Online Learned Player Recognition Model Based Soccer Player Tracking and Labeling for Long-Shot Scenes

Weicun Xu†, Qingjie Zhao†, Yuxia Wang†, Nonmembers, and Xuanya Li††, Student Member

SUMMARY Soccer player tracking and labeling suffer from the similar appearance of the players in the same team, especially in long-shot scenes where the faces and numbers of the players are too blurry to identify.

In this paper, we propose an efficient multi-player tracking system. The tracking system takes the detection responses of a human detector as inputs. To realize real-time player detection, we generate a spatial proposal to minimize the scanning scope of the detector. The tracking system utilizes the discriminative appearance models trained using the online Boosting method to reduce data-association ambiguity caused by the appearance similarity of the players. We also propose to build an online learned player recognition model which can be embedded in the tracking system to approach online player recognition and labeling in tracking applications for long-shot scenes by two stages. At the first stage, to build the model, we utilize the fast k-means clustering method instead of classic k-means clustering to build and update a visual word vocabulary in an efficient online manner, using the informative descriptors extracted from the training samples drawn at each time step of multi-player tracking. The first stage finishes when the vocabulary is ready. At the second stage, given the obtained visual word vocabulary, an incremental vector quantization strategy is used to recognize and label each tracked player. We also perform importance recognition validation to avoid mistakenly recognizing an outlier, namely, people we do not need to recognize, as a player. Both quantitative and qualitative experimental results on the long-shot video clips of a real soccer game video demonstrate that, the proposed player recognition model performs much better than some state-of-the-art online learned models, and our tracking system also performs quite effectively even under very complicated situations.

key words: soccer player recognition, player tracking and labeling, online learning, visual words, fast k-means clustering

1. Introduction

In this paper, we concentrate on soccer player tracking and labeling in long-shot scenes of soccer game videos. Target identity labeling during the tracking process is very necessary in many specific scenes. For example, player identity labeling in soccer game broadcast can help the audience to find one specific player in long-shot scenes much easier, making the soccer matches live on television or the Internet more enjoyable; moreover, by tracking the labeled players, the motion data of each player can be obtained and used to analyze their status and behavior.

An enormous volume of literature has been devoted to player tracking in sports videos. However, target identity labeling is not considered in most tracking methods proposed before.

Okuma et al. [1] proposed a multi-layer color model for hockey player tracking. Liu et al. [2] proposed a soccer player tracking system which could classify players into different teams. Mazzeo et al. [3] concentrated on automatic soccer player classification in multi-camera environments. The player tracking system proposed by Kasuya et al. [4] reduces the effects of occlusion, interaction and shadow by using two fixed cameras to capture the soccer video. However, the player recognition problem is not involved in the works mentioned above.

Sullivan et al. [5] proposed an effective player tracking system running on high-resolution videos captured by several fixed cameras. The proposed tracking system effectively identifies the soccer players by building histograms summarizing the players’ positions relative to their teammates from long-term tracks. However, it is difficult to obtain long-term tracks for each player to build reliable relative position histograms from common broadcast videos, as the tracks tends to be interrupted frequently by switches of camera, or departure of the players.

Bertini et al. [6] proposed to identify the soccer players by faces, jersey numbers and text annotation in closed shots. However, both the players’ faces and the numbers tend to be very small and blurred in long-shot scenes. Meanwhile, they are not always visible, as pointed out in [7]. Scale-invariant feature transform (SIFT) method [8] and its variants, e.g., the SURF features [9], were applied for object tracking [10], [11] and recognition [12] by many researchers. However, none of the methods mentioned above focuses on the player recognition problem.

The online learning methods, e.g., the online Boosting method [13] and the online Random Forest (ORF) [14] method, are widely used to build target-specific classifiers to realize robust multi-person tracking [15]–[17]. However, the methods mentioned above are prone to be too adaptive to store history data, as they are mainly designed to train appearance models for tracking applications rather than recognition tasks.

Kuo et al. [16] proposed to build Online Learned Discriminative Appearance Models (OLDAMs) for short and reliable track-lets using the AdaBoost method [18]. However, batch learning from extensive training data which maybe necessary to training reliable recognition models will make OLDAMs degenerate to off-line learned models. Lu...
et al. [7] proposed a basketball player identification system based on an appearance model combining color, SIFT visual words [19], and MSER regions [20]. However, due to the expensive time cost of extracting and comparing the SIFT descriptors, the batch clustering process needed to build the visual word vocabulary is prone to be too time intensive to be performed online.

The main goal of this paper is to realize soccer identity labeling in long-shot scenes. However, soccer player recognition is a very challenging task suffering from the appearance similarity of the players, as players of the same team wear uniforms of the same pattern. To achieve this goal, we need to construct an on-line learned identity recognition model and embed the model in the tracking framework. The recognition model should be invariant to 3D affine transformation caused by abrupt view changes when camera switching occurs.

To this end, we build a robust player tracking system and a player recognition model which can be embedded in the tracking system. The player recognition model is built incrementally based on an online learning process using the training samples obtained during player tracking.

