Automatic Player Detection, Labeling and Tracking in Broadcast Soccer Video

Jia Liu[1,2], Xiaofeng Tong[3], Wenlong Li[3], Tao Wang[3], Yimin Zhang[3],
Hongqi Wang[1], Bo Yang[4], Lifeng Sun[4], Shiqiang Yang[4]

[1]Institute of Electronics, Chinese Academy of Sciences, Beijing 100080, China
[2]Graduate University of Chinese Academy of Sciences, Beijing 100039, China
[3]Intel China Research Center, Beijing 100080, China
[4]DCST, Tsinghua University, Beijing 100080, China

liujia_iecas@hotmail.com, {xiaofeng.tong, wenlong.li, tao.wang, yimin.zhang}@ intel.com
wiecas@sina.com, yangbo1984@gmail.com, {sunlf, yangshq}@tsinghua.edu.cn

Abstract

Automatic player detection, labeling and tracking in broadcast soccer video are significant while quite challenging tasks. In this paper, we present a solution to perform automatic multiple player detection, unsupervised labeling and efficient tracking. Players' position and scale are determined by a boosting based detector. Players' appearance models are unsupervised learned from hundreds of samples automatically collected by detection. Thereafter, these models can be utilized for player labeling (Team A, Team B and Referee). Player tracking is achieved by Markov Chain Monte Carlo (MCMC) data association. Some data driven dynamics are proposed to improve the Markov chain's efficiency. The testing results on FIFA World Cup 2006 video demonstrate that our method can reach high detection and labeling precision, and reliably tracking in cases of scenes such as multiple player occlusion, moderate camera motion and pose variation.

1 Introduction

Automatic player localization, labeling and tracking is critical for team tactics, player activity analysis and enjoyment in broadcast sports videos. It is quite challenging due to many difficulties such as player-to-player occlusion, similar player appearance, varying number of players, abrupt camera motion, various noises, video blur, etc.

Many algorithms have been presented to deal with the multiple target tracking problem, such as particle filter [1][2], joint probabilistic data association filter (JPDAF) [3], multiple hypothesis tracking (MHT) [4], MCMC data association [5][6] and track linking [7][8][9]. Several researchers also investigated the specific problem of labeling and tracking of players in sports video [2][10][11]. In [10], a clustering based trajectory matching method was proposed to solve the tracking of players in soccer video. In their work, labeling of individuals was achieved by supervised classification. [11] built a track graph, and took the tracking problem as inference in a Bayesian network. In both

of these two works, a multi-camera system was used to get a stationary, high-resolution and wide-field view of soccer game. This setting ensured a reliable background subtraction can be obtained. In our application, the camera is not fixed, which results in moving background. Thus, we need robust and adaptive background modeling and effective object association technologies. In another aspect, unsupervised player labeling is preferred for its generalization ability.

In this paper, we propose a solution for player detection, labeling and tracking in broadcast soccer video. The system framework is illustrated in Figure 1. The whole procedure is a two-pass video scan. In the first scan, we (1) learn video dominant color via accumulated color histograms, and (2) unsupervised learn players' appearance models over hundreds of player samples collected by a boosted player detector. In the second scan, that is the testing procedure, we first use the dominant color for playfield segmentation and view-type classification. Then we apply a boosting player detector to localize players. Afterwards, the players are labeled as Team A, Team B or Referee with prior learned models. Finally, we perform data-driven MCMC association to generate players' trajectories, in which track length, label consistency and motion consistency are used as criterions for associating observations across frames.

