A Scheme for Ball Detection and Tracking in Broadcast Soccer Video

Dawei Liang[1], Yang Liu[1], Qingming Huang[2], and Wen Gao[1,2]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
{dwliang, yliu}@jdl.ac.cn
[2] Graduate School of Chinese Academy of Sciences, Beijing 100080, China
{qmhuang, wgao}@jdl.ac.cn

Abstract. In this paper we propose a scheme for ball detection and tracking in broadcast soccer video. There are two alternate procedures in the scheme: ball detection and ball tracking. In ball detection procedure, ball candidates are first extracted from several consecutive frames using color, shape, and size cues. Then a weighted graph is constructed, with each node representing a candidate and each edge linking two candidates in adjacent frames. Finally, Viterbi algorithm is employed to extract the optimal path as ball's locations. In ball tracking procedure, Kalman filter based template matching is utilized to track the ball in subsequent frames. Kalman filter and the template are initialized using detection results. In each tracking step, ball location is verified to update the template and to guide possible ball re-detection. Experimental results demonstrate that the proposed scheme is promising.

1 Introduction

In the past decade, sports video analysis from the standpoint of computer vision has attracted much attention, especially in ball games such as soccer [1-7], American football [8], tennis [9], snooker [10] etc. Through detection and/or tracking of the moving objects (players, ball), several high level analysis can be done, e.g. highlight extraction [2], event detection [10] and tactic analysis [6]. This paper focuses on ball detection and tracking in broadcast soccer video, which is a challenging task.

There are some literatures stating the problem of soccer ball detection and tracking. Chromatic and morphological features are utilized to detect ball in [1]. In [2] the authors use template matching to detect ball in difference image after camera motion compensation at regular intervals, and then ball tracking is carried out between such intervals. In [3] ball's location is initialized manually, after that Kalman filter and template matching are applied to track it. In [4] the authors employ motion information in ball detection and tracking, but in their case the cameras are fixed. A modified version of the directional Circle Hough Transform is used to detect ball in real (not broadcast) image sequences in [5]. In [6], ball candidates are first obtained in each video frame based on playfield detection. Afterwards, Kalman filter is employed to generate candidate trajectories from which ball trajectories are selected and extended. In [7], the authors exploit a coarse-to-fine strategy to identify ball in a single frame after playfield detection, and then CONDENSATION algorithm is used to track the ball.

Y.-S. Ho and H.J. Kim (Eds.): PCM 2005, Part I, LNCS 3767, pp. 864 – 875, 2005.

Although the above-mentioned methods provide certain solutions to the ball detection and tracking issue, the problem is not fully resolved yet. The challenges associated with ball detection and tracking can be attributed to the following factors:

- The ball's attributes (color, shape, size and velocity etc.) change over frames;
- The ball becomes a long blurred strip when it moves fast;
- The ball is sometimes only a little white blob;
- The ball is sometimes occluded by players, merged with lines, or hidden in the auditorium;
- Many other objects are similar in appearance to the ball, such as some regions of the players, some white line segments in the playfield and so on;
- Playfield appearance varies from place to place and from time to time, and cannot be modeled appropriately;
- etc.

Some typical ball samples in broadcast soccer video are shown in Fig.1. These ball samples further testify the challenges of ball detection and tracking in broadcast soccer video.

Fig. 1. Some typical ball samples in broadcast soccer video

In this paper, we propose a scheme for ball detection and tracking in broadcast soccer video. It can work well in various playfield conditions. The flowchart is shown in Fig.2. It is composed of two alternate procedures: ball detection and ball tracking. In ball detection procedure, color, shape and size are used to extract ball candidates in each frame. Then a weighted graph is constructed, with each node representing a candidate and each edge linking two candidates in adjacent frames. Finally, Viterbi algorithm is applied to extract the optimal path which is most likely to be ball's path as ball's locations. The method can enhance the robustness of ball detection, since it holds multiple hypotheses of ball's locations. Once the ball is detected, the tracking procedure based on Kalman filter and template matching is started. Kalman filter and the template are initialized using detection results. In each tracking step, ball location is verified to update the template and to guide possible ball re-detection.

The remainder of the paper is organized as follows. In section 2, the method for ball detection is discussed in detail. In section 3, Kalman filter based ball tracking is introduced. Some experimental results are provided in section 4 and the conclusions are presented in the last section.

