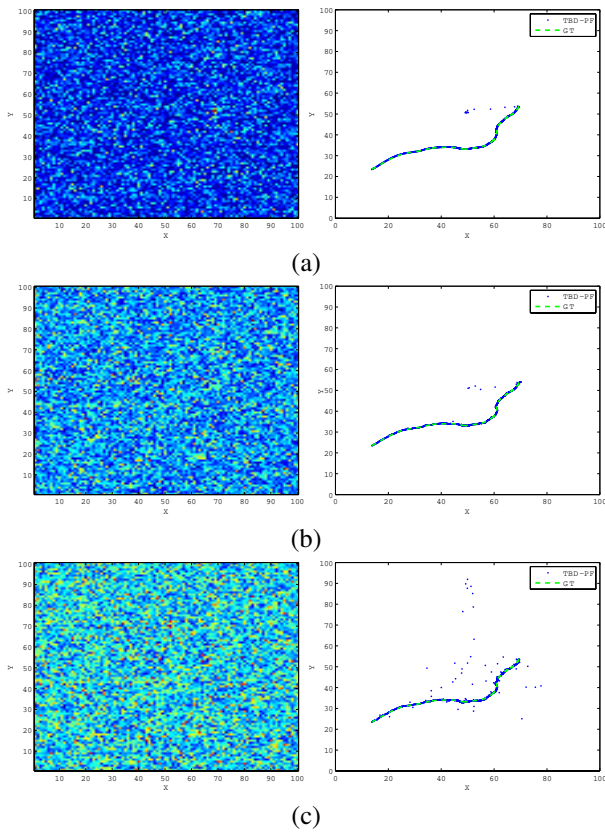


Fig. 1. Sample single target track-before-detect results with varying SNR values: (left) Sample frames from input data illustrating that target is difficult to detect by visual inspection at low SNR, (right) tracking results; (a) SNR = 18.3422db, (c) SNR = 8.6969 db and (e) SNR = 6.2613db. (Blue dots: estimated positions; green dashes: ground truth).

to accommodate new-born targets.

$$\mathbf{x}_{k|k-1}^n = \begin{cases} q_k(\mathbf{x}_k | \mathbf{x}_{k-1}, Z_k) & \text{surviving particles} \\ p(\mathbf{x}_k | Z_k) & \text{new-born particles} \end{cases} \quad n = 1, \dots, L_{k-1} + J_k \quad (10)$$

In the *update* step, the likelihood $p(z_k(i, j) | \mathbf{x}_k^n)$, for the combined set of $L_{k-1} + J_k$ particles, for each pixel (i, j) is computed as

$$p(z_k(i, j) | \mathbf{x}_k^n) = \exp\left(-\frac{h_k(i, j)(h_k(i, j) - 2z_k(i, j))}{2A^2}\right). \quad (11)$$

Since the pixels are assumed to be conditionally independent, the likelihood of the whole image is computed by taking the product over the pixels, thus the updated particle weights are computed as

$$\omega_{k|k-1}^n = \frac{\prod_{i=C_i(\mathbf{x}_{k|k-1}^n)} \prod_{j=C_j(\mathbf{x}_{k|k-1}^n)} p(z_k(i, j) | \mathbf{x}_k^n)}{\sum_{n=1}^{L_k+J_k} \omega_{k|k-1}^n}, \quad (12)$$

where $C_i(\cdot)$ and $C_j(\cdot)$ indicates that only the pixels affected by the target are used in the likelihood computation.