The main contributions of our method are: (1) robust player detection achieved by background filtering and a boosted cascade detector; (2) unsupervised player appearance modeling, the referee can be identified in addition to two teams players without any manual labeling; (3) efficient global data association for player tracking, which solves the difficulties like tracker coalescence and correspondence maintaining after occlusions.
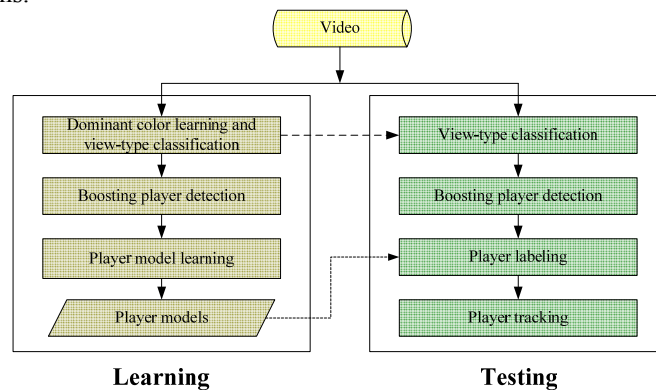

Figure 1: System framework

The paper is organized as follows: Section 2 describes the dominant color learning and view-type classification. Section 3 presents the boosting based player detection method. Player labeling and tracking algorithm are described in section 4 and 5. Experimental results are shown in section 6, and conclusions are made in section 7.

2 Dominant Color Learning and View-type Classification

We first learn the dominant color of background (corresponds to grass color of the playfield) by accumulating HSV color histograms. Then the playfield is extracted

through dominant color segmentation, morphological filtering and connect-component analysis. According to the size of playfield and non-field regions, we use a decision tree to classify each view into one of four pre-defined view types: global view, medium view, close-up, and out of view. The detailed algorithm is described in our previous work [12]. Only global views are fed into player detection module.

3 Player Detection

Player detection is achieved by running a boosted cascade of Haar features [13] on global views. We manually labeled about 6000 players as positive samples. These samples are carefully selected in order to partially capture the appearance variation of players caused by body articulate motion. Negative samples are background patches randomly cropped from soccer video images. Some training samples are shown in Figure 2. All these samples are properly scaled to a resolution of 32×64 pixels.

Figure 2: Some of positive and negative training samples used in training

A boosted cascade detector is then trained on this sample set. To accelerate the training process, we randomly select only a fraction $\varepsilon = 0.01$ of Haar features for each round training as in [14]. This process brings dramatic speed up with only small impact on the final detection performance.

In detection phase, playfield segmentation is first used to filter out the background regions. This process accelerates the detection and reduces false alarms. The detector is then scanned across the filtered image regions at multiple scales. Multiple detections will usually occur around each player after scanning the image. We merge adjacent detected rectangles and get final detections with proper scale and position.

4 Player Labeling

The task of player labeling is to distinguish players' identities (Team A, Team B, Referee). Since now, we use word "player" indiscriminately to represent both team players and referee when there is no confusion. At present, we do not consider the problem of labeling goalkeepers.

4.1 Player Appearance Modeling

To learn the player appearance model, we run the player detector on every 50 frames to collect training samples from the input video. About 500 frames are processed and approximately 1,500 player samples are extracted from a half game video (about 45 mins, 25 fps).

We utilize a *bag of feature* representation of players. This process is illustrated in

Figure 3. For each player sample, the background is subtracted by using the dominant color model firstly, and only the upper body region is used for training. A large pool of pixels is then collected from these regions and transformed into CIE-Luv color space. We estimate a Gaussian Mixture Model (GMM) with *N* components in the space by Expectation-Maximization (EM) [15] clustering. Centers of these components are named *prototypes*. The adjacent components with small center distance are then merged together. The resultant merged components are called *meta-prototypes*. Each player sample is then represented as a histogram by binning all pixels in upper body region into the corresponding meta-prototype.
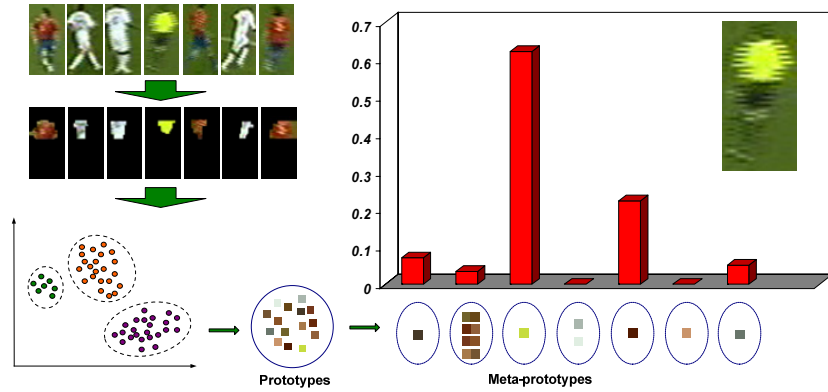