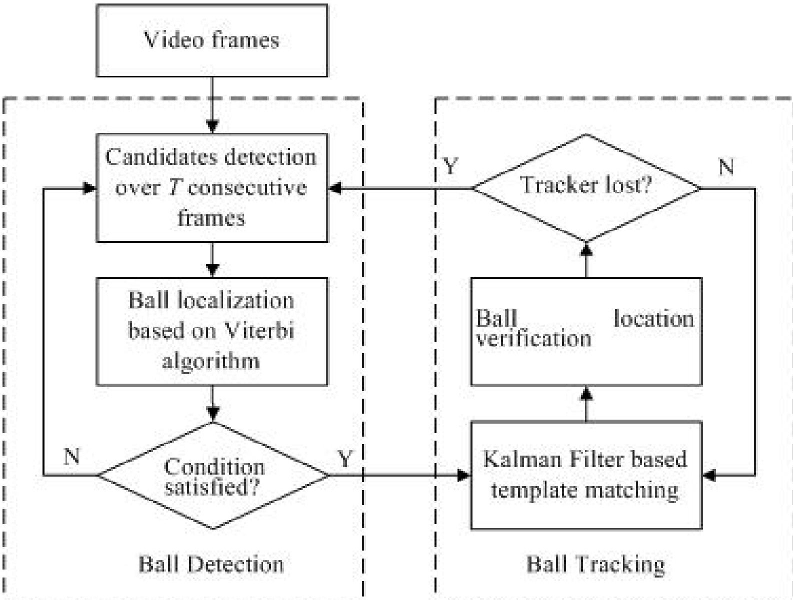


Fig. 2. The flowchart of the proposed scheme

2 Ball Detection

The basic idea of ball detection is to use graph to hold multiple hypotheses of the ball's locations. We extract the optimal path of the graph as the ball's locations rather than identify whether a single object is a ball or not in a single frame, since there are many objects similar to the ball.

2.1 Ball Candidates Detection

Based on the observation that the ball's color is nearly white in long view shots, white pixels are first segmented according to (1) in normalized RGB color space.

$$B(x,y)=\begin{cases}1 & (r(x,y)-1/3)^2+(b(x,y)-1/3)^2\le a^2 \ \wedge\ I(x,y)\ge b\\ 0 & otherwise\end{cases} \tag{1}$$

where B is a binary image and (x, y) is the pixel location, $r(x, y)$ and $b(x, y)$ denote normalized red and blue component, respectively, $I(x, y)$ denotes luminance value. The thresholds are set to $a = 0.05$ and $b = 160$ empirically. Morphological close operation is used to eliminate noises, after that a new region growing algorithm [11] is employed to connect pixels into regions and smooth the boundaries. To obtain ball candidates, several features are used, including the size of the object, the ratio of length and width of the object's minimal bounding rectangle (MBR), the area ratio of the object and its MBR. In order to adapt to various ball appearances, the thresholds are set as loosely as possible to ensure that the true ball region is included in. In our experiment, the threshold of the first feature is set differently as the object appears at different image position, the threshold of the second one is set to 1.5, and the threshold of the third one is set to 0.5.

2.2 Graph Construction

After candidates detection over T consecutive frames, a weighted graph is constructed. Each graph node represents a ball candidate. Since the ball's locations in two adjacent frames are close to each other, only those candidate pairs (between adjacent frames) whose Euclidean distance (in image plane) is smaller than the threshold $d_{\max}$ contribute to the graph edge set. According to formula (2) each node is assigned a weight representing how it resembles a ball. Meanwhile each edge is assigned a weight through formula (4) to represent how likely the two nodes correspond to the same object.

$$v_i^t = \begin{cases} 1-\sqrt{c_i^t} & c_i^t \le 1 \\ 0 & c_i^t > 1 \end{cases} \tag{2}$$

where

$$c_i^t = \frac{1}{M\mu_r^2}\sum_k (\|p_k - \mu\| - \mu_r)^2 \tag{3}$$

$$e_{i,j}^t = (\varpi_s s_{i,j}^t + \varpi_g g_{i,j}^t) / \sqrt{1+(d_{i,j}^t / d_{\max})^2} \tag{4}$$