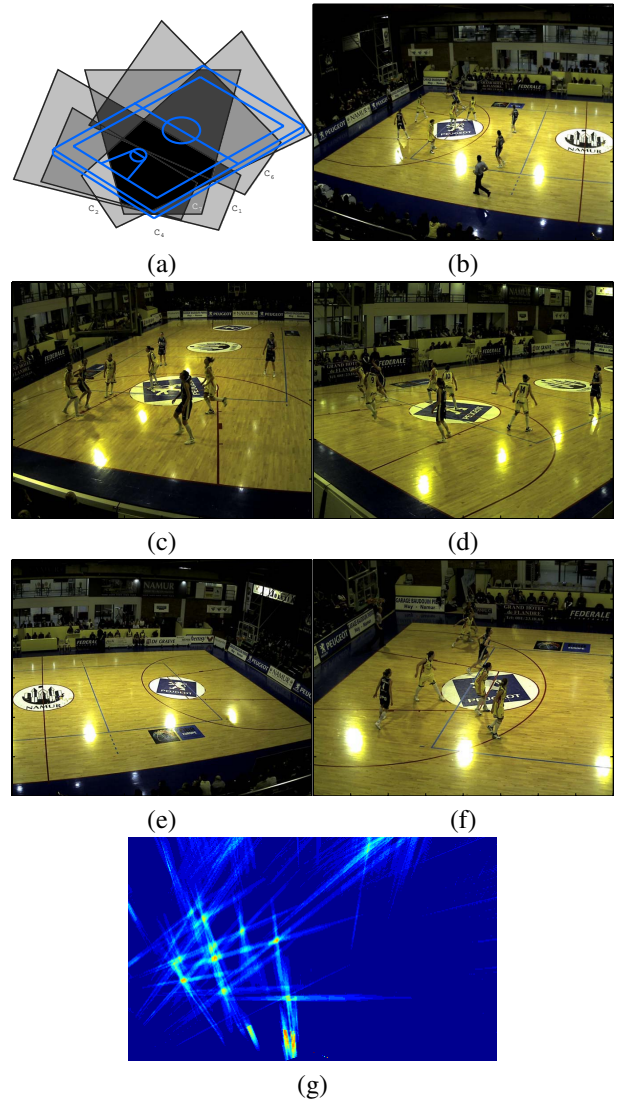


Fig. 2. Camera views and projection result (a) Configuration of cameras in basketball court (excluding two top-mounted cameras with fish eye lenses). (b) Camera 1; (c) camera 2; (d) camera 4; (e) camera 6; (f) camera 7; (g) Sample projection of detection mask from multiple cameras to a top-view.

To avoid the degeneracy problem, the combined set of $L_{k-1} + J_k$ particles are resampled to reduce the number to L_k only by selecting particle for which $\omega_k^n > \lambda_\omega$, where λ_ω is the minimum allowed particle weight. If $\omega_k^n > \lambda_\omega, \forall n$ then $L_k = L_{k-1} + J_k - J_{k+1}$ where $J_{k+1} = N_{min}$ and N_{min} is the minimum number of new-born particles at each time k . Figure 1 shows an example of our single target track-before-detect particle filter using three different SNR values of synthetic data. Although with SNR= 8.6969db and SNR= 6.2613db the target cannot be observed visually due to the noise, it was correctly tracked (Fig. 1(b,c)). When SNR= 6.2613db, the algorithm had some difficulties in identifying the target location, however once enough particles were drawn around the target it was tracked consistently. Note here we use only 1 particle per pixel as compared to other approaches [19] where 25% more particles were used.

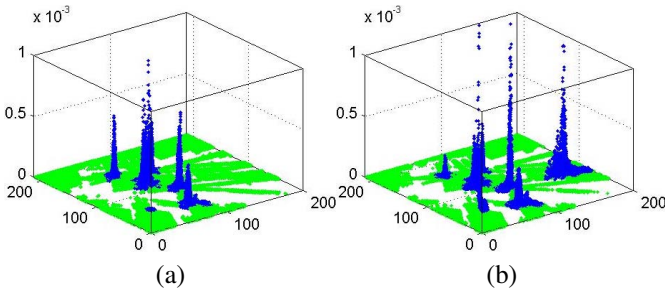


Fig. 3. Example of particle weights and positions. (a) Without the proposed update strategy (one target has very small weights and another one is missing); (b) with the proposed update strategy. As the weights for weak targets are very low without the proposed update, this results in track loses.

