Collaborate Ball and Player Trajectory Extraction in Broadcast Soccer Video

Yi Zhang, Hanqing Lu
NLPR, Institute of Automation, Chinese Academy of Sciences
{yizhang,luhq}@nlpr.ia.ac.cn

Changsheng Xu
Institute for Infocomm Research, Singapore
xucs@i2r.a-star.edu.sg

Abstract

Enormous accessible broadcast soccer videos demand an efficient ball and player trajectory extraction framework to represent the tactic semantics for the automatic analysis. Camera motions, noise and blurs in broadcast videos make it difficult to extract the trajectories with a single existing object tracking algorithm. In this paper, we propose a novel framework for ball and player trajectory extraction in broadcast soccer videos. The framework generates candidate ball trajectories and player trajectory segments, then it searches the optimal ball trajectory with the likelihood ranking and refines player trajectories with MCMC data association. Instead of extracting ball and player trajectories respectively, our framework employs the motion relationship of the ball and players to build a collaborate scheme to improve the tracking and trajectory refinement results. The experimental results show the proposed framework is more effective than previous works.

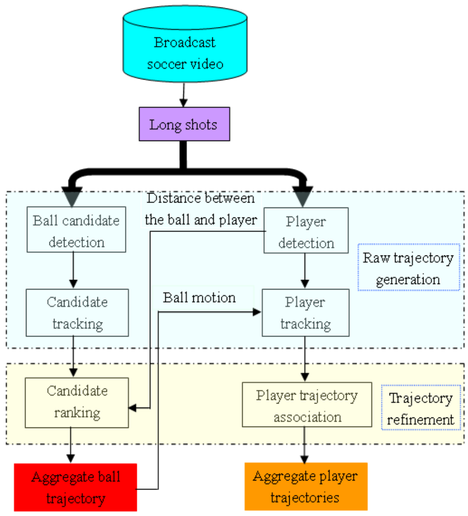

Figure 1. The framework of ball and player trajectory extraction.

1. Introduction

Tremendous broadcast soccer videos on the Internet demand an automatic semantic representation for event and tactic analysis. Trajectories of the ball and players include the temporal and spacial information of the main objects in the game, which makes them the best feature for semantic representation. However, the broadcast video's low resolution, motion blur and noise bring challenges to the trajectory extraction.

Some related works on trajectory extraction have been done in ball games. Scaramuzza[1], Ren[2] and Misu[3], all proposed schemes and frameworks to obtain ball and player trajectories in soccer videos. Unfortunately, these algorithms rely on the camera models which is inaccessible for broadcast video analysis. To deal with the noisy and informal broadcast video, Tong[4] claimed a non-ball elimination algorithm for ball detection in soccer videos. Zhu[5] detected the ball and players with the support vector classifier and tracked them with the support vector regression particle filter. In [6], graph model was employed to select the optimal player and ball trajectory segment and generate components of the aggregate trajectory, but the relationship between the movement of the ball and players were not involved.

In this paper, we propose a novel collaborative ball and player trajectory extraction framework for broadcast soccer videos. Unlike existing works which track the ball and players respectively, the algorithm proposed in this paper integrates them together for a collaborate trajectory extraction. Fig. 1 illustrates the framework of the trajectory extraction in broadcast soccer videos. To represent the raw trajectories, long shots are segmented into temporal intervals. In each of the interval, detection and tracking algorithms are developed to generate the raw ball and player trajectories. A new candidate ball ranking technology is employed to find the optimal ball trajectory. With the temporal and spacial MCMC data association, segments of raw player trajectories are selected and connected to refine the aggregate trajectories. The motion relationship of the ball and players are considered in the ball tracking and player trajectory refinement. The refinement reduces occlusions and false alarms in the trajectory.

The rest of the paper is organized as follows. Ball and player trajectory extractions are discussed in section 2 and 3 respectively. Experiments are shown in section 4 for performance evaluation. Finally, the paper is concluded in section 5.

2. Ball trajectory extraction

2.1. Ball detection and tracking

The ball detection algorithm generates several candidate balls in the frame by removing non-ball objects. The model of non-balls includes features of the size, color, shape and position.