In above formulae, superscript t denotes the relative serial number of frame; subscripts i and j denote the ith candidate in frame t and the jth candidate in frame $t+1$, respectively. In (3), c_i^t is called Circular Variance (CV) [12]. The less CV is, the more the contour resembles a circle. p_k is a contour point, M is the number of contour point, μ is the centroid of the contour, and μ_r is the average distance from contour points to the centroid. In (4) $s_{i,j}^t$ and $g_{i,j}^t$ are the size and the gray level similarity of two candidates respectively with ϖ_s and ϖ_g as the corresponding weights. For simplicity we set $\varpi_s = \varpi_g = 0.5$. $d_{i,j}^t$ is the Euclidean distance (in image plane) of two candidates.

We assume $(\Delta w, \Delta h)$ obeys Gaussian distribution, where Δw is the width difference between MBRs of two candidates, and Δh is the height difference between MBRs of two candidates. Therefore, $s_{i,j}^t$ can be defined as (5), where Σ can be estimated from ball samples. In (6) $g_{i,j}^t$ is the gray level normalized cross correlation of two candidate regions, where vectors $\vec{I}_1$ and $\vec{I}_2$ are obtained through raster scanning of the candidate regions. If the candidate regions are not equal in size, they are adjusted to equal size before scanning.

$$S_{i,j}^{t} = N(0,\Sigma) \tag{5}$$

$$g_{i,j}^{t} = \frac{\sum_k \vec{I}_1(k)\cdot\vec{I}_2(k)}{\sqrt{\sum_k \vec{I}_1(k)\cdot\vec{I}_1(k)}\sqrt{\sum_k \vec{I}_2(k)\cdot\vec{I}_2(k)}} \tag{6}$$

2.3 Ball's Path Extraction

Finding the optimal path of a graph is a typical dynamic programming problem. Viterbi algorithm is employed to extract it based on the constructed graph. Note that the graph can be constructed incrementally. Viterbi algorithm is described in Fig.3 similar to that in [13].

1. Initialization: $\delta_i^1 = v_i^1$, $\psi_1(i) = 0$, $1 \le i \le N_1$;
2. Recursion: $\delta_j^t = \max_{1\le i\le N_{t-1},\, d_{i,j}^{t-1}\le d_{\max}} (\delta_i^{t-1} + e_{i,j}^{t-1} + v_j^t)$,

   $\psi_t(j) = \arg\max_{1\le i\le N_{t-1},\, d_{i,j}^{t-1}\le d_{\max}} (\delta_i^{t-1} + e_{i,j}^{t-1} + v_j^t)$,

   $1 \le j \le N_t,\ 2 \le t \le T$;
3. Termination: $q_T = \arg\max_{1\le i\le N_T} (\delta_i^T)$;
4. Path backtracking: $q_t = \psi_{t+1}(q_{t+1})$, $t = T-1, T-2, \cdots, 1$

Fig. 3. Ball localization based on Viterbi algorithm

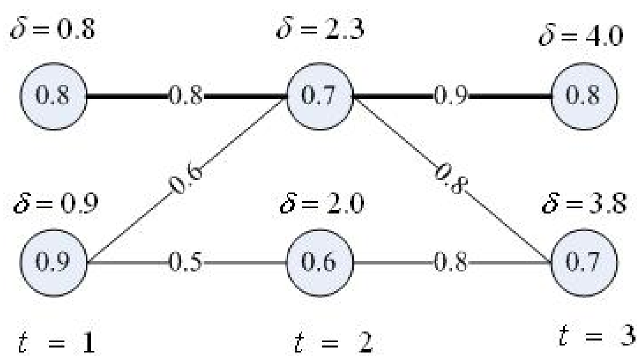


Fig. 4. Illustration of the weighted graph. The optimal path is marked with bold line

Let P_j^t be the optimal path ending at jth candidate in frame t, then the notations in Fig.3 are explained as follows. N_t is the number of candidates in frame t, δ_j^t is the sum of node and edge weights along P_j^t, $\Psi_t(j)$ is the index linking to the candidate in frame $t-1$ on P_j^t, and $\{q_t\}_{t=1,\cdots,T}$ is the optimal path. If the number of candidates on the optimal path is less than T, the observation window is moved forward by one frame and then the ball detection procedure is run again. An illustration of the graph and its optimal path is shown in Fig.4.