B. Multi-sensor, multi-target track-before-detect

Information fusion is applied to enable multi-sensor tracking. We project the detection mask from each camera on the top-view through multi-level homographic transformations similar to [3]. These projected planes are then collapsed to generate a detection volume (Fig. 2). In case of multiple targets, the measurement at each pixel (i, j) can have a contribution from all the targets and Eq. 6 can be modified as

$$z_k(i, j) = \begin{cases} \sum_1^{N_k} h_k(i, j)(\mathbf{x}_k) + w_k(i, j) & \text{if } N_k \neq 0 \\ w_k(i, j) & \text{otherwise} \end{cases}, \quad (13)$$

where N_k is the number of targets at time k . The approximation shown in Eq. (7) is based on point target assumption and is nothing but a truncated 2-D Gaussian density with circular symmetry. Similar approximation can be used in the case of multiple targets in the projected domain by tuning the values for Δ_x , Δ_y and A , respectively. This enables the filtering out of noise that is due to parallax error.

The particle filter may perform poorly when the posterior is multi-modal as the result of multiple-targets [11]. To solve this problem, instead of using the existence variable and the jump Markov model [13], [16], we employ clustering of the particles. The prediction step remains the same as in the case of single targets as defined in Eq. 10. If all targets follow the same motion model, this prediction step is correct as each particle contains the velocity components (\dot{x}_k, \dot{y}_k) of the target it represents. Tracking targets with a different dynamic model can be performed by incorporating Interacting Multiple Models (IMM) [20]. As different targets may have different intensity levels and in TBD the weight update is a function of the target intensity, this results in lower weight assignment to weaker targets. To address this issue we consider each target individually in the update step and Eq. 12 can be re-written for multiple targets TBD as

$$\omega_{k|k-1}^{n,t} = \frac{\prod_{i=C_i}(\mathbf{x}_{k|k-1}^{n,t}) \prod_{j=C_j}(\mathbf{x}_{k|k-1}^{n,t}) p(z_k(i, j)|\mathbf{x}_k^n)}{\sum_{n=1}^{N_t} \omega_{k|k-1}^{n,t} \sum_{n=1}^{L_k+J_k} \omega_{k|k-1}^n}, \quad (14)$$

where $\mathbf{x}_{k|k-1}^{n,t}$ is the n^{th} particle at time k belonging to t^{th} target. Here the component $\sum_{n=1}^{L_k+J_k} \omega_{k|k-1}^n$ is used to further

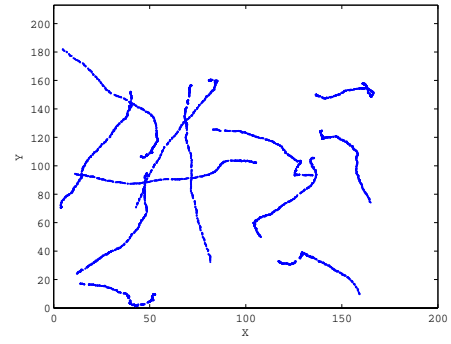


Fig. 4. Tracking results on 500 frames of simulated data for 12 targets.

normalize the weights between 0 to 1 instead of number of targets N_k as there are some particles generated using another proposal density $p(\mathbf{x}_k|Z_k)$. Figure 3 shows a comparison between evolution of particle weights with and without the proposed update strategy. It can be seen that without the proposed update strategy (Fig. 3(a)), one of the targets is completely missed while another one has very low weight and is lost in the next frame. Following the update step, the particles are clustered using mean-shift.

C. Particle clustering

We perform particle clustering using mean-shift for the association of the identity to each particle. Mean-shift clustering climbs the gradient of a probability distribution to find the nearest dominant mode or peak ([21]). Mean-shift is preferred here as it is a nonparametric clustering technique that does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters.