All the candidate balls detected are accepted as the initialization of the tracking algorithm. We employ SVR particle filters[5] with first-order linear motion model to track these candidates. Making the decision on whether the candidate is a ball is delayed until after the tracking.

2.2. Candidate trajectory ranking

After the generation of the candidate ball trajectories, a decision scheme is required to select the optimal ball trajectory among these candidates. We introduce the ball trajectory likelihood to find the best trajectory for the ball. The ranking procedure is performed simultaneously with the tracking. Each time the tracking algorithm extends the trajectories, we update the likelihoods of all candidate trajectories. This can help us stop tracking and discard the candidates with low likelihoods. The tracking ends up with all candidates' likelihood falling below a predefined threshold. We select the trajectory with the highest likelihood as the extracted ball trajectory of the time interval. The Adaboost algorithm is employed to learn the ball trajectory likelihood with the five features of ball position, velocity, acceleration, trajectory length and distance between the ball and the nearest player.

The distance between the ball and the nearest player is a feature of position relationship of the ball and players. The ball tracking results are often affected by surrounding players. False alarms most likely appear beside players as a part of the body. The distance indicates the confidence of the tracking results. Small distances between the ball and player imply low confidence of the tracking results.

The ball trajectory and non-ball trajectory are labeled with 1 and 0 in the training dataset respectively. For each of these features, a weak learner determines a threshold for the classification with the training dataset of 1000 frames. The output of the Adaboost $H(x) = \sum_{i=1}^{5} \alpha_i h_i$ lies in [0, 1], which is defined as the ball trajectory likelihood. We set the accept threshold of the ball trajectory as 0.5 to stop tracking and discard the candidates with low likelihoods.

3. Player trajectory extraction

3.1. Player detection and tracking

For player trajectory extraction, long shots are segmented into equal intervals. In the first frame of the temporal interval, players are detected for tracking initialization. In long shots, fields are extracted with the dominant color and contours of objects in the field are derived with the background substraction. The contours are detected as players according to the scale and shape. In broadcast soccer videos, player contours are merged and connected which significantly affects the precision of the player detection. In order to reduce the merge of players, the dominant player scale is introduced to determine whether the detected contours should be split. The dominant player scale is learnt from the scale histogram of contours in the field. The most frequent scale is selected as the dominant scale of players. We split the contour if its scale is 20% larger than the dominant scale, which can effectively reduce the merge.

For each of the detected players in the first frame of the interval, a particle filter tracker is assigned. The posterior $p(x_t|z_t)$ is presented by random samples of the posterior distribution, where x_t and z_t are the state and observation of the ball at time t. The posterior density is presented by:

$$p(x_t|z_{1:t}) = \sum_{i=1}^{N} w_t^i \delta(x_t - x_t^i) \quad (1)$$

where $\delta()$ is the Dirac delta function. The weights are updated by:

$$w_t = w_{t-1} \frac{p(z_t|x_t)p(x_t|x_{t-1})}{q(x_t|x_{0:t-1}, z_{0:t})} \quad (2)$$

Taking the proposal distribution $q(x_t|x_{0:t-1}, z_{0:t}) = p(x_t|x_{t-1})$, the equation is reduced to $w_t = w_{t-1}p(z_t|x_t)$.

$p(z_t|x_t) = p(z_{1t}|x_t)p(z_{2t}|x_t)$ where $p(z_{1t}|x_t)$ is presented by the Bhattacharyya distance between the histograms of the target and the initial template. $p(z_{2t}|x_t) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{\theta^2}{2\sigma^2})$, where $\theta \in [-\pi, \pi]$ is the angle between the ball's and the player's moving direction. We utilize the consistency of the ball and player's moving direction to model their motion relationship for the collaborate tracking.

$p(x_t|x_{t-1})$ is derived by the first-order linear motion model:

$$x_t = Ax_{t-1} + v \quad (3)$$

where v is the Gaussian white noise.

We resample the state with sampling importance resampling strategy to avoid the effect of degeneracy. After normalizing the weights $[w_t^1, w_t^2, \dots, w_t^N]$, we generate a new distribution of the samples and resample from it.