Our contributions mainly lie in:

1) We propose to build an online learned player recognition model during the tracking process. Instead of classic k-means batch clustering, we introduce the fast k-means clustering method [21] to build and update a SIFT visual word vocabulary [19] incrementally at each time step, using the training samples obtained during multi-player tracking. The fast k-means clustering method is very suitable for online applications as it uses a mini patch of training data to update the clustering centers at each iteration. Given the obtained detection responses, we first compute the assignment probabilities between the detection responses and the targets existing at last time step accordingly to their affinity in position, scale and appearance as described in Sect. 2.3. Then the goal of multi-player tracking is achieved by computing the optimal configuration between the detection responses and the targets using a MAP estimation process that is described in Sect. 2.4. The flowchart of the tracking framework is shown in Fig. 1.

2) We build an efficient multi-player tracking system. The tracking system takes the detection responses provided by a human detector as inputs and computes the optimal configuration between the detection responses and the tracks using MAP estimation. Differing from other tracking-by-detection methods [1], [7], [15], [16], at each time step, we firstly generate a compact spatial proposal that only covers the areas where the players possibly appear using edge detection. To realize real-time player detection, the human detector only scans the area covered by the spatial proposal instead of globally scanning the whole image.

The remainder of this paper is organized as follows. The player tracking system is described in Sect. 2. Section 3.2 presents a detailed description of the online learned player recognition model. Extensive quantitative and qualitative experimental results on soccer game video sequences are given in Sect. 4. Finally, Sect. 5 concludes the paper and suggests future work.

### 2. The Player Tracking System

#### 2.1 System Overview

At each time step, our player tracking system firstly detects the players using a human detector. To realize real-time player detection, we determine the scanning scope of the human detector according to a spatial proposal that only covers the areas where the players possibly appear, as described in Sect. 2.2.

Given the obtained detection responses, we firstly compute the assignment probabilities between the detection responses and the targets existing at last time step accordingly to their affinity in position, scale and appearance as described in Sect. 2.3. Then the goal of multi-player tracking is achieved by computing the optimal configuration between the detection responses and the targets using a MAP estimation process that is described in Sect. 2.4. The flowchart of the tracking framework is shown in Fig. 1.

#### 2.2 Real-Time Player Detection

In many tracking-by-detection approaches [15], [22], the detection responses are obtained by global scanning using the human detector, which is too time intensive for time-critical online tracking systems.

Instead of global scanning, to realize real-time player detection, a spatial proposal is used by our tracking system to determine the areas scanned by the human detector according to the 2-D spatial distribution of the foreground objects in the scene. To estimate the distribution of the foreground objects, we perform edge detection by using the Canny edge detector [23] which can detect the players while filtering most of the court except the marker lines.

A player may generate many edges. Given the fact that the edges adjacent with each other are possibly from the same object, we firstly partition the edges into several edge-sets by performing an edge-clustering process, and the edges partitioned into the same edge-set are considered from the same object.

At current time step $t$, given the obtained edge-sets represented by $B_t = \{b_i\}$, where $b_i$ represents the rectangular bounding-box surrounding the $i_{th}$ edge-set, we draw searching candidates in the neighborhoods of the edge-sets for player detection.

The spatial proposal determines $Q_{B_t}(c|B_t)$ which denotes the possibility of a sub-region denoted by $c$ being drawn as a searching candidate, according to the degree of
the sub-region overlapping with each edge-set. A sub-region will be drawn as a candidate if it overlaps severely with any edge-set, and otherwise not, as described in Eq. (1):

\[ Q_s(c|B_t) = \max \left\{ 1, \sum_{i} K_s \left( \frac{c \cap b_i}{c} \right) \right\} \tag{1} \]

where \( \gamma \leq 1 \) is a user-defined threshold, and \( K_s(\cdot) \) is a flat kernel [10]:

\[ K_s(t) = \begin{cases} 1, & \text{if } ||t|| \leq u \\ 0, & \text{if } ||t|| > u \end{cases} \tag{2} \]

where \( ||\cdot|| \) represents the Euclidean norm. Figure 2 shows an example of the player detection process.

### 2.3 Assignment Probability Estimation

Let \( x_{i;1:2}^{(j)} = \{x_{i;1}^{(j)}, \ldots, x_{i;2}^{(j)}\} \) represent the \( i_{th} \) target, where \( x_{i;1}^{(j)} = (x, y, s, id) \) represents its state at time \( k \) with \( (x, y) \) representing the 2-D position, \( s \) representing the scale, and \( id \) representing the identity; \( id = 0 \) means that the target is not labeled yet; \( i_1 \) represents the time step at which the target is created; \( i_2 \) (\( i_2 > i_1 \)) represents the most recent time step at which the target is detected.

Let \( X_{t-1} = \{x_{i;1:2}^{(j)} | i_2 \leq t - 1\} \) represent the targets, and \( Y_t = \{y_{t;1}^{(j)}\} \) represent the detection responses provided by the human detector at current time step \( t \), where \( y_{t;1}^{(j)} \) represents the \( j_{th} \) detection response. For sake of brevity, we omit the superscripts indicating the indexes of the targets and the detection responses when there is no confusion in the following.

Similar with [24], the assignment probability, namely, the probability of assigning the \( i_{th} \) target with a detection response \( y_t \) is computed based on the affinity of their position, scale and appearance, represented by \( \mathcal{A}_p(y_t|x_{i;1:2}^{(j)}) \), \( \mathcal{A}_s(y_t|x_{i;1:2}^{(j)}) \), and \( \mathcal{A}_a(y_t|x_{i;1:2}^{(j)}) \), respectively:

\[ P(x_{i;1:2}^{(j)}, y_t) = \mathcal{A}_p(y_t|x_{i;1:2}^{(j)}) \mathcal{A}_s(y_t|x_{i;1:2}^{(j)}) \mathcal{A}_a(y_t|x_{i;1:2}^{(j)}) \tag{3} \]

We estimate \( \mathcal{A}_t(y_t|x_{i;1:2}^{(j)}) \) by using an observation model consisting of two strong classifiers based on different features, both of which are specifically trained for the target by using the online Boosting method [13].