Figure 3: Bag of feature representation of players

Then we learn the appearance model for each identity (Team A, Team B, Referee). A naive clustering on player samples is not enough here. Actually, we manually labeled some samples collected from a real match video, and plotted the first two dimensions of PCA in feature space, as shown in Figure 4. It can be seen that referee samples do not clearly form a cluster in feature space due to their small quantity. A direct clustering may lead them to be absorbed into a nearby cluster. The existing of outliers will also incur an imprecise localization of clusters.

To deal with these problems, we first use EM clustering to estimate *K* clusters over the meta-prototype histogram of all player samples. *K* should be large enough to make sure that the referee samples form at least one cluster. Centers of these *K* clusters are named *sub-models*, which are denoted as $SM = \{sm_i, i = 1, 2, ..., K\}$. Then we merge adjacent clusters into four clusters by hierarchical clustering. Their centers are named *real-models*, and formally denoted as $RM = \{rm_i, i = 1, 2, 3, 4\}$. A labeling function *L* assigns each real-model and sub-model exactly one label in a label set $LS = \{$Team A, Team B, Referee, Outlier$\}$. The two real-models with the largest size are labeled as Team A and Team B, as well as their corresponding sub-models. Then we compute a *minimum average distance* (MAD) from other two real-models to the team sub-models. Denote samples in a real-model as $s_i, i = 1, 2, ..., |rm|$. All the sub-models labeled as Team A or Team B are denoted as $TM = \{tm_j : tm_j \in SM \ and \ L(tm_j) \in \{$Team A,Team B$\}, j = 1, 2...\}$. The MAD is defined as:

$$MAD(rm) = \sum_{i=1}^{|rm|} \min_j (Bd(s_i, tm_j)) \Big/ |rm| \qquad (1)$$

Where *Bd* is the Bhattacharyya distance defined as in [2]:

$$Bd(h_1, h_2) = \left[1 - \sum_{n=1}^{N} \sqrt{h_1(n)h_2(n)} \right]^2 \qquad (2)$$

The real-model with a larger MAD is labeled as Referee, and the other one is labeled as Outlier. All the sub-models labeled as Outlier are discarded and never used for future testing, i.e., We only maintain a set of sub-models take label from label set $LS = \{\text{Team A, Team B, Referee}\}$.
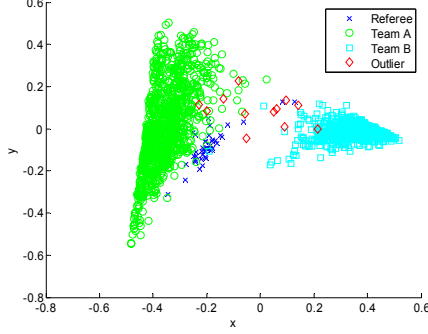

Figure 4: Scatter plot of first two dimensions of PCA in feature space

4.2 Player Labeling

In player labeling phase, each player sample is represented by its meta-prototype histogram, which is denoted as s. We calculate the Bhattacharyya distance between s and each sub-model sm as in formula (2). The sample is assigned the sub-model's label with the nearest distance (nearest neighbor).