3 Ball Tracking

In ball tracking procedure, Kalman filter based template matching (in terms of gray level normalized cross correlation) is exploited. Kalman filter predicts the ball's location in the next frame and filters the tracking result in the current frame. Template matching is used to obtain observation. Kalman filter and the template are initialized using detection results.

Kalman filter addresses the general problem of estimating the state X of a discrete time process that is governed by the linear stochastic difference equation

$$X_{k+1} = AX_k + w_k \tag{7}$$

with a measurement Z that is

$$Z_k = HX_k + v_k \tag{8}$$

The random variables w_k and v_k represent the process and measurement noise respectively. They are assumed to be independent of each other and have normal distribution. In this paper first order dynamics model is employed, i.e.

$$X = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}, Z = \begin{bmatrix} x \\ y \end{bmatrix}, A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{9}$$

where (x, y) denotes the ball's center, and $(\dot{x}, \dot{y})$ denotes the ball's velocity. Due to lack of space, please refer to [14] for more details about Kalman filter.

A simple but effective method is adopted to make the tracker adaptable to the ball's scale change over frames. A slightly larger block $(x1-\Delta, y1-\Delta, x2+\Delta, y2+\Delta)$ is generated for matched ball region $(x1, y1, x2, y2)$, where $(x1, y1)$ and $(x2, y2)$ are the top-left coordinates and the bottom-right coordinates of the matching region, respectively. The same method in section 2.1 is used to extract object and formula (2) is used to evaluate whether it is a ball. If ball is detected, the template is updated. The number of consecutive missing detections is counted, and if it is larger than a predefined threshold (say 5), the ball detection procedure is run again.

4 Experiments

The proposed scheme is tested on several video clips containing video frames varying from 200 to 1000, with each being MPEG-2 compressed with the resolution of 352×288 at 25 frames per second recorded from TV. Our test video clips are all long

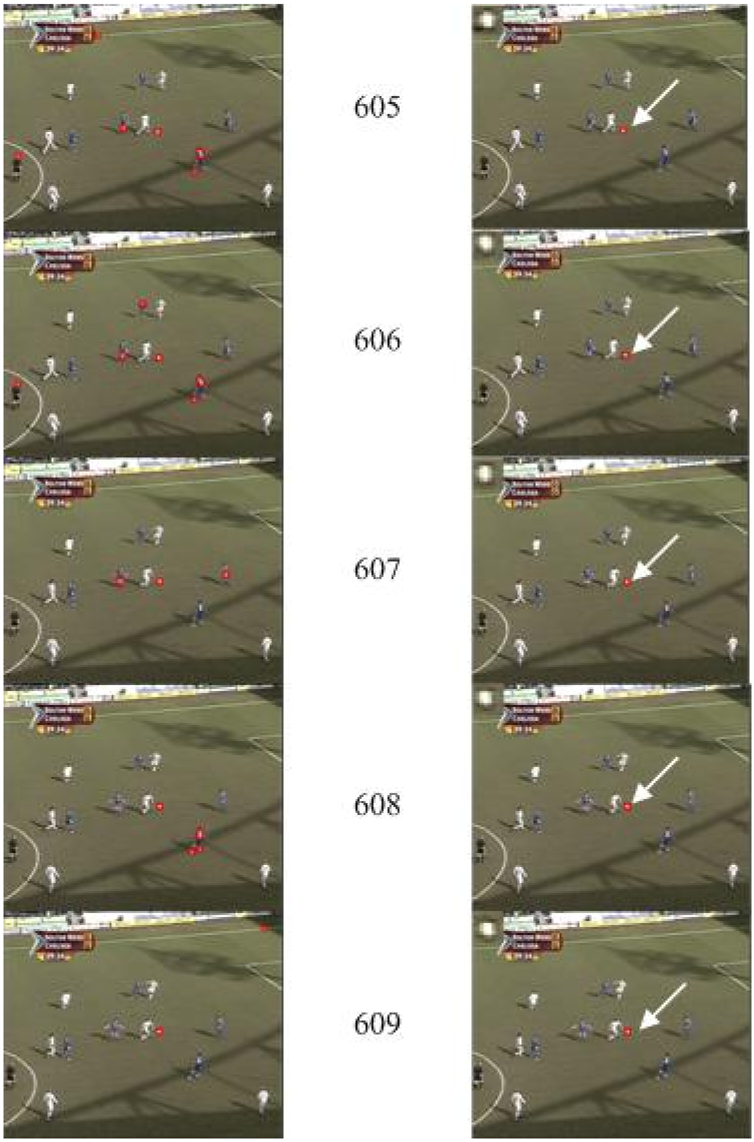
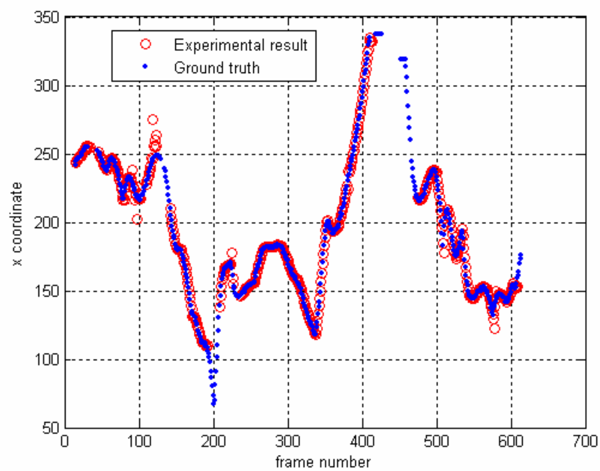