Given $L_k + J_k$ particles $\{\mathbf{x}_k^n, n = 1, \dots, L_k + J_k\}$ on a 2-dimensional space \mathbb{R}^2 using (x_k, y_k) only, the multivariate kernel density estimate obtained with kernel $\mathbf{K}(\mathbf{x})$ and bandwidth h is

$$f(\mathbf{x}_k) = \frac{1}{(L_k + J_k)h^2} \sum_{n=1}^{L_k+J_k} \mathbf{K}\left(\frac{\mathbf{x} - \mathbf{x}_k^n}{h}\right). \quad (15)$$

The bandwidth h is set as $h = 2q_1$ i.e. based on the target covariance Q (see Eq. 5). The mean-shift algorithm tends to maximize the density whose modes are located at the zeros of the gradient $\nabla f(\mathbf{x}_k) = 0$.

After clustering, a cluster merging process is performed to fuse similar clusters. The fusion is based on the proximity $\gamma_{o,p} < \lambda_\mu$ and $\beta_{o,p} < \lambda_A$, where λ_μ and λ_A are the mean and covariance thresholds, $\gamma_{o,p}$ is the Euclidean distance between clusters o and p , and $\beta_{o,p}$ is the covariance of the merged cluster. Finally, an identity is assigned to each particle based on their cluster membership. If all the particles in a cluster are new-born, then a new identity is issued; otherwise all cluster members are assigned the identity with the highest population within that cluster.

D. Resampling

To avoid the degeneracy problem [8], we resample the particles. Resampling is performed according to the particle

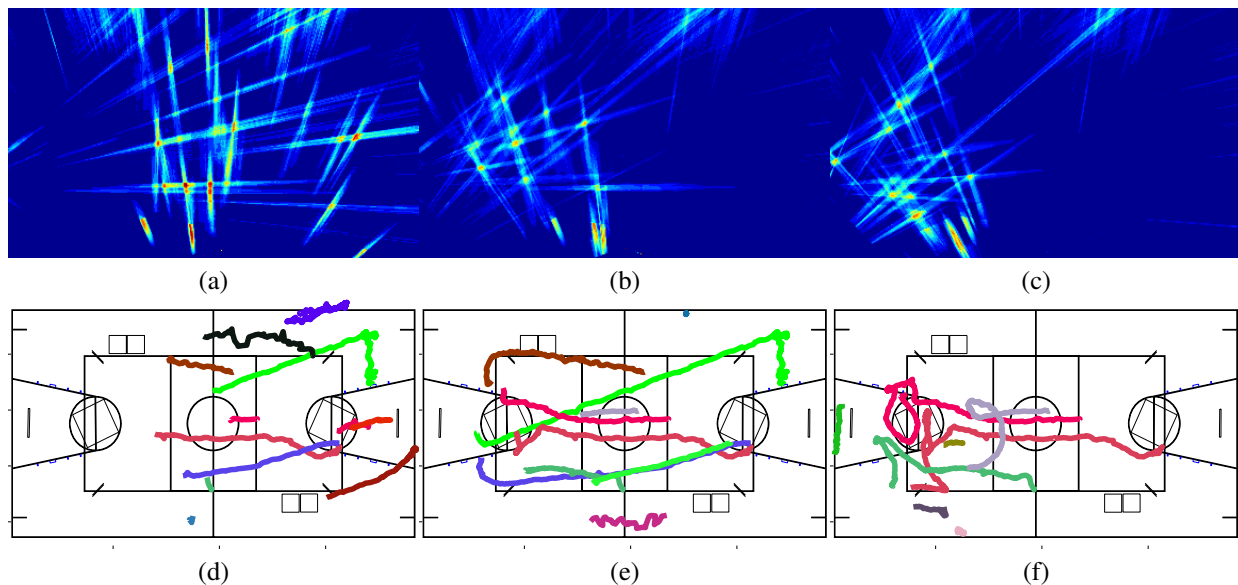


Fig. 5. Sample fusion and multi-target tracking results on top view for frames 500, 590 and 765. (First row) Projection on the top-view. (Second row) Tracking results obtained with the proposed approach.

weights. Here again the single target resampling strategy based on the cumulative distribution function *cpdf* of particle weights will not work as it is insensitive to the particle location. Particles with lower weights (such as those associated to newborn targets) will not be able to have enough representation in the mixture distribution. This will create a hindrance in initializing new tracks in the presence of existing targets. To this extent, the resampling is performed individually for each cluster of particles utilizing weights associated to its members.