3.2. Player trajectory association

Occlusions of players and false alarms make one trajectory segment in the interval not correspond to one player. Player trajectories in temporal intervals derived above require data association for the aggregate trajectory extraction. Player trajectory extraction is formulated as a probabilistic problem of multiple hypotheses estimation and temporal trajectory segment structure analysis. Temporal and spacial MCMC is employed for the player trajectory refinement.

We define the trajectories in time intervals $[1, T]$ as $\omega = \{\omega_0, \omega_1, \dots, \omega_k\}$ where ω_0 is the false alarm and ω_k is the kth trajectory. The likelihoods of the trajectories are described with the objects' motion, color, the trajectories' length and overlap. With the motion equation in Equ. 3, the motion likelihood is presented as follows:

$$L_{Motion}(\omega_k(t_{i+1}^f)|\omega_k(t_i^l)) = \frac{1}{2\pi|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2} e_i^T \Sigma^{-1} e_i) \quad (4)$$

where $\omega_k(t_i^f)$ and $\omega_k(t_i^l)$ denote the first and last node in the t_i interval of the kth trajectory. $e_i = \omega_k(t_{i+1}^l) - \bar{\omega}_k(t_{i+1}^l)$, where $\omega_k(t_{i+1}^l)$ and $\bar{\omega}_k(t_{i+1}^l)$ denote the prior and posterior estimates.

The color likelihood of the trajectory is defined as:

$$L_{Color}(\omega_k(t_{i+1}^f)|\omega_k(t_i^l)) = \exp(-\lambda D(\omega_k(t_{i+1}^f), \omega_k(t_i^l))) \quad (5)$$

where $D(\cdot)$ is the Bhattacharyya distance of histograms.

Long trajectories with less overlaps are preferred by our refinement algorithm. We define the trajectory length prior as:

$$p(Length) = \prod_{k=1}^{K} C_1 \exp(-len(\omega_k)) \quad (6)$$

where $len(\cdot)$ is the number of intervals in the trajectory.

The spacial overlap between two trajectories is represented as:

$$p(Overlap) = C_2 \exp(-\frac{\sum_{i,j,i \neq j} \sum_{t=0}^{T} I(\omega_i(t) = \omega_j(t))}{\sum_{k=1}^{K} len(\omega_k)}) \quad (7)$$

where $I(\cdot)$ is the indicate function.

Combining the above priors and likelihoods, we have the whole posterior in Equ. 8.

$$p(\omega|Y) \propto L_{Motion} \cdot L_{Color} \cdot p(Length) \cdot p(Overlap) \quad (8)$$

where Y is the observation.

Metropolis-Hastings sampling is utilized for the MCMC trajectory association[7][8]. In order to accelerate the convergence, we initialize the state of the chain with the results of the deterministic trajectory association[6]. Two spacial and three temporal moves are defined to compute the state change probability.

The spacial moves include segmentation and aggregation.

Segmentation: If more than one trajectory's prediction $w_k(t)$ is overlapped at time interval t, the overlapped part of the trajectory is a candidate for a segmentation move. We randomly select such a candidate and generate the same trajectory segment as the candidate for all trajectories including the candidate. The probability of the move is $q_{seg} = \frac{1}{C_s} p_s$, where C_s is the candidate number (similar as the following C_a, C_m and C_{sw}), and p_s is the prior of the spacial move.

Aggregation: If one trajectory has more than one possible following interval segment at time t, we randomly select a candidate and merge the interval segments to form a new trajectory with the probability $q_{agg} = C_a p_s$.

Three temporal moves are defined as follows.

Merge: If a trajectory's end interval node is close enough to another trajectory's start node, these two trajectories are a pair of candidate for a merge move with the probability $q_{mer} = \frac{1}{C_m} p_t$, where p_t is the prior of temporal move.

Figure 2. Trajectory extraction in the long shot.