Fig. 5. The ball detection results from frame 605 to 609. In the left column, ball candidates are marked with red rectangle. Extracted ball locations using Viterbi algorithm are shown in the right column with a white arrow indicating the ball's location in each image. The ball is also enlarged and shown on the top left corner of each image to make it seen clearly. The frame number is shown in the middle column.

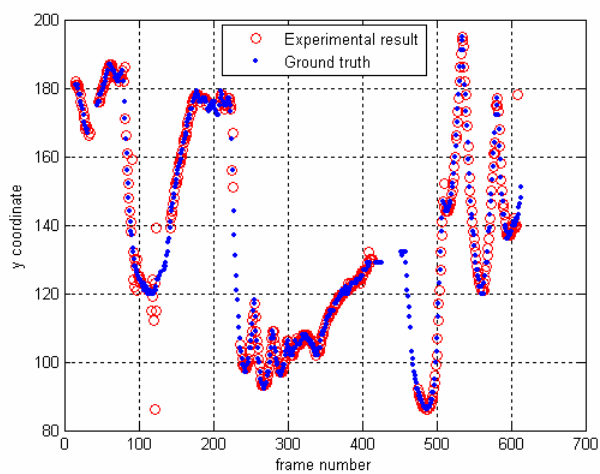
view shots, since tracking the soccer ball in close-up shots seems to be meaningless for soccer video analysis. In all of the experiments we set $T = 5, d_{max} = 20$.

4.1 Ball Detection

Fig. 5 shows some experimental results of the ball detection procedure. In the left column, ball candidates are marked with red rectangle. In the right column, extracted ball



(a)



(b)

Fig. 6. Ball detection and tracking results on a video sequence with 650 frames. Experimental results are shown in red circle. The ground truths are marked with blue dot. (a) The x-coordinate (in image plane). (b) The y-coordinate (in image plane).

locations based on Viterbi algorithm are also marked with red rectangle with a white arrow indicating the ball's location in each image. The ball is also enlarged and shown on the top left corner of each image to make it seen clearly. Frame number is shown in the middle column. From the results, we can see that the objects that are most likely to be a ball are socks of the player, jersey number, line segments in the playfield and so on. Although so many objects are similar in appearance to the ball, the ball's locations are correctly obtained based on Viterbi algorithm using temporal information.