Let \( q_f(y_t) \) represent the observation probability estimated by the first classifier and \( q_s(y_t) \) represent the second one. Then \( \mathcal{A}_r \) can be computed as:

\[ \mathcal{A}_r(y_t|x_{i;1:2}^{(j)}) = q_f(y_t) + q_s(y_t) \tag{4} \]

Definitions of the terms \( q_f(y_t) \) and \( q_s(y_t) \) are presented in the subsection of implementation details.

#### 2.3.1 Implementation Details

Intuitively, the likelihood of two targets being similar in both color and texture is quite low. Therefore, similarly with [15] which showed the advantages of the combination of different features in terms of robustness and accuracy, our first classifier uses a normalized color histogram, and the second one uses the LBP features.

Figure 3 explains the reason why combining the LBP feature and the color feature can improve the distinctive ability of the appearance model. As shown in the figure, the LBP features drawn from two player patches are obviously different, while the patches are similar in colors, which is to say, the appearance model can distinguish targets of similar colors much better by combining the LBP features.

More specifically, given an image patch of a training sample or a detection response, we extract the color features from it by building a normalized color histogram consisting of \( H = H_h \times H_s \times H_c \) blocks under the HSV color space. In our experiments, we set \( H_h = H_s = H_c = 20 \) by experience. Similarly, to extract the LBP features, we firstly divide the image patch into \( 3 \times 6 \) blocks, and then build a 25-dimensional LBP histogram from each block to obtain \( 3 \times 6 \times 25 = 450 \) features.

Let us denote the weight of the \( k_{th} \) weak classifier of the first classifier by \( \tau_f^{(k)} \), and the hypothesis generated by the \( k_{th} \) weak classifier as \( h_f^{(k)} \). Following [13], given a detection response \( y_t \), \( q_f(y_t) \) can be computed as:

\[ q_f(y_t) = \text{sign} \left( \sum_k \tau_f^{(k)} \cdot h_f^{(k)}(y_t) \right) \tag{5} \]

\( q_s(y_t) \) can be computed similarly as in Eq. (5).

#### 2.4 Data Association via MAP Estimation

Let \( \Omega = \{\omega_i|x_{i;1:2}^{(j)} \in X_i\} \) represent a possible configuration between the targets and the detection responses, where \( \omega_i \) represents the index of the detection response assigned to the \( i_{th} \) target in \( \Omega \), and \( \omega_i = 0 \) means that no detection response is assigned to it. The optimal configuration \( \Omega^* \) is determined
by performing a MAP estimation process:
\[
\Omega^* = \arg\max_{\Omega} \Psi(\Omega) \prod_i \left[p_i(x_i^{(t)}, y_i^{(o)}) \right] = \arg\max_{\Omega} \Psi(\Omega) \prod_i \left[\mathcal{A}_r(y_i^{(a)})\mathcal{A}_v(y_i^{(o)})\mathcal{x_i^{(t)}} \right] \times \mathcal{A}_v(y_i^{(o)}|x_i^{(t)})
\]  

(6)

where \(\Psi(\cdot)\) is a function used to determine whether any detection response is assigned to more than one target or not in \(\Omega\). For a set \(S = \{s_i\}\) consisting of real numbers, \(\Psi(S)\) can be computed as follows:
\[
\Psi(S) = \begin{cases} 
1 & \text{if } \prod_{s_i \neq 0} (s_i - s_j) \neq 0, \\
0 & \text{otherwise.}
\end{cases}
\]  

(7)

The detection responses which are not assigned to any existing target will be taken as new targets. A target not assigned with any detection response will be abandoned if it exists less than two time steps, as it is difficult to determine if it is a false alarm or not.

3. Online Learned Player Recognition Model

In this paper, online player recognition and labeling is realized by two stages: 1) tracking for online learning, and 2) learning for online labeling. At each time step of the first stage, we firstly extract training data by performing multi-player tracking, and then build and update the visual word vocabulary for the player recognition model by using the fast k-means clustering method [21]. At the second stage, the obtained vocabulary is used for online player recognition and labeling using our incremental vector quantization strategy. Note that the recognition model is essentially different from the appearance model. The appearance model is used to generate tracks by estimating the assignment likelihood of detection responses and the targets. However, the recognition model is used to determine the identity of each unlabeled track that is newly generated. The flowchart of the player tracking and labeling system is shown in Fig. 5.

Section 3.1 presents the online learning method based on fast k-means clustering. Sections 3.2 and 3.3 present the details of the first stage and the second stage, respectively.

3.1 Online Learning Algorithm

Sculley [21] proposed a fast k-means clustering method using a mini patch of training data to update the clustering centers at each iteration. Fast k-means clustering is very suitable for online applications, and is guaranteed to converge to the optimal clustering centers. Hence, we utilize the fast k-means clustering method to build the visual word vocabulary in an online manner.

