A Low Cost Algorithm for Expected Goal Events Detection in Broadcast Soccer Video
Masoomeh Zameni, Mahmood Fathy, Amin Sadri

Department of Computer Engineering, Iran University of Science and Technology,
zameni_m@comp.iust.ac.ir, mahfathy@just.ac.ir, a_sadri@elec.iust.ac.ir


Abstract

Recently, mining information in sports video data, especially soccer video, has become an active research topic. In this paper, a new algorithm for detection of expected goal events in soccer video is proposed. The proposed algorithm is composed of two main steps. Firstly, video is segmented into its constituent shots and these shots are categorized into two groups, namely long shots and non-long shots. Secondly, long shots are examined to detect expected goal events. Our scheme uses the playfield boundary feature for shot boundary detection, shot classification, and expected goal events detection. Therefore, this feature is extracted only one time and the algorithm is speedy. It is noticeable that the proposed algorithm is robust to blurring and spatial downsampling. Moreover, experimental results on various broadcast soccer videos show that our algorithm can achieve high accuracy and especially high speed.

Keywords: Shot, Shot Boundary, Shot Classification, Goal Line, Occlusion

1. Introduction

Soccer is one of the most appealing sports in the world. Considering the limitation of bandwidth to distribute soccer video over network and not having enough time to watch all details of the match, the automatic recognition of important events has been the focus of a number of research efforts in recent years. Generally, soccer video analysis methods can be divided into two main groups: 1) methods use cinematic (low-level) features and 2) methods detect events with the aid of object-based features.

Cinematic features refer to those that result from common video composition and production rules, such as shot types and replays. Objects are described by their spatial, e.g., color, texture, and shape, and spatiotemporal features, such as object motions and interactions. Object-based features enable high-level domain analysis, but their extraction may be computationally costly for real-time implementation. Cinematic features, on the other hand, offer a good tradeoff between the computational requirements and the resulting semantics [1].

Many algorithms have been developed [2, 3] based on low-level features. On the other hand, algorithms in [4, 5, 6, 7] have used object-based features. In [8] a method is proposed for highlight extraction in soccer videos which uses both object-based and low-level features. It is based on the observations that the appearance of goal-mouth points to a high likelihood of exciting action in soccer videos and that highlight is composed of certain types of scene views which exhibit certain transition rules. Li and Sezan summarize football video by play/break and slow-motion replay detection using both cinematic and object descriptors [9]. Scene cuts and camera motion parameters are used for soccer event detection in [10]. A mixture of cinematic and object descriptors is employed in [11]. Text information from closed captions and visual features are integrated in [12] for event-based football video indexing. Seo et al. [13] proposed an algorithm that does ball tracking in broadcast soccer video (BSV). They use a Kalman filter-based template matching procedure to track the ball and use backprojection to predict possible occlusion. [14] presents a novel trajectory-based detection and tracking algorithm for locating the ball in BSV. First a set of ball-candidates for each frame is generated, and then they are used to compute the set of ball trajectories. A replay based approach, assuming that the replayed actions are usually highlights, is proposed in [15]. Occurrence of a goal is generally followed by a special pattern of cinematic features, which is what is exploited in goal detection algorithm
proposed in [1]. A goal event leads to a break in the game. During this break, the producers convey the emotions on the field to the TV audience and show one or more replay(s) for a better visual experience. A semantic video indexing algorithm based on finite-state machines and low-level motion indices extracted from the MPEG compressed bit-stream is proposed in [16]. Gong et al. [17] exploited the play field and player position information for the purpose of highlight extraction. In [18] the knowledge of the soccer domain is encoded into a set of finite state machines, each of which models a specific highlight. Highlight detection exploits visual cues that are estimated from the video stream, and particularly, ball motion, the currently framed playfield zone, players’ positions and colors of players’ uniforms. The highlight models are checked against the current observations.

In spite of many previous works, the problem of important events detection is still challenging. We propose a method which employs both object-based and low-level features and its speed is still very fast. Furthermore, unlike many previous algorithms, it is robust to blurring and spatial downsampling. These blurred images caused fast camera motions aroused from strong kick, shoot, etc. Methods based on goal-mouth detection, central circle detection and field-line detection; mostly, have lack of robustness in blurred images. Playfield boundary is a feature which is used frequently in this algorithm. It is extracted accurately; hence, the accuracy of whole algorithm is increased.

The rest of the paper is organized as follows. Section 2 describes the main flowchart of our algorithm. Sections 3 and 4 present the proposed methods for detection of shot boundaries and classifying shots, respectively. Our system for expected goal event detection is introduced in section 5, with experimental results presented in section 6. We conclude this paper in section 7.