5 Tracking

5.1 Problem Formulation

We formulate the tracking of multiple players as a data association problem. The whole detection and labeling result over a time period $[1, T]$ is taken as the observation set, which is denote as $Z_{1:T}$. Let Z_t be the observations at time t, and Z_t^i be the i th observation at time t. To reduce the computational complexity, we define a neighborhood graph $G = (V, E)$ on the observation set, with each node in the graph represents a single observation, and edges defined between neighboring nodes. The neighboring of nodes is defined as:

$$N = \left\{ (Z_{t1}^i, Z_{t2}^j) : \left\| Z_{t1}^i - Z_{t2}^j \right\| \le \left\| t1 - t2 \right\| \times v_{\max} \text{ and } \left\| t1 - t2 \right\| \le T_{\max} \right\} \qquad (3)$$

where $v_{\max}$ is the maximum speed of targets and $T_{\max}$ is the maximum time interval of consecutive observations in a track. A partition of such neighborhood graph can be represented as $\omega = \{\tau_0, \tau_1, \tau_2, ..., \tau_K\}$, where τ_0 denotes false alarms and τ_k denotes the k th track. τ_k takes the form $\tau_k = \{\tau_k(t_1), \tau_k(t_2), ..., \tau_k(t_{|\tau_k|})\}$, where $\tau_k(t_n)$ is an observation at time t_n. The global optimal association can be represented by a partition ω^* which maximizes a posterior of ω given observations:

$$\omega^* = \arg\max \left(P(\omega \mid Z) \right) \qquad (4)$$

The posterior is formulated in a Gibbs distribution form as follows:

$$P(\omega \mid Z) = \frac{1}{C} \exp\left[\left(-\sum_{k=1}^{K} U(\tau_k) - \sum_{i \neq j} V(\tau_i, \tau_j)\right)\middle/ T\right] \qquad (5)$$

where C is normalizing constant, T is temperature, U is a potential based on temporal compatibility within a single track, and V is a potential based on spatial compatibility between different tracks.

The potential U of a track τ_k is formulated as:

$$U(\tau_k) = U_{length}(\tau_k) + U_{label}(\tau_k) + U_{motion}(\tau_k) \qquad (6)$$

with U_{length} penalizes short track, U_{label} favors consistent labeling, and U_{motion} penalizes inconsistent motion and long range association. Let $T_d(\tau_k)$ be the duration of τ_k, L_{th} be the expected length of tracks, U_{length} is defined as:

$$U_{length}(\tau_k) = \begin{cases} \alpha\left[1 - 2 \cdot \left(T_d(\tau_k)/L_{th}\right)^2\right], & \text{if } T_d(\tau_k) < L_{th} \\ -\alpha, & \text{if } T_d(\tau_k) \geq L_{th} \end{cases} \qquad (7)$$

We define the label of a track to be the most frequently appeared observation label in the track. The label potential U_{label} is defined as:

$$U_{label}(\tau_k) = \beta \sum_{i=1}^{|\tau_k|} U_l(L(\tau_k(t_i)) \mid L(\tau_k)) \qquad (8)$$

where L is the label of an observation or a track. U_l is defined as:

$$U_l = \begin{cases} -\varpi_s, & \text{if } L(\tau_k(t_i)) = L(\tau_k) \\ \varpi_d, & \text{if } L(\tau_k(t_i)) \neq L(\tau_k) \end{cases} \qquad (9)$$

where ϖ_s and ϖ_d are positive potential. The motion potential U_{motion} is defined as:

$$U_{motion}(\tau_k) = \sum_{i=1}^{|\tau_k|-1} U_m(\tau_k(t_{i+1}) \mid \tau_k(t_i))$$
$$= \sum_{i=1}^{|\tau_k|-1} \left\{ \varepsilon \cdot \left[\|\tau_k(t_{i+1}) - \tau_k(t_i)\|/\left[(t_{i+1} - t_i) \times v_{max}\right]\right]^2 + \sigma \cdot \left[(t_{i+1} - t_i)/T_{max}\right]^2 \right\} \qquad (10)$$

The potential term V penalizes spatial overlapping between tracks with same label:

$$V(\tau_i, \tau_j) = \begin{cases} \sum_{m=1}^{|\tau_i|} \sum_{n=1}^{|\tau_j|} \theta\rho(\tau_i(m), \tau_j(n)) & \text{if } L(\tau_i) = L(\tau_j) \\ 0 & \text{if } L(\tau_i) \neq L(\tau_j) \end{cases} \qquad (11)$$

where ρ is the spatial overlap between two observation nodes.