Let \(F_t = \{f_i\}\) be new training data obtained at current time step \(t\), and \(w = (r, v)\) be a visual word, where both \(f \in \mathbb{R}^{128}\) and \(r \in \mathbb{R}^{128}\) represent 128-dimensional SIFT descriptors, and \(v\) represents the number of descriptors centering \(w\); \(V_{t-1} = \{w_j\}\) represents the vocabulary obtained at last time step. Let \(\mathcal{K}\) be the expected size of the vocabulary, and \(M(V_{t-1}, f_i)\) be a function returning the index of the visual word in \(V_{t-1}\) nearest to \(f_i\):
\[
M(V_{t-1}, f_i) = \arg\min_j ||r_j - f_i||.
\]  

(8)

Given new training data \(F_t\), we update \(V_{t-1}\) using fast k-means clustering as described in Algorithm 1, where \(\tau > 1\) is a constant value and \(\eta\) represents the updating rate of a visual word that is determined by the number of descriptors centering it. An example of the fast k-means clustering process is shown in Fig. 6.
The vocabulary for the player recognition model, at time $t$, is to estimate the identities of the unlabeled targets denoted $X_t$. The tracking-for-learning process is shown in Fig. 7. After performing multi-player tracking to obtain new samples of the players, we extract the informative descriptors and incrementally update the sub-vocabulary of each player using the descriptors, to get the updated vocabulary $V_t$.

![Image](391x650 to 419x714)

**Fig. 7** The tracking-for-learning process at time $t$. After performing multi-player tracking to obtain new samples of the players, we extract the informative descriptors and incrementally update the sub-vocabulary of each player using the descriptors, to get the updated vocabulary $V_t$.

### Algorithm 1 Online learning via fast k-means clustering

**Input:** $V_{t-1} = \{w_i\}$//the vocabulary obtained at last time step  
**Input:** $F_t = \{f_i\}$//new training data  
**Output:** $V_t$//new vocabulary

1. for each descriptor $f_i \in F_t$ do  
2. \[ l \leftarrow M(V_{t-1}, f_i) \]  
3. \[ k \leftarrow M(V_{t-1} - w_l, f_i) \]  
4. if $|V_{t-1}| \geq K$ or $||f_i - f_l|| < 2\sqrt{	ext{dim}}$ then  
5. \[ \eta \leftarrow \frac{1}{K+t} \]  
6. \[ w_l \leftarrow w_l + \eta f_i \]  
7. \[ w_l = (1-\eta)w_l + \eta f_i \]  
8. else  
9. create a new visual word $\hat{w} = (f_i, 1)$  
10. $V_{t-1} \leftarrow V_{t-1} + \hat{w}$  
11. end if  
12. end for  
13. $V_t \leftarrow V_{t-1}$

### 3.2 Tracking for Online Learning

The tracking-for-learning process at one time step is shown in Fig. 7. As depicted in the figure, to build the visual word vocabulary for the player recognition model, at time $t$ of this stage, we firstly perform multi-player tracking to obtain new samples of the targets, and then extract the informative descriptors from the samples to update the visual word vocabulary using Algorithm 1. A descriptor is considered informative in this paper if it is generated by the same target for at least two consecutive frames. Reasonably, we think that the other descriptors are either false alarms or too unstable to be used as training data. Hence, we only use informative descriptors to build and update the vocabulary.

![Image](437x650 to 465x714)

**Fig. 8** The matched SIFT descriptors of two player patches drawn at adjacent frames.

We independently build and update a visual word sub-vocabulary for each player using the informative descriptors from the corresponding target. To obtain the correspondence between the targets and the players, we manually label the targets when they are created. Let $V_n$ represent the sub-vocabulary of the player with identity $n \in R^+ \cup R^-$; $\text{sign}(n)$ indicates the team which the player belongs to, and $|n|$ indicates the jersey number. The visual word vocabulary consisting of all the sub-vocabularies can be denoted by $\mathcal{V} = \{V_n\}$. We reuse $\mathcal{K}$ to represent the size of each sub-vocabulary:

$$|V_n| = \mathcal{K} \quad (\forall V_n \in \mathcal{V})$$

In practice, it takes $T$ time steps to update the $\mathcal{K}$ visual words of each sub-vocabulary.

#### 3.2.1 Implementation Details

In practice, SIFT matching is used to test whether the descriptors generated by a target at current time step are informative or not: a descriptor is considered informative if it matches with the descriptors extracted at last time step or the next time step. An example of the SIFT matching process is shown in Fig. 8.

Let us reuse $F_t = \{\hat{f}_i\}$ to represent the descriptors from a target at time $t$, and $G(F_t, \hat{f})$ be a function returning the index of the descriptor in $F_t$ matching with a descriptor $\hat{f}$, while $G(F_t, \hat{f}) = 0$ means that $\hat{f}$ does not match with any descriptor in $F_t$. Then a descriptor $f_{i,j} \in F_t$ is informative if:

$$G(F_{t-1}, f_{i,j}) + G(F_{t+1}, f_{i,j}) > 0$$

### 3.3 Learning for Online Labeling

Let $X_t \supseteq \tilde{X}_t = \{x_{i,j}^{0,i} | d_i = 0; t - i \geq L\}$ be the targets that are required to be labeled at time $t$ of the learning-for-online-labeling stage, where $L$ a constant value determining if a target is old enough to be labeled or not. Let $z_i$ represent the identity of the $i_{th}$ target in $X_t$, which is unknown before online labeling. Given the visual word vocabulary $\mathcal{V}$, and the informative descriptors from each target, online labeling is to estimate the identities of the unlabeled targets denoted by $Z = \{z_j | x_{j,t}^{0,j} \in X_t, z_j = id_j \text{ if } id_j > 0\}$ maximizing the objective function:

$$p(Z | \tilde{X}_t) = C^{-1} \cdot \Psi(Z) \prod_{d_i = 0} p(z_i | \{x_{i,j}^{0,i} | j \geq t\})$$

$$= C^{-1} \cdot \Psi(Z) \prod_{d_i = 0} \prod_{j} \{p(z_j | r_j)m(r_j | x_{j,t}^{0,j})\},$$