2. Our Approach

The overall event detection scheme can be best described by the diagram in Figure 1. Because exciting areas near the goal are important to the users, we first determine frames near to the goal area. In order to increase the speed and the accuracy of high-level analyses, we focus only on long view shots and detect players. Then using some simple domain rules, we check if these frames can be related to a goal event or not. These rules are obtained by observing many soccer videos.

![Diagram](image)

**Figure 1. Framework of the scheme**

3. Shot Boundary Detection

Sports video is arguably one of the most challenging domains for robust shot boundary detection [1]. In this section, a domain knowledge based algorithm for detection of shot
boundaries in soccer videos is proposed. Generally, we have noticed that in each transition the number of top non-grass pixels differs. Following steps describe our shot boundary detection algorithm.

3.1. Playfield Detection

First step in our shot boundary detection algorithm is segmentation of playfield. For this purpose, the method in [19] is used in which the authors assigned $N_H$ bins for Hue channel, $N_S$ for Saturation channel, and $N_V$ for Value channel. In this work, we set $N_H = 64$, $N_S = 64$, $N_V = 256$ experimentally. Furthermore, $\theta_1$, $\theta_2$, and $\theta_3$ are set to be all zero experimentally. This simple histogram based classification is very fast. Since the ground color may vary, we run the playfield detector on every 100 frames.

3.2. Playfield Boundary Detection

After segmenting the playfield, the next step is to extract boundary of playfield. To detect the playfield boundary, we exploit top non-grass pixels. The boundary consists of a sequence of the number of top non-grass pixels in each column of image. To speed up the execution of the algorithm, we use a 20 pixels $\times$ 20 pixels block and count the top non-grass pixels in each column (the width of each column is considered to be 20 pixels empirically). In each block, the percentage of non-grass pixels is calculated. If the percentage is greater than an empirically determined threshold (the threshold is set to .8 experimentally), then the block is marked as a non-green block, else it is marked as a green block. In each column of the image, we find the first green block and record its location. The first green block in each column lies on the boundary of playfield. At this time, to accurately approximate the boundary, vertical displacement of the block is changed to 1 pixel and the threshold becomes larger (.9 in our implementation), and we continue until reaching a green block. Then, we examine the next column (Figure 2).

![Figure 2. Counting top non-grass pixels. (a) Binary image, (b) Non-green blocks are shown on binary image.](image)

3.3. Shot Boundary Detector

Final step in our shot boundary detection algorithm is detection of shot boundaries. This method uses a new feature, the sum of the square differences between top non-grass pixels sequences of two successive frames, denoted by $G_d$. Computation of $G_d$ between $i$th and $(i-k)$th frames is given by (1), where $g^m_i$ represents the $m$th value of top non-grass pixels in the sorted sequence of $i$th frame (We have determined $k$ to be 10 for speed purposes).

$$G_d[i,k] = \sum_{m=1}^{floor(width/masklength)} (g^m_i - g^m_{i-k})^2$$

(1)
If the $G_j$ is larger than a predefined threshold, a shot boundary is detected. Our reason for sorting top non-grass pixels sequence is to enhance the robustness against small motions of objects and camera.

4. Shot Classification

After determining shot boundaries, we classify each shot in next step. We have presented a new view classification scheme based on playfield boundary of soccer video. This scheme classifies the shots into two types: the long shot, and non-long shot. The long shot frames are images captured in a long distance, thus most parts of the shot tend to be occupied by ground. For the other type of frames, the ground color would have little impact on the viewing experience since the viewers’ eye gazes are usually drawn by interesting objects [19]. Due to the computational simplicity of our algorithm, we select five frames of the shot, find the class of five frames and assign the shot class to the label of the majority of frames. Assuming $T$ is the length of shot, we select frames of $1/6$ T, $2/6$ T, $3/6$ T, $4/6$ T and $5/6$ T locations.

In Figure 3 different shapes of playfield boundary in long view frames which are near goal area are shown. These shapes are obtained by observing many soccer videos. As it can be seen, all these patterns have linear playfield boundaries.

![Figure 3. Different shapes of playfield boundary in long view frames which are near goal area (green areas represent the playfield)](image)

According to our observations on soccer videos, a linear playfield boundary corresponds to a long shot frame (Figure 4(a)), while a non-linear boundary indicates that the frame is a non-long shot frame (Figure 4(b)).

![Figure 4. Frame view types. (a) A long shot frame, (b) A non-long shot frame (boundaries are determined in white)](image)

Furthermore, there are some non-long views in which playfield boundary sequence have mostly either zero values or high values.