5.2 MCMC Data Association

It is usually impossible to compute the global optimal solution for (5) analytically. We adopt MCMC strategy to explore the solution space, and estimate the optimal solution by a simulated annealing scheme. Here we use Metropolis—Hastings (MH) sampler [16] as the MCMC sampler. The design of moves in MH algorithm is crucial to the efficiency of Markov chain. Our move set M consists of seven types of moves (shown in Figure 5): $M = \{birth, death, extension, reduction, split, merge\}$. They are similar to those used in [6]. These moves are grouped into reversible pairs: $birth / death$ pair, $extension / reduction$ pair and $split / merge$ pair. Now we describe these move pairs and deduce the proposal ratio for them. We denote current state as $\omega = (\tau_0, \tau_1, \tau_2, ..., \tau_K)$ and proposed state as ω' in each move. Proposal ratio for each move is denoted as R_m, $m \in M$.

(1) Birth/Death Move Pairs

The birth/death move consists of adding/removing a track from current state. For birth move, denote proposed state as $\omega' = (\tau_0', \tau_1, \tau_2, ..., \tau_K, \tau_{K+1}')$. First, we select u.a.r

(uniform at random) a node $z_c \in \tau_0$ in ω, associate it to a new track τ'_{K+1}. An extension direction flag d is selected u.a.r from direction set $D=\{forward, backward\}$. Then we start an association procedure: For each $z_i \in N^d(z_c)$, assign an association probability $p_d(z_i)$. Choose z_i with probability $p_d(z_i)$, associate it to track τ'_{K+1}, and set $z_c = z_i$. Continue the association with probability γ ($0 < \gamma < 1$). Repeat this procedure n times until an association is rejected or no observation can be added. For death move, denote proposed state as $\omega' = (\tau'_0, \tau_1, \tau_2, ..., \tau_{K-1})$. We select u.a.r a track τ_k in ω and remove it. The proposal ratios are given by:

$$R_{birth}(\omega \to \omega') = 2|\tau_0|\Big/\Big[(K+1)p_{end}\gamma^{n-1}\prod_{s=1}^{n}p_d(z_i^s)\Big] \tag{12}$$

$$R_{death}(\omega \to \omega') = \frac{K\gamma^{(|\tau_K|-2)}\left\{p_{end-forward}\prod_{s=1}^{(|\tau_K|-1)}p_{forward}(z_i^s) + p_{end-backward}\prod_{s=1}^{(|\tau_K|-1)}p_{backward}(z_i^s)\right\}}{2(|\tau_0|+|\tau_K|)} \tag{13}$$

where p_{end} is the probability of ending association, $p_{end-forward}$ and $p_{end-backward}$ are the probability of ending association at tail and head. The values of these probabilities are 1 if no observation can be added or $1-\gamma$ otherwise.
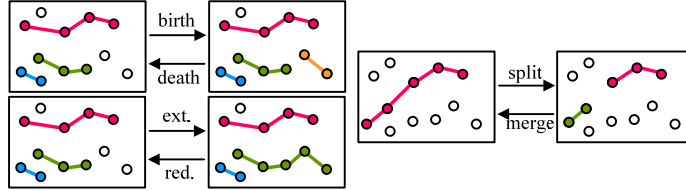

Figure 5: MCMC moves used in proposal

(2) Extension/Reduction Move Pairs

The extension/reduction move consists of extending/reducing a track in one direction. For extension move, denote proposed state as $\omega' = (\tau'_0, \tau_1, \tau_2, ..., \tau'_K)$. First, a track τ_k is selected u.a.r in ω and extension direction flag d is selected u.a.r from D. Then we repeat the same association procedure n times as in birth move on d. For reduction move, denote proposed state as $\omega' = (\tau'_0, \tau_1, \tau_2, ..., \tau'_K)$. First, a track τ_k is selected u.a.r in ω. Then a cutting index is selected u.a.r from $\{2, 3, …, |\tau_k|-1\}$. A reduction direction flag d is selected u.a.r from D, and all the observations on this direction are removed from the track. The proposal ratios are given by:

$$R_{extension}(\omega \to \omega') = 1\Big/\Big[(|\tau_K|+n-2)p_{end}\gamma^{n-1}\prod_{s=1}^{n}p_d(z_i^s)\Big] \tag{14}$$

$$R_{reduction}(\omega \to \omega') = (|\tau_k|-2)p_{end}\gamma^{|\tau'_k|-|\tau_k|-1}\prod_{s=1}^{|\tau'_k|-|\tau_k|}p_d(z_i^s) \tag{15}$$

(3) Split/Merge Move Pairs

The split/merge move consists of splitting a single track or merging two neighboring tracks. For a split move, denote proposed state as $\omega' = (\tau_0, \tau_1, \tau_2, ..., \tau'_K, \tau'_{K+1})$. First, we select u.a.r a track τ_k in ω. Then select a split point t_{split} with probability $p(\tau_k(t_{split}))$ and split the track into two new tracks. For a merge move, denote proposed state as $\omega' = (\tau_0, \tau_1, \tau_2, ..., \tau'_{K-1})$. The move select two tracks τ_i and τ_j with probability $p_{merge}(\tau_i, \tau_j)$ from all the pair candidates in ω, and merge them into a single track τ'_{K-1}. The proposal ratios are given by:

$$R_{split}(\omega \to \omega') = p_{merge} \cdot K\Big/ p(\tau_k(t_{split})) \tag{16}$$

$$R_{merge}(\omega \to \omega') = p(\tau'_{K-1}(t_{split}))\Big/\Big[(K-1)\cdot p_{merge}\Big] \tag{17}$$

5.3 Data Driven Proposal

In order to increase the efficiency of moves, proposals are all driven by the observation data. Let $p_{asso}(z_{t1}^i, z_{t2}^j)$ be an association likelihood between two neighboring observation node z_{t1}^i and z_{t2}^j, we define $p_{asso}(z_{t1}^i, z_{t2}^j)$ as:

$$p_{asso}(z_{t1}^i, z_{t2}^j) = \exp\left\{ -\eta_1 \delta(L(z_{t1}^i) \neq L(z_{t2}^j)) - \eta_2 U_m(z_{t1}^i \mid z_{t2}^j) \right\} \tag{18}$$

where δ is the Dirac function, and U_m is defined as in (10). Denote neighboring relation between tracks as $\tau_m \sim \tau_n$, Then $p_d(z_i)$, $p(\tau(t_{split}))$ and $p_{merge}(\tau_i, \tau_j)$ in proposals take the formulation as follows:

$$p_d(z_i) = p_{asso}(z_i, z_c) \Big/ \sum\nolimits_{z_j \in N^d(z_c)} p_{asso}(z_j, z_c) \tag{19}$$

$$p(\tau(t_{split})) = (1 - p_{asso}(\tau(t_{split}+1), \tau(t_{split}))) \Big/ \sum\nolimits_{i=2}^{|\tau|-1} (1 - p_{asso}(\tau(t_{i+1}), \tau(t_i))) \tag{20}$$

$$p_{merge}(\tau_i, \tau_j) = p_{asso}(\tau_i(t_{|\tau_i|}), \tau_j(t_1)) \Big/ \sum\nolimits_{\tau_m \sim \tau_n} p_{asso}(\tau_m(t_{|\tau_m|}), \tau_n(t_1)) \tag{21}$$

6 Results

We test our algorithm on World Cup 2006 MPEG2 videos (image size: 720x576, 25 fps). First, our detection algorithm (without tracking) is tested on the France vs. Spain and Brazil vs. Japan videos. We randomly select hundreds of frames from each video (50-100 frames interval between adjacent selected frames) and manually labeled the ground truth. The performance of detection is shown in Table 1.