(10)
where $C$ is a normalization value that can be computed as follows:

$$C = \sum_{i \in I} \sum_{j} m(r_j) x_{i1;j2}^{(0)}.$$  

(11)

$\Psi(Z)$ determines if the identities of any two targets are the same according to Eq. (7). $p(z_i | r_j)$ determines if $r_j$ belongs to $V_{z_i}$:

$$p(z_i | r_j) = \begin{cases} 1 & \text{if } r_j \in V_{z_i}, \\ \epsilon & \text{otherwise.} \end{cases}$$  

(12)

with $\epsilon \ll 1$ being a small positive value; $m(r_j) x_{i1;j2}^{(0)}$ represents the frequency of visual word $r_j$ appearing in the image patches of the $i_{th}$ target, which is estimated by using an incremental vector quantization strategy proposed in the following subsection.

3.3.1 Incremental Vector Quantization

It is reasonable to think that labeling errors caused by statistical noise can be avoided given adequate observations. Hence, we label each unlabeled target based on the SIFT descriptors of the target obtained from the beginning $L$ timesteps of tracking.

For the $i_{th}$ target which is unlabeled, firstly build a frequency histogram denoted by $\mathcal{H} = \{h_i | w_i \in \mathcal{V}\}$ for it when it is created, and then initialize the value of each element in $\mathcal{H}$ to be zero. At each time step $t$ ($t > i$), given new descriptors $F_t$ from the target, we incrementally update the frequency histogram using the method described in Algorithm 2, where $F_{j;i}$ indicates the index of the visual word nearest to the descriptor $f_{j;i} \in F_t$ if $f_{j;i}$ is considered informative:

$$j_{j;i} = \begin{cases} M(V, f_{j;i}) & \text{if } G(F_{t-1}, f_{j;i}) > 0, \\ 0 & \text{otherwise.} \end{cases}$$  

(13)

Algorithm 2 Incremental vector quantization method

Output: Updated frequency histogram;

1: for each descriptor $f_{j;i} \in F_t$ do
2: \hspace{1cm} $j \leftarrow G(F_{t-1}, f_{j;i})$/determine if $f_{j;i}$ is informative
3: \hspace{1cm} if \hspace{0.5cm} $j > 0$ then
4: \hspace{1.5cm} if $j_{j;i} > 0$ then
5: \hspace{2cm} $k_{j;i} = j_{j;i}$
6: \hspace{1cm} \hspace{0.5cm} else
7: \hspace{1.5cm} $h_k = h_k + 1$
8: \hspace{1.5cm} end
9: \hspace{1cm} else
10: \hspace{1.5cm} $j_{j;i} = \text{argmin}_j \| r_i - f_{j;i} \|$//see Eq. (8) and Eq. (13)
11: \hspace{1.5cm} \hspace{0.5cm} $k_{j;i} = j_{j;i}$
12: \hspace{1.5cm} $h_k = h_k + 2$
13: \hspace{1cm} end
14: end if
15: for

The frequency histogram $\mathcal{H}$ will be used to label the target after $L$ times of updating. At time $i + L$, given $\mathcal{H}$, for a visual word $r_j \in \mathcal{V}$, $m(r_j) x_{i1;j2}^{(0)}$ can be computed as:

$$m(r_j) x_{i1;j2}^{(0)} = \sum_{k} m(r_j) x_{i1;j2}^{(0)} = h_{j}$$  

(14)

where $i_2$ is equal to $i + L$.

It is important to note that we do not label interacting targets, since interaction may lead to identity switches making labeling meaningless.

3.3.2 Importance Recognition Validation

In practice, some outliers, namely, people we do not need to track or recognize, may also appear in the scene, e.g., the coaches, the players newly come into the scene after substitution, the referees, the goal keepers, and the detection responses from the stadium area. To avoid mistakenly recognizing an outlier as a player, we introduce a strategy that has been applied in SIFT matching [8] and face recognition via sparse representation [25], named the importance recognition validation (IRV) strategy in this paper.

The motivation of the IRV strategy is that, the assignment probability between a target and its true identity should be much higher than any incorrect identity. Let $z^* \in Z$ be the optimal assignment of the $i_{th}$ target, and $\hat{z}$ be the suboptimal assignment. Then we can compute the acceptance possibility of the optimal assignment by:

$$a_i = \max \left\{ 1, \frac{P(z_i = z^* | x_{i1;j2}^{(0)})}{\alpha P(z_i = \hat{z} | x_{i1;j2}^{(0)})} \right\}$$  

(15)

where $\alpha > 1$ represents an importance coefficient which is a constant value. If the optimal assignment of the $i_{th}$ target is not accepted, we let

$$p(z_i | x_{i1;j2}^{(0)}) = \begin{cases} 1 & \text{if } z_i = 0, \\ 0 & \text{otherwise.} \end{cases}$$  

(16)

so that the target will not be recognized as any player.