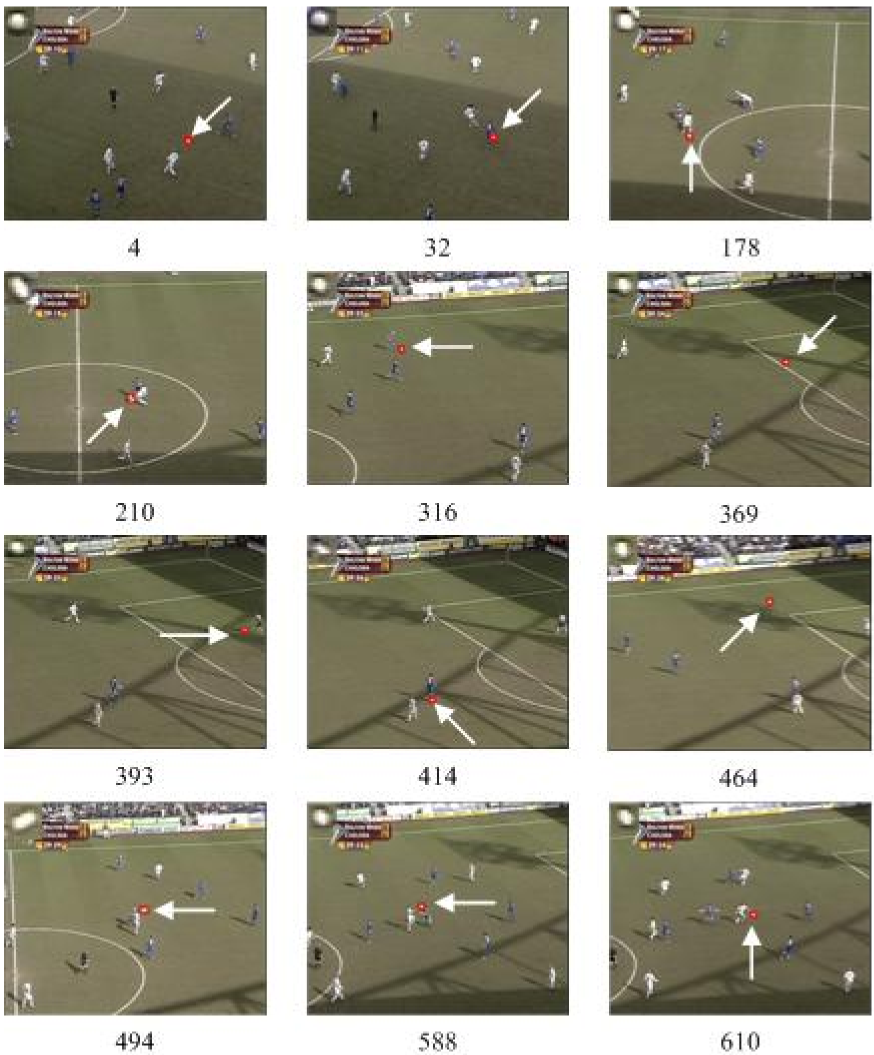


Fig. 7. Some video frames of ball detection and tracking results on a video clip with 650 frames (414 is a false positive)

4.2 Evaluation

Fig. 6 shows ball detection and tracking results on a video clip with 650 frames. The ball's true locations are marked manually on ball's centers. We say a frame contains a ball, if we can find it not depending on the adjacent frames. The experimental result can also be reached at http://www.jdl.ac.cn/user/dwliang/index.htm in video format. Fig. 7 provides some video frames in the process of soccer ball detection and tracking.

To better evaluate the scheme's performance, we did some statistics on two video sequences with each having more than 600 frames. Table 1 provides some statistics of the experimental results. *Precision* and *Recall* rate are used to quantitatively evaluate the experiments. In some clips, such as "sequence1-101-200", the recall rate is very low. This is mainly because that the ball is occluded by players, merged with lines and so on, which makes ball detection and tracking failure. In some clips, such as "sequnce1-401-500", the precision rate is very low. This is mainly because the ball is occluded, and the socks of the player or some line segments are most likely to be the ball. Since the assumption of our scheme is that the ball is nearly circular, our scheme can not deal well with the long blurred case. Nevertheless, from the table, we can observe that the overall precision and recall rate are acceptable.