III. EXPERIMENTAL RESULTS

We have evaluated the algorithm on synthetic and real datasets. The synthetic data consisted of 12 simulated targets moving with moderate speed with some maneuvering. The real data¹ consist of a basketball match scenario captured using five partially overlapping cameras (Fig. 2(a-f)) and two top-mounted with fish eye lenses. Each video is of resolution 800×850 recorded at 25Hz. There are total 12 targets in the video (10 players and 2 referee). The players have similar appearances and are difficult to distinguish from the background color.

Figure 4 shows tracking results on 500 frames of synthetic data. Typical acceleration and noise in target intensity were set to be $q_1 = 0.001$ and $q_2 = 0.1$ respectively, whereas, the amount of blurring introduced by the sensor was 0.2. These values allow tracking of targets under low SNR values. In this data several targets start very close together and cross each other after some interval. To obtain individual non-merged tracks without false detections the values chosen for minimum target weight was $\lambda_\omega = 1e - 5$ whereas the thresholds for mean distance and variance for cluster merging was $\lambda_\mu = 1$ and $\lambda_A = 2$. The bandwidth chosen for mean-shift was $h = 5$

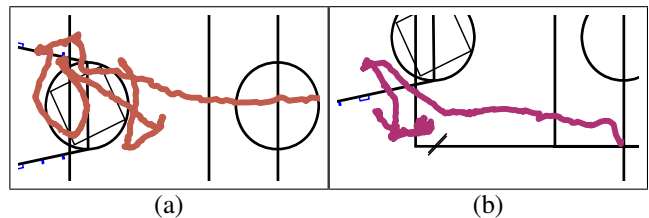


Fig. 6. Example tracks of high maneuvering targets. (a) 5 maneuvers (b) 3 maneuvers.

which is appropriate to cluster particles generated around the target which is effected by a blurring with $A = 0.37$ while using $(\Delta_x, \Delta_y) = (1, 1)$. The tracking was done using 4000 particles per target.

The visualization of results from the proposed approach on real data is in Fig. 5. The projection of the detection mask on top-view using multi-layer homography is shown in Fig. 5(a-c). Several issues regarding the data can be observed from these results. First all the targets are not represented by the same level and spread of intensity values. Some targets have very low visibility without significant amount of spread among neighboring pixels, whereas, others have high intensity values which vary over time. The parallax error can also be easily observed due to which targets have different amount of noisy spread of intensity values in different regions. The shift in intensity values primarily due to increase in cameras overlap is also clearly visible in these projections. These issues makes the less visible targets challenging to track. The tracking results obtained through the proposed multi-target particle filtering track-before-detect (MT-PF-TBD) technique are shown in Fig. 5(d-f). The tracks are shown on a schematic of a basketball court for clarity. Most of these targets maneuver highly as these are players who rapidly change their paths based on the location of the ball and flow of the game (see Fig 6). Although

¹<http://www.apidis.org/Dataset/>, Last accessed: 14 April 2009

we used a constant velocity model, the proposed tracker can still handle maneuvering targets as we model acceleration with a higher value than in the synthetic experiments. Furthermore, the distribution $p(\mathbf{x}_k|Z_k)$ generates a number of newborn particles proportional to the measurement and also helps coping with maneuvering targets. The high value of q_1 also allows the tracker to quickly concentrate around new-born targets which usually do not start with initial zero velocity. However, this also increases the spread of particles around the target and results in target merging. This merging was minimized by using the kernel bandwidth $h = 5$ for mean-shift as in the case of synthetic targets. The remaining parameters were the same as for the synthetic data. These parameters are valid for the sub-sampled version of the data having resolution of 388×225 . The generated tracks appears smooth due to using 3000 particles per target. Using fewer particles can result in jerky tracks.