3.4 Discussions on Computational Cost

It costs about 0.013 second to extract the informative features from one target sample. Hence, the proposed system can process 10 frames per second (FPS) when learning the sub-vocabularies or performing identity recognition for 10 players at each time.

In practice, to make sure that the system run at real-time or near real-time, when there are more than 10 targets in the scene, the target will be divided into two groups of equal size, and then the system learns or labels one group of targets at each time step, and process the other group at the next time step.

4. Experiments and Discussions

This section presents the testing results of the proposed algorithms on the broadcast video of one soccer match of the
Spanish Primera League (Season 2010/2011, Barcelona vs. Real Madrid) we purchased. The video (1920×1080 pixels, 25 fps) consists of long shots, middle shots and closed shots, and we perform the experiments on the long-shot clips of the video. The experiments are performed using our non-optimized implementation in C++ on a 2.7GHZ dual-core processor.

### 4.1 Experimental Settings

Before the experiments, we perform multi-player tracking using our tracking system to obtain the testing samples, namely, 1000 image patches for each player drawn from the manually labeled targets after the training process of our recognition model finishes. Note that we only track and label the players of the starting lineup of both teams in the tests. To demonstrate the ability of the proposed method to recognize outliers, we also manually draw 2000 image patches from background and the stadium area as outlier samples.

A C^4 human detector [26] is introduced in our tracking system for automatic human detection. The C^4 detector detects human using the contour cues, and achieves a higher speed than the conventional detectors with similar accuracy. The detector used in our implementation is a generic version that is publicly available, rather than specifically trained for our application.

Table 1 shows the values of some parameters used in the experiments. Note that we set γ = 0.8 so that some marginal parts of the players not detected in edge detection are not missed by the spatial proposal. The other parameters are set experimentally and remain identical in all the experiments.

#### 4.2 Tests of the Incremental Vector Quantization Strategy

Since the identity of each target is determined by the incremental vector quantization strategy as described in Sect. 3.3, we firstly explore the performance of the proposed method under different values of L. To begin with, we partition the image patches of each player into many smaller groups, each of which consists of L image patches from consecutive frames. Then we evaluate the recognition performance of the proposed method by performing the experiments of recognizing each group of image patches.

The results shown in Table 2 indicate that the recognition accuracy increases by the value of L when L is smaller than 9, mainly because that more descriptors can be obtained from longer tracks, which leads to more accurate recognition results. Note that the accuracy stops changing when L is over 9. Considering that higher values of L lead to more expensive time cost as more descriptors need to be extracted while the vector quantization process also takes more time, we set the value of L to be 9 in the following experiments.

#### 4.3 Player Recognition Results

Two tests are performed to verify the advantages of the proposed method: the first test is performed on the player samples to test the ability of our method to differentiate the players which are quite similar in appearance; the second test is to verify the ability of our method to recognize outliers, and the outlier samples and the player samples are taken as positive and negative samples in the second test, respectively. Note that the outlier samples are recognized one-by-one in the test as they are drawn independently, and the player samples are recognized group-by-group as mentioned in Sect. 4.2.

The proposed method is compared with two state-of-the-art online learning methods: the online Boosting method and the Online Random Forests method. For a fair comparison, the training data and the training time used by the two competitors are the same with our method. The implementations of the competitors are both publicly available versions supplied by the authors. We also test our method using both informative and uninformative descriptors in both the learning process and the labeling process, referred to as “our method (unfiltered data)” in Fig. 9, to prove the advantages of using only informative descriptors.

Considering player recognition as a classification problem, we demonstrate the advantages of our method by plotting the ROC curves of different methods. To draw the ROC curves indicating the performance of our method in the first test, we firstly compute the true positive rate and the false positive rate for each player, and the mean values are then used as the results of our method.

Figure 9 (a) and Fig. 9 (b) show the results of the first test and the second one, respectively. Our method performs much better in both tests when only the informative descriptors are used in learning and labeling as shown in the figures. Also, our method performs substantially better than the two competitors, both of which are too adaptive to store history data, as they are designed for training discriminative and adaptive appearance models in tracking applications, rather than object recognition tasks.

| Parameter | Value |
|-----------|-------|
| γ         | 0.80  |
| τ         | 2     |
| ε         | 0.001 |
| t         | 1200  |
| K         | 200   |
| α         | 2     |

| Value | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 |
|-------|---|---|---|---|---|----|----|----|----|
| Accuracy | 0.78 | 0.85 | 0.91 | 0.92 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 |
Fig. 9 ROC curves of different methods. Figure 9 (a) and Fig. 9 (b) show the results of the first test and the second one, respectively. Note that our method performs substantially better than the online Boosting method and the Online Random Forests method in both tests.

4.4 Tracking and Online Labeling Results

Besides the off-line player recognition tests mentioned in Sect. 4.3, we also have our tracking system tested on 28 long-shot video clips consisting of 20311 frames. To demonstrate the performance of both the player tracking system and the player recognition model, we performed two tests on the same tested video clips. In the first test, we performed multi-player tracking using our tracking method without labeling the targets. In the second test, we labeled each new target during multi-player tracking as described in Sect. 3.3.