Video	Correct	False	Missing	Precision(%)	Recall(%)	F-score(%)
FRA vs. SPA	1133	145	96	88.65	92.19	90.39
BRA vs. JAP	5927	489	746	92.38	88.82	90.57

Table 1: Detection performance

We also test our labeling performance on the ground truth. Only those correctly detected players are used for evaluation. The confusion matrix of labeling is shown in Table 2, with GT stands for *ground truth*, INF stands for *inferred*. R denotes Referee, TA denotes Team A and TB denotes Team B. All the percentages are row-normalized.

INF / GT	R (%)	TA(%)	TB(%)
R	82.93	17.07	0.00
TA	0.38	98.30	1.32
TB	0.00	2.31	97.69

INF / GT	R (%)	TA(%)	TB(%)
R	97.12	2.88	0.00
TA	1.23	97.75	1.02
TB	3.18	1.23	95.59

(a) FRA vs. SPA   (b) BRA vs. JAP

Table 2: Confusion matrix of labeling with percentages row-normalized

We manually labeled a global view clip consisting of 100 consecutive frames from France vs. Spain video and test our tracking algorithm. The input observation set of these frames is shown in Figure 6(b). Some tracking results are shown in Figure 6(a). Automatic tracked players are enclosed by rectangles. The red rectangles denote "Team A", blue denote "Team B" and yellow denote "Referee". The ID of player is shown on

the top of each rectangle. It can be noticed that occlusion occurs between ID12, ID14 and ID18 from frame 40 to frame 70. Our algorithm correctly tracks and classifies them even if ID18 is totally occluded by ID12 in frame 51. Occlusion also occurs between ID8 and ID11 in frame 40, and between ID6 and ID7 in frame 70, which are all correctly handled by the algorithm.

Figure 6(c) shows trajectories generated by tracking algorithm, with totally 24 players entering/leaving the scene. Our tracking algorithm can reach a precision of 99.32% and recall of 94.43% on the ground truth. Most miss detection is due to failure of boosting detection in several consecutive frames caused by video blur. It also can be noticed that the performance of labeling is significantly improved after tracking.
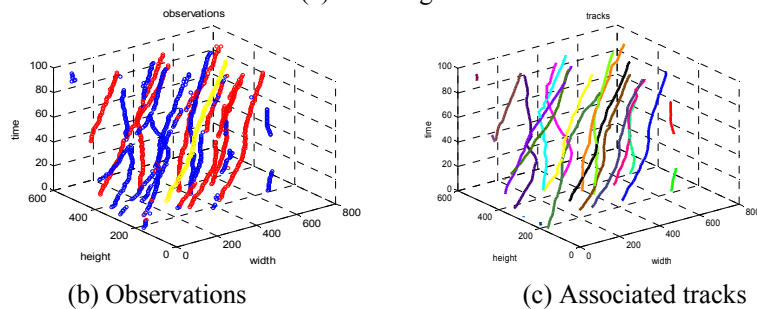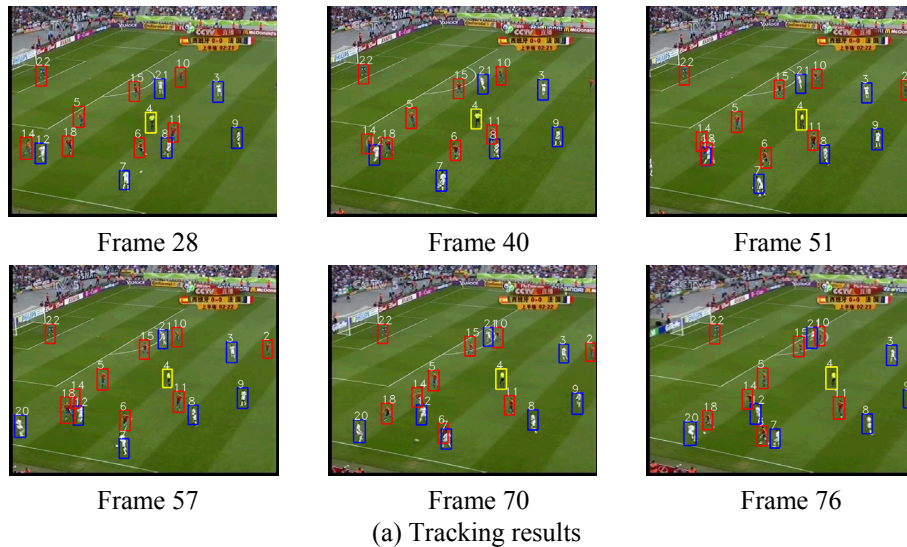