Table 1. Results of trajectory extraction

Sample shots	Balls	Players	Correct ball position		Correct Player position		Ball trajectory precision		Player trajectory precision	
			Ours	method in [6]	Ours	method in [6]	Ours	method in [6]	Ours	method in [6]
Shot1	150	1273	129	120	1130	1098	86.0%	80.0%	88.8%	86.3%
Shot2	150	1121	131	122	1110	997	87.3%	81.3%	99.0%	88.9%
Shot3	150	907	139	133	847	833	92.7%	88.7%	93.6%	91.8%
Shot4	150	835	140	135	768	768	93.3%	90.0%	92.0%	92.0%
Shot5	150	1079	135	127	973	955	90.0%	84.7%	90.2%	88.5%
Shot6	150	716	150	144	716	704	100%	96.0%	100%	98.3%
Shot7	150	804	141	137	745	685	94.0%	91.3%	92.7%	85.2%
Total	150	6735	965	918	6289	6040	91.9%	87.4%	93.4%	89.7%

Split: We select a trajectory ω_k uniformly at random and select a break point on the trajectory according to the probability $sp_k(i) = \frac{-\log L_{Motion}(\omega_k(t_{i+1}^f)|\omega_k(t_i^l))}{\sum_{i=1}^{len(\omega_k)} -\log L_{Motion}(\omega_k(t_{i+1}^f)|\omega_k(t_i^l))}$. The nodes after the break point are moved to a new trajectory with the probability $q_{spl} = \frac{1}{K} sp_k(i) p_t$.

Switch: If there exists a common node $\omega_{k_1}(t_i) = \omega_{k_2}(t_i)$ in two trajectories ω_{k_1} and ω_{k_2}, the node is a candidate for a switch move. We randomly select a candidate and switch the nodes after the candidate node with the probability $q_{swi} = \frac{1}{C_{sw}} p_t$.

4. Experimental results

The test has been conducted on the videos of the Soccer World Cup 2006. The videos are compressed in MPEG-4 with the frame rate of 25fps and the resolution of 960×544. Parameters are experimentally set as $\Sigma = I, \sigma = 1, \lambda = 0.1, C_1 = 10^{-5}, C_2 = 1, p_s = p_t = 0.3$. The results of the ball and player trajectory extraction algorithm are listed in Table 1. Seven detected long shots in the video with the total number of 1050 frames are used as the test data of the trajectory extraction. We compare the ball and player trajectories obtained by our framework with the manually labeled groundtruth and accept a ball or player position as a correct result if the distance of the ball position between our algorithm and the groundtruth is no more than 5 pixels. A comparison with the algorithm in [6] is also carried out on the same data. Fig. 2 illustrates the results of a sample shot. The results show our framework is able to effectively extract ball and player trajectories with the collaboration of ball and player motions in long shots of broadcast soccer videos.

5. Conclusions

A new framework for collaborative ball and player trajectory extraction is proposed in broadcast soccer video. Online detection and tracking and offline trajectory refinement schemes are unified to overcome the difficulties brought with the broadcast video. The results show the collaborate tracking, ball candidate trajectory ranking and probability based trajectory association efficiently improve the accuracy of the trajectory extraction.

References

[1] Scaramuzza D. Pagnottelli S. Valigi P. Sports video analysis and structuring. In *International Conference on Robotics and Automation*, pages 1561–1566. IEEE, Apr 2005.

[2] Ren J. Orwell J. Jones G. A. Xu M. A general framework for 3d soccer ball estimation and tracking. In *Proceedings of the 11th International Conference on Image Processing*, pages 24–27. IEEE, Oct 2004.

[3] Misu T. Matsui A. Naemura M. Fujii M. Yagi N. Distributed particle filtering for multiocular soccer ball tracking. In *International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 937–940. IEEE, Apr 2007.

[4] Tong X. Lu H. Liu Q. An effective and fast soccer ball detection and tracking method. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pages 795–798. IEEE, Aug 2004.

[5] Zhu G. Xu C. Huang Q. Wen G. Automatic multi-player detection and tracking in broadcast sports video using support vector machine and particle filter. In *International Conference on Multimedia and Expo*, pages 1629–1632. IEEE, Jul 2006.

[6] Zhu G. Huang Q. Xu C. Rui Y. Jiang S. Wen G. Yao H. Trajectory based event tactics analysis in broadcast sports video. In *ACM Multimedia*, pages 58–67, 2007.

[7] Qian Y. Medioni G. Cohen I. Understanding the metropolis-hastings alogrithm. *American Statistician*, 49:327–335, 1995.

[8] Chib S. Greenberg E. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *Computer Society Conference on Computer Vision and Pattern Recogniton*, pages 18–25. IEEE, Jun 2007.