IV. CONCLUSIONS

A multi-target tracking algorithm has been presented that estimate the location of low visibility targets under noise and is applied in a multi-sensor scenario. The algorithm considers the entire signal as a measurement and extends the target state with signal intensity. The multi-target track-before-detect particle filtering is performed by clustering particles using mean-shift. Then a cluster-based weight update and resampling strategies are proposed to avoid loss of tracks. As future work, to better disambiguate merging targets, we will apply the clustering in a 4D space that also incorporates target velocity components.

REFERENCES

- [1] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey," *Image Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 294–307, March 2005.
- [2] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proc. of Int. Conf. on Computer Vision Systems*, Nice, FR, October 2003.
- [3] S. M. Khan and M. Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," in *Proc. of the European Conf. on Computer Vision*, Graz, AT, May 2006.
- [4] N. D. D. Delannay and C. D. Vleeschouwer, "Detection and recognition of sports (wo)man from multiple views," in *Proc. of ACM/IEEE Int. Conf. on Distributed Smart Cameras*, Como, IT, 30 August–02 September 2009.
- [5] M. Fallon and S. J. Godsill, "Multi target acoustic source tracking using track before detect," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, October 2007.
- [6] I. Garvanov and C. Kabakchiev, "Sensitivity of track before detect multiradar system toward the error measurements of target parameters," *Cybernetics and Information Technologies*, vol. 7, no. 2, January 2007.
- [7] M. G. S. Bruno and J. M. F. Moura, "Multiframe detector/tracker: optimal performance," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 37, no. 3, pp. 925–945, 2001.
- [8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.
- [9] E. Maggio, F. Smeraldi, and A. Cavallaro, "Adaptive multifeature tracking in a particle filtering framework," *IEEE Trans. on Circuits System and Video Technology*, vol. 17, no. 10, pp. 1348–1359, 2007.
- [10] T.-L. Liu and H.-T. Chen, "Real-time tracking using trust-region methods," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 397–402, March 2004.
- [11] J. Vermaak, A. Doucet, and P. Perez, "Maintaining multimodality through mixture tracking," in *Proc. of IEEE Int. Conf. on Computer Vision*, vol. 2, Nice, FR, October 2003, pp. 1110–1116.
- [12] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *European Conf. on Computer Vision*, vol. 1, Prague, CZ, 2004, pp. 28–39.
- [13] J. Czyz, B. Ristic, and B. Macq, "A particle filter for joint detection and tracking of color objects," *Elsevier Journal of Image and Vision Computing*, vol. 25, pp. 1271–1281, July 2006.
- [14] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. London, UK: Artech House, 2004.
- [15] D. J. Salmond and H. Birch, "A particle filter for track-before-detect," in *Proc. of the American Control Conference*, Arlington, VA, USA, June 25–27 2001.
- [16] Y. Boers and J. Driessen, "Multitarget particle filter track before detect application," *IEE Proc.-Radar Sonar Navig.*, vol. 151, no. 6, pp. 1271–1281, December 2004.
- [17] O. Nichten and S. R. Rotman, "Parameter adjustment for a dynamic programming track-before-detect-based target detection algorithm," *EURASIP Journal on Advances in Signal Processing*, January 2008.
- [18] M. Rutten, N. Gordon, and S. Maskell, "Recursive track-before-detect with target amplitude fluctuations," *IEEE Trans. on Radar, Sonar Navigation*, vol. 152, no. 5, pp. 345–352, October 2005.
- [19] K. Punithakumar, T. Kirubarajan, and A. Sinha, "A sequential monte carlo probability hypothesis density algorithm for multitarget track-before-detect," in *Proc. of SPIE*, September 2005.
- [20] Z. Jia, A. Balasuriya, and S. Challa, "Vision based data fusion for autonomous vehicles target tracking using interacting multiple dynamic models," *Elsevier Journal of Computer Vision and Image Understanding*, vol. 109, no. 1, pp. 1–21, 2008.
- [21] D. Comaniciu and P. Meer, "Distribution free decomposition of multivariate data," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 2, no. 1, pp. 22–30, 1999.