In order to find the frame view, we first inspect the playfield boundary sequence. If most of these values are zero or high, we classify the frame as a non-long view. Otherwise, we check
linearity of the playfield boundary. If the boundary is linear, the frame is assigned to a long view category; else, it is classified as a non-long view frame. To check the linearity of boundary, we assume there are two intersecting lines in playfield boundary and find the intersection point of these lines. Steps for finding the intersection point are:

1) At first, we create a new sequence (B). Each element of the sequence is the difference of corresponding and subsequent elements of playfield boundary sequence A. The values of Sequence B indicate to approximate slopes of line segments of playfield boundary line (equation 2).

\[ B(i) = A(i) - A(i+1) \]  

(2)

2) The intersection point corresponds to an element of sequence B which the difference of average values of its both side values is maximum (equations 3, 4).

\[ C(i) = \frac{\sum_{j=1}^{i-1} B(j)}{i} - \frac{\sum_{j=i+1}^{N} B(j)}{N-i} \]  

(3)

Where \( N \) is the number of elements of sequence A.

\[ \text{intersection point} = \arg \max_j \{ C(j) \} \]  

(4)

After finding the intersection point, the variance of values in both sides of intersection point in sequence A is calculated. If variances are quite small, the playfield boundary is linear and the frame is classified as a long view; otherwise, it is classified as a non-long view. It should be noticed here that, in case of only one line or no line in boundary, both variances are small and this method works properly.

5. Detection of Expected Goal Events

The purpose of this section is detecting expected goal events considering the results of two previous sections. Since most of important events occur near goal area, we first find near goal areas in long shots. There are also some non-expected goal events that occur near goal area. For ignoring these events, we use position of players as an additional feature.

5.1. Near Goal Area Detection

Most of important events occur near goal area; hence, detection of near goal frames is very important. To detect goal areas, we exploit the fact that in long views, the line which is near goal area appears as a gradient-line with a reasonable gradient value. Therefore, for each frame we must first find the intersection point of boundary lines using the method proposed in section 4. Then, the slope of each boundary line is obtained. If there is a line which its slope is reasonable (the image width is 4 times longer than the goal line), we consider the frame as a near goal area image.

5.2. Detection of Players

We have used information about the number and position of players in near goal area images as a cue for expected goal event detection. In our proposed technique, the players are detected by their jersey colors. To increase the speed and the accuracy of algorithm, we examine only the
regions limited to the playing field. Starting at the leftmost column, the examining process is carried out vertically by a block of predefined size (15 pixels × 6 pixels). If the number of jersey color pixels in a block exceeds a determined threshold, this block is considered as a player candidate. The size of block and the value of threshold are obtained experimentally.

In case there is no color pixel of both teams in a block, the block is labeled with 0, otherwise if the number of the team A jersey color pixels is more than the team B, label is 1, and else the block is labeled with 2. The reason for using different labels is to distinguish players’ identities (Team A, Team B). This method improves detection of occluded players.

Finally, we find connected components of these blocks using BFS\(^1\) algorithm and considering 8-neighborhood. Each connected component is considered as a player.

5.3. Important Event Detection

We have created some simple rules to decide whether or not the events are expected goal events. These rules are:
1) If the image width is 4 times longer than the goal line, the image does not belong to an expected goal event, otherwise go to next step.
2) If there is no offensive player in the image, the image is not considered as an important image, otherwise go to next step.
3) If offensive player is more near than defensive players to goal line, the image is important, otherwise go to next step.
4) If the number of offensive players is more than or equal to the number of defensive players, the image belongs to an expected goal event, otherwise go to next step.
5) If the number of defensive players is two times the number of offensive players, the image is not considered important; otherwise the image belongs to an expected goal event.

For each near goal area image we investigate importance of it. After that, we consider a sequence of 70 frames. If most of these frames are important, this sequence is considered as an expected goal event segment.

6. Experimental Results

For evaluation of the proposed algorithm, we have used a dataset of about 2 hours of soccer video. The dataset is composed of 4 AVI clips, 2 of which (Uro 2008: Chelsea vs. Liverpool (half 1 & 2)) in 624×352 resolution at 30 fps and the two others (UEFA Cup Final 2008: Zenit St. Petersburg vs. Rangers (half 2) and UEFA. Champions. League. 2009: Manchester. United. vs. Arsenal (half 1)) in 640×352 resolution at 25 fps.

In Table 1, the recall and the precision rates of the shot boundary detector are given for each match and for the whole set. On the average, the algorithm works at 81% recall and 78% precision rates.

<table>
<thead>
<tr>
<th>Match</th>
<th>Chelsea vs. Liverpool (half 1)</th>
<th>Chelsea vs. Liverpool (half 2)</th>
<th>Zenit St. Petersburg vs. Rangers</th>
<th>Manchester United vs. Arsenal</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>00:30:00</td>
<td>00:30:00</td>
<td>00:30:00</td>
<td>00:30:00</td>
<td>00:120:00</td>
</tr>
<tr>
<td>Correct</td>
<td>120</td>
<td>114</td>
<td>105</td>
<td>112</td>
<td>451</td>
</tr>
<tr>
<td>False</td>
<td>28</td>
<td>29</td>
<td>35</td>
<td>32</td>
<td>124</td>
</tr>
<tr>
<td>Miss</td>
<td>30</td>
<td>25</td>
<td>27</td>
<td>22</td>
<td>104</td>
</tr>
<tr>
<td>Precision</td>
<td>81%</td>
<td>80%</td>
<td>75%</td>
<td>78%</td>
<td>78%</td>
</tr>
<tr>
<td>Recall</td>
<td>80%</td>
<td>82%</td>
<td>80%</td>
<td>84%</td>
<td>81%</td>
</tr>
</tbody>
</table>

The results of our shot classification method are shown in Table 2.