Table 1. Some statistics of the experiments

Video sequences	Frame number	Ground truth	Experimental results			
		#ball	#detected & tracked	#false positive	Precision (%)	Recall (%)
Sequence1 (bad playfield)	1-100	88	67	2	97.0	73.9
	101-200	92	53	0	100	57.6
	201-300	99	97	10	89.7	87.9
	301-400	97	97	3	96.9	96.9
	401-500	80	62	20	67.7	52.5
	501-600	96	92	5	94.6	90.6
	601-650	48	45	13	71.1	66.7
	total	600	513	53	89.7	76.7
Sequence2 (good playfield)	1-100	92	93	8	91.4	92.4
	101-200	100	99	3	97.0	96
	201-300	94	95	11	88.4	89.4
	301-400	81	71	23	67.6	59.3
	401-500	64	39	3	92.3	56.2
	501-600	82	84	10	88.1	90.2
	601-700	99	99	4	96.0	96.0
	701-719	19	18	3	83.3	78.9
	total	631	598	65	89.1	84.5

5 Conclusions and Future Work

In this paper we propose a scheme for ball detection and tracking in broadcast soccer video. Viterbi algorithm is used to extract the optimal path of a graph as the ball's locations to enhance the robustness of ball detection, and then a Kalman filter based tracker is initialized using the detection results. Since only a small portion of each frame is processed, the tracking procedure runs very fast. A simple but effective method is also employed to verify the ball's locations during tracking and to guide possible ball re-detection. Experimental results show that the scheme can work well even in bad playfield conditions. In future work, we will use the ball's trajectory with other information for high level semantics analysis.

Acknowledgements

This work is partly supported by NEC Research China on "Context-based Multimedia Analysis and Retrieval Program" and "Science 100 Plan" of Chinese Academy of Sciences.

References

1. Gong Y., Sin L. T., Chuan C. H., Zhang H. J., and Sakauchi M.: Automatic Parsing of TV Soccer Programs. In: Proceedings of Second International Conference on Multimedia Computing and Systems (1995) 167-174.
2. Yow D., Yeo B., Yeung M., and Liu B.: Analysis and Presentation of Soccer Highlights from Digital Video. In: Proceedings of Second Asian Conference on Computer Vision (1995) 499-503.
3. Seo Y., Choi S., Kim H., and Hong K.: Where are the ball and players? Soccer Game Analysis with Color-based Tracking and Image Mosaic. In: Proceedings of 9th International Conference on Image Analysis and Processing, vol.2 (1997) 196-203.
4. Ohno Y., Miura J. and Shirai Y.: Tracking Players and Estimation of the 3D Position of a Ball in Soccer Games. In: Proceedings of 15th International Conference on Pattern Recognition, vol.1, (2000) 145-148.
5. D'Orazio T., Ancona N., Cicirelli G., and Nitti M.: A ball detection algorithm for real soccer image sequences. In: Proceedings of International Conference on Pattern Recognition, vol.1 (2002) 210-213.
6. Yu X., Xu C., Leong H. W., Tian Q., Tang Q., and Wan K. W.: Trajectory-Based Ball Detection and Tracking with Applications to Semantic Analysis of Broadcast Soccer Video. In: Proceedings of ACM Conference on Multimedia, (2003) 11-20.
7. Tong X., Lu H., and Liu Q.: An Effective and Fast Soccer Ball Detection and Tracking Method. In: Proceedings of International Conference on Pattern Recognition, vol. 4, (2004) 795-798.
8. Intille S.S., Bobick A.F.: Closed-World Tracking. In: Proceedings of International Conference on Computer Vision (1995) 672-678.
9. Pingali G.S., Opalach A., and Jean Y.: Ball Tracking and Virtual Replays for Innovative Tennis Broadcasts. In: Proceedings of International Conference on Pattern Recognition, Vol. 4, (2000) 152-156.

10. Denman H., Rea N., and Kokaram A.: Content-based analysis for video from snooker broadcasts. Journal of Computer Vision and Image Understanding (CVIU) 92 (2003) 176-195.
11. Ye Q., Gao W., and Zeng W.: Color Image Segmentation using Density-based Clustering. In: Proceedings of International Conference on Acoustics, Speech, and Signal Processing, Vol.3, (2003) 345-348.
12. Peura M. and Iivarinen J.: Efficiency of Simple Shape Descriptors. In: Arcelli C. et. al. (Eds.) Advances in Visual Form Analysis, World Scientific, Singapore (1997) 443-451.
13. Rabiner L.R.: A Tutorial on Hidden Markov Model and Selected Applications in Speech Recognition. In: Proceedings of the IEEE, Vol. 77, No. 2, (1989).
14. Welch G. and Bishop G.: An Introduction to the Kalman Filter. Technical Report TR95-041. Department of Computer Science, University of North Carolina at Chapel Hill. (1995) http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html