The online labeling results were evaluated by the metrics described in Table 3, and the results presented in Table 4 shows that, the proposed method performed much better than the other two online learning models. Note that the tracking of each player was interrupted by camera switching for many times; therefore, there are much more tracks (TR) than the number of the tracked players (GT).

Some results of the first test are also presented in Fig. 10(a), and the results of the second test on the same frames are presented in Fig. 10(b), respectively. In Fig. 10(a), the numbers upon the rectangles indicate the indexes of the targets. In Fig. 10(b), the identity of each player is indicated by his team (‘M’ for Madrid, ‘B’ for Barcelona) and player number. More specifically, “M2” represents the player in Madrid whose player number is 2 (Ronaldo), and “B10” represents the player with number 10 in Barcelona (Messi).

The improvement contributed by the player recognition model to the tracking system can be demonstrated by comparing the results shown in the sub-figures of Fig. 10. As indicated in Fig. 10(a), most targets are consistently tracked under common situations; however, if the appearance or view-point of a missed target (target 1) changes abruptly, it may be mistakenly considered as a new target (target 19) when it appears again, rather than recognized by its own Boosting classifiers. Moreover, after a short-term camera switch, most targets are considered as new targets because of abrupt changes in appearance as shown in frame 21699 (left-bottom) of Fig. 10(a), implying that online Boosting classifiers are not suitable for recognition tasks. On the other hand, the recognition model is insensitive to abrupt appearance changes, making the tracking system more robust under these complicated situations. Note that most of the targets can be correctly recognized and labeled under various view points as shown in Fig. 10(b).

4.5 Labeling Tests on the PETS 2009 Dataset

To show the performance of the proposed model on different scenes, we also tested the recognition model on the PETS 2009 dataset†. In the tests, the recognition model was learned on video “View_008” and tested on video “View_005”. Since those two videos were captured on the same scene by cameras of different viewpoints as shown in Fig. 11, the test results can indicate the proposed model’s robustness to camera switching.

Some test results are shown in Fig. 12, where the characters in the middle of each rectangle represent the recognized identities. As shown in the figure, although the viewpoints of “View_005” and “View_008” are very different, most persons in the scene were still correctly recognized and labeled. The persons surrounded by solid rectangles in

### Table 3

| Metrics   | Description                             |
|-----------|-----------------------------------------|
| GT        | The number of players tracked and labeled|
| CS        | The number of camera switches            |
| TR        | The number of tracks                    |
| CL        | The number of tracks that were correctly labeled |
| CLR       | The ratio between CL and TR              |

### Table 4

| Methods            | GT  | CS  | TR  | CL        | CLR     |
|--------------------|-----|-----|-----|-----------|---------|
| Our method         | 20  | 73  | 944 | 812       | 86.02%  |
| Online Boosting    | 20  | 73  | 944 | 330       | 34.96%  |
| Online Random Forests | 20  | 73  | 944 | 377       | 39.94%  |

†Available at http://www.cvg.rdg.ac.uk/PETS2009/data.html.
Fig. 10 Some tracking and labeling results of our method. Note that after a short-term camera switch, most of the targets were mistakenly initialized as new targets rather than recognized by their Boosting classifiers, as shown in frame 21699 (left-bottom) of Fig. 10(a); on the other hand, although observed under different viewpoints, most of the targets were correctly recognized and labeled before and after the camera switch, as shown in Fig. 10(b).
frame #183 were not recognized because of the long distance between them and the camera.

5. Conclusions

In this paper, we have presented a multi-player tracking system and an online learned player recognition model that can be embedded in the tracking system, to achieve the goal of player tracking and recognition in long-shot scenes of broadcast soccer game videos.

The tracking system generates a spatial proposal to minimize the scanning scope of the human detector to realize real-time player detection, and the assignment probability between a target and a detection response is determined by the online Boosting classifiers of the target, making the tracking system robust under complicated situations, such as players similar in appearance interacting with each other.

The recognition model is used to realize player recognition and labeling in an online manner during the tracking process. Extensive tests are performed on a real soccer game video, and both quantitative and qualitative results are reported to verify the advantages of the proposed methods. Note that the recognition model can also be used in other tracking applications for target recognition, if training samples from different viewpoints of the targets are available.

Soccer player tracking and recognition still face many difficulties, e.g., abrupt appearance changes may cause tracking failures, and manual inputs are still necessary during the learning process of the player recognition model. We will work on these problems in the future work.

Acknowledgements

The paper was partly supported by a grant from the National Natural Science Foundation of China (No.60772063, No.61175096) and International S&T Cooperation Program of Beijing Institute of Technology.