| Frame 28 | Frame 40 | Frame 51 |

| Frame 57 | Frame 70 | Frame 76 |

(a) Tracking results

(b) Observations   (c) Associated tracks

Figure 6: Tracking results and generated trajectories

7 Conclusion

In this paper, we proposed an automatic player detection, unsupervised labeling and efficient player tracking method applied for broadcast soccer videos. The detection module combines background modeling and boosting detection. Labeling is achieved through unsupervised player appearance learning. MCMC data association is applied for tracking players. The result can be utilized for team tactics and player activity analysis, high-light detection, etc. The method can also be applied to other applications such as surveillance and vision-based human-computer interaction. While most of players can

be detected and tracked by our method, some cases such as long occlusions, serious video blur, abrupt camera motion and player tangle may still lead to failure. We plan to design more efficient MCMC proposals, and improve the labeling and tracking performance by playfield registration and trajectories inference in future works.

References

[1]   M. Isard and J. MacCormick, "BraMBLe: A Bayesian Multiple-Blob Tracker," In Proc. IEEE Int'l Conf. on Computer Vision (ICCV), 2001.

[2]   K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A Boosted Particle Filter: Multitarget Detection and Tracking," In Proc. European Conf. on Computer Vision (ECCV), 2004

[3]   Y. Bar-Shalom and T. Fortmann, Tracking and Data Association. San Diego, CA: Academic Press, 1988.

[4]   D. Reid, "An Algorithm For Tracking Multiple Targets," IEEE Transaction on Automatic Control, vol. 24, no. 6, pp. 843–854, December 1979.

[5]   S. Oh, S. Russell, and S. Sastry, "Markov Chain Monte Carlo Data Association for Multiple-Target Tracking," University of California, Berkeley, Technical Report UCB//ERL M05/19, June 2005.

[6]   Q. Yu, G. Medioni, "Map-Enhanced Detection and Tracking from a Moving Platform with Local and Global Data Association," IEEE Workshop on Motion and Video Computing (WMVC'07), 2007.

[7]   A.G.A. Perera, C. Srinivas, A. Hoogs, G. Brooksby, Wensheng Hu, "Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions," IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR), 2006.

[8]   C. Stauffer, "Estimating Track Sources and Sinks", In Proc. IEEE Workshop on Event Mining in Video, 2003.

[9]   R. Kaucic, A. G. A. Perera, G. Brooksby, J. Kaufhold, A. Hoogs, "A Unified Framework for Tracking Through Occlusions and Across Sensor Gaps", In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR), 2005.

[10] J. Sullivan and S. Carlsson, "Tracking and Labeling of Interacting Multiple Targets," In Proc. European Conf. on Computer Vision (ECCV), 2006.

[11] P. Nillius, J. Sullivan and S. Carlsson, "Multi-Target Tracking – Linking Identities using Bayesian Network Inference," In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR), 2006.

[12] J. Li, T. Wang, W. Hu, Y. Zhang, "Soccer Highlight Detection using Two-Dependent Bayesian Network," In Proc. of IEEE International Conf. on Multimedia & Expo (ICME), 2006.

[13] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," In Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recognition (CVPR), 2001.

[14] J. Shotton, J. Winn, C. Rother, A. Criminisi, "TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation," In Proc. European Conf. on Computer Vision (ECCV), 2006.

[15] J.A. Bilmes, "A Gentle Tutorial of the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," TR-97-021, Berkeley, April 1998.

[16] J.S. Liu, Monte Carlo Strategies in Scientific Computing, Springer, 2001.