\(^1\) Breath First Search
A Low Cost Algorithm for Expected Goal Events Detection in Broadcast Soccer Video
Masoomeh Zameni, Mahmood Fathy, Amin Sadri

Table 2. Shot classification results

<table>
<thead>
<tr>
<th>Match</th>
<th>Overall</th>
<th>Manchester.United. vs. Arsenal</th>
<th>Zenit St.Petersburg vs. Rangers</th>
<th>Chelsea vs. Liverpool(half1)</th>
<th>Chelsea vs. Liverpool(half2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>00:12:00</td>
<td>00:30:00</td>
<td>00:30:00</td>
<td>00:30:00</td>
<td>00:30:00</td>
</tr>
<tr>
<td>Correct</td>
<td>131</td>
<td>125</td>
<td>136</td>
<td>124</td>
<td>136</td>
</tr>
<tr>
<td>False</td>
<td>20</td>
<td>15</td>
<td>16</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>Accuracy</td>
<td>87%</td>
<td>89%</td>
<td>89%</td>
<td>91%</td>
<td>89%</td>
</tr>
</tbody>
</table>

Because of the fact that expected goal events are detected at the following steps, the errors of shot boundary and shot classification algorithms have a little effect on final result. Table 3 shows the detection rate of players for 200 long view frames which are chosen randomly. The detection rate represents the percentage of correctly detected players among all existing players in all frames. Miss detection was due to the mistake in extra-field region exclusion, and oversight was due to the blur caused by fast camera motion.

Table 3. Player detection results

<table>
<thead>
<tr>
<th>Correct</th>
<th>False</th>
<th>Miss</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1200</td>
<td>163</td>
<td>195</td>
<td>88%</td>
<td>86%</td>
</tr>
</tbody>
</table>

The results of our expected goal event detection algorithm are listed in Table 4. The results are promising.

Table 4. Expected goal event detection results

<table>
<thead>
<tr>
<th>Match</th>
<th>Overall</th>
<th>Manchester.United. vs. Arsenal</th>
<th>Zenit St.Petersburg vs. Rangers</th>
<th>Chelsea vs. Liverpool(half1)</th>
<th>Chelsea vs. Liverpool(half2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>00:12:00</td>
<td>00:30:00</td>
<td>00:30:00</td>
<td>00:30:00</td>
<td>00:30:00</td>
</tr>
<tr>
<td>Correct</td>
<td>33</td>
<td>34</td>
<td>29</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>False</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Miss</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>Precision</td>
<td>80%</td>
<td>83%</td>
<td>81%</td>
<td>86%</td>
<td>83%</td>
</tr>
<tr>
<td>Recall</td>
<td>97%</td>
<td>94%</td>
<td>91%</td>
<td>92%</td>
<td>94%</td>
</tr>
</tbody>
</table>

The processing time per frame for different parts of our algorithm is given in Table 5. Temporal sampling also is applied without performance degradation. As it can be seen, the algorithm has low computational cost.

Table 5. The time cost of the proposed algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Frame Size (pixels)</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Playfield Detection</td>
<td>352*240</td>
<td>2.586</td>
</tr>
<tr>
<td>Shot Boundary Detection</td>
<td>352*240</td>
<td>0.324</td>
</tr>
<tr>
<td>View Classification</td>
<td>352*240</td>
<td>0.214</td>
</tr>
<tr>
<td>Player Detection</td>
<td>352*240</td>
<td>1.523</td>
</tr>
<tr>
<td>Checking The Defined Rules</td>
<td>352*240</td>
<td>0.89</td>
</tr>
</tbody>
</table>

7. Conclusion

In this paper, we have presented a novel algorithm for automatic detection of expected goal events in soccer video. This algorithm is robust to blurring and spatial downampling. First,
video is segmented into its shots and then expected goal events are detected in long shots. Usually important events occur near goal area. To detect goal areas, we exploited the fact that in long views, the line which is near goal area appears as a gradient-line with a reasonable gradient value. We also used information about the number and position of players in images which are near goal area as a cue for expected goal event detection. Finally, we created some heuristic rules to decide whether or not the events are expected goal events. Our future work will be to develop more comprehensive rules to detect expected goal events and also cover tracking ball in near goal areas.

8. References