References

[1] K. Okuma, A. Taleghani, N. Freitas, J. Little, and D. Lowe, “A boosted particle filter: Multitarget detection and tracking,” Proc. ECCV, pp.28–39, 2004.
[2] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, and H. Wang, “Automatic player detection, labeling and tracking in broadcast soccer video,” Pattern Recognit. Lett., vol.30, no.2, pp.103–113, 2009.
[3] P. Mazzeo, P. Spagnolo, M. Leo, and T. D’Orazio, “Soccer players classification in a multi-camera environment,” Adv. Concepts Int. Vision Syst., pp.143–154, 2010.
[4] N. Kasuya, I. Kitahara, Y. Kameda, and Y. Ohta, “Robust trajectory estimation of soccer players by using two cameras,” Proc. Int. Conf. on Pattern Recognit., pp.1–4, 2008.
[5] J. Sullivan and S. Carlsson, “Tracking and labelling of interacting multiple targets,” Proc. ECCV, pp.619–632, Springer, 2006.
[6] M. Bertini, A. Del Bimbo, and W. Nunziati, “Player identification in soccer videos,” Proc. ACM SIGMM Int. Workshop Multimed. Inf. Retr., pp.25–32, 2005.
[7] W. Lu, J. Ting, K. Murphy, and J. Little, “Identifying players in broadcast sports videos using conditional random fields,” Proc. IEEE Conf. CVPR, pp.3249–3256, 2011.
[8] D. Lowe, “Distinctive image features from scale-invariant keypoints,” INT J COMPUT VISION, vol.60, no.2, pp.91–110, 2004.
[9] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” Comput. Vision–ECCV 2006, pp.404–417, 2006.
[10] Y. Cheng, “Mean shift, mode seeking, and clustering,” IEEE Trans. Pattern Anal. Mach. Intell., vol.17, no.8, pp.790–799, 1995.
[11] W. He, T. Yamashita, H. Lu, and S. Lao, “Surf tracking,” Proc. IEEE 12th ICCV, pp.1586–1592, 2009.
[12] M. Leo, T. D’Orazio, P. Spagnolo, P. Mazzeo, and A. Driste, “Sift based ball recognition in soccer images,” Image Signal Process., pp.263–272, 2008.
[13] H. Grabner and H. Bischof, “On-line boosting and vision,” Proc. IEEE Conf. on CVPR, pp.260–267, 2006.
[14] O. Elgawi, “Online random forests based on corrfs and corrb,” Proc. IEEE Conf. on CVPR Workshops, pp.1–7, 2008.
[15] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, “Online multiperson tracking-by-detection from a single, uncalibrated camera,” IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.9, pp.1820–1833, 2011.
[16] C. Kuo, C. Huang, and R. Nevatia, “Multi-target tracking by on-
XU et al.: ONLINE LEARNED PLAYER RECOGNITION MODEL BASED SOCCER PLAYER TRACKING AND LABELING FOR LONG-SHOT SCENES

line learned discriminative appearance models,” Proc. IEEE Conf. on CVPR, pp.685–692, 2010.

[17] X. Song, J. Cui, H. Zha, and H. Zhao, “Vision-based multiple interacting targets tracking via on-line supervised learning.” Proc. ECCV., pp.1–511, 2001.

[18] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features.” Proc. IEEE Conf. on CVPR, pp.642–655, 2001.

[19] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman, “Discovering object categories in image collections,” Proc. ICCV, 2005.

[20] M. Donoser and H. Bischof, “Efficient maximally stable extremal region (mser) tracking.” Proc. IEEE Conf. CVPR, pp.553–560, 2006.

[21] D. Sculley, “Web-scale k-means clustering.” Proc. 19th Int. Conf. on World Wide Web, pp.1177–1178, ACM, 2010.

[22] S. Stalder, H. Grabner, and L. Van Gool, “Cascaded confidence filtering for improved tracking-by-detection,” Comput. Vision–ECCV 2010, pp.369–382, 2010.

[23] J. Canny, “A computational approach to edge detection,” Read. Comput. Vision: Issues, Probl., Princ., Paradig., vol.184, pp.87–116, 1987.

[24] C. Huang, B. Wu, and R. Nevatia, “Robust object tracking by hierarchical association of detection responses,” Proc. 10th ECCV, pp.788–801, 2008.

[25] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” IEEE Trans. Pattern Anal. Mach. Intell., vol.31, no.2, pp.210–227, 2009.

[26] J. Wu, C. Geyer, and J. Rehg, “Real-time human detection using contour cues,” IEEE Int. Conf. on Robot. Autom., pp.860–867, IEEE, 2011.

Weicun Xu received his B.S. degree in Computer Science from China University of Petroleum, China, in 2007. He is currently a Ph.D. candidate in the School of Computer Science and Technology at Beijing Institute of Technology. His current research interests are in the areas of image processing, visual tracking, signal processing and intelligent computing.

Qingjie Zhao received the B.S. and M.S. degrees in Control Science and Engineering, from China University of Petroleum in 1988, and Shandong University in 1996, respectively. She received the Ph.D. degree in Computer Science and Technology from Tsinghua University, in 2003. She was a visiting fellow in the School of Computer Science and Electronic Engineering, University of Essex, U.K., from Sep. 2008 to Aug. 2009. She is currently an associate professor in the School of Computer Science and Technology, Beijing Institute of Technology, China. Her current research interests include computing intelligence, machine vision, system theory and application.

Yuxia Wang received the B.S degree in School of Mathematics and Information Science, Hebei Normal University, China, in 2006. In 2010, she was studying for her M.S degree in School of Computer Science of Beijing Institute of Technology, China. Currently, she is a Ph.D. student in School of Computer Science of Beijing Institute of Technology, China. Her research interests include computer vision and machine learning, especially in visual object tracking.

Xuanya Li received the Bachelor’s degree in engineering from National University of Defense Technology, Changsha, China, in 2008, the Doctoral degree from Beijing Institute of Technology, Beijing, China, in 2013. He is currently an assistant professor at National Engineering Laboratory for Information Security Technologies, Institute of Information Engineering, in Chinese Academy of Sciences. His main research interests include mobile computing, Internet of Things, and information security.