Automatic Soccer Video Analysis and Summarization

Ahmet Ekin and A. Murat Tekalp[†]

Department of Electrical and Computer Engineering
University of Rochester, Rochester, NY 14627
[†](also with) College of Engineering, Koc University, Istanbul, Turkey
{ekin,tekalp}@ece.rochester.edu
www.ece.rochester.edu/∼ {ekin,tekalp}

ABSTRACT

We propose a fully automatic and computationally efficient framework for analysis and summarization of soccer videos using cinematic and object-based features. The proposed framework includes some novel low-level soccer video processing algorithms, such as *dominant color region detection, robust shot boundary detection,* and *shot classification*, as well as some higher-level algorithms for *goal detection, referee detection*, and *penalty-box detection.* The system can output three types of summaries: i) all slow-motion segments in a game, ii) all goals in a game, and iii) slow-motion segments classified according to object-based features. The first two types of summaries are based on cinematic features only for speedy processing, while the summaries of the last type contain higher-level semantics. The proposed framework is efficient, effective, and robust for soccer video processing. It is *efficient* in the sense that there is no need to compute object-based features when cinematic features are sufficient for the detection of certain events, e.g. goals in soccer. It is *effective* in the sense that the framework can also employ *object-based features* when needed to increase accuracy (at the expense of more computation). The efficiency, effectiveness, and the robustness of the proposed framework are demonstrated over a large data set, consisting of more than 13 hours of soccer video, captured at different countries and conditions.

Keywords: Soccer video processing, cinematic features, object-based features, shot classification, slow-motion replay detection, semantic event detection, soccer video summarization.

1. INTRODUCTION

Sports video distribution over various networks should contribute to quick adoption and widespread usage of multimedia services worldwide, because sports video appeals to large audiences. The valuable semantics in a sports video generally occupy only a small portion of the whole content, and the value of sports video drops significantly after a relatively short period of time.[1] Therefore, sports video processing needs to be completed *automatically,* due to, otherwise, its intimidating size, in *real,* or *near real-time,* and the processing results must be *semantically meaningful.* In this paper, we propose a novel soccer video processing framework that satisfies these requirements.

Semantic analysis of sports video generally involves use of *cinematic* and *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, and spatio-temporal features, e.g. object motions and interactions.[2] 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 trade-off* between the computational requirements and the resulting semantics.

In the literature, object color and texture features are employed to generate highlights[3] and to parse TV soccer programs.[4] Object motion trajectories and interactions are used for football play classification[5] and for soccer event detection.[6] Both[5] and[6] however, rely on pre-extracted accurate object trajectories, which is done manually in[5]; hence, they are not practical for real-time applications. LucentVision[7] and ESPN K-Zone[8] track only specific objects for tennis and baseball, respectively. Cinematic descriptors are also commonly employed. The plays and breaks in soccer games are detected by frame view types in.[9] Li and Sezan summarize football video by play/break and slow-motion replay detection using both cinematic and object descriptors.[10] Scene cuts and camera motion parameters are used for soccer event detection in[11] where usage of very few cinematic features prevents reliable detection of multiple events. A mixture of cinematic and object descriptors is employed in[12] and.[13] Motion activity features are proposed

for golf event detection.[14] *Text* information from closed captions and visual features are integrated in[15] for event-based football video indexing. *Audio* features, alone, are proposed to detect hits and generate baseball highlights.[16] In this paper, we propose a new framework for *automatic*, *real-time* soccer video analysis and summarization by systematically using cinematic and object features. A flowchart of the proposed framework is shown in Fig. 1. The main contributions are:

- We propose new dominant color region and shot boundary detection algorithms that are *robust to variations in the dominant color*. The color of the grass field may vary from stadium to stadium, and also as a function of the time of the day in the same stadium. Such variations are automatically captured at the initial non-supervised training stage of our proposed dominant color region detection algorithm. Variations during the game, due to shadows and/or lighting conditions, are also compensated by automatic adaptation to local statistics.
- We propose two novel features for shot classification in soccer video for *robustness to variations in cinematic features,* which is due to slightly different cinematic styles used by different production crews. The proposed algorithm provides as high as 17.5% improvement over an existing algorithm as shown in Sec. 4.
- We introduce new algorithms for automatic detection of i) goal events, ii) referee, and iii) penalty box in soccer videos. Goals are detected based solely on cinematic features resulting from common rules employed by the producers after goal events to provide a better visual experience for TV audiences. Distinguishing jersey color of the referee is used for fast and robust referee detection. Penalty box detection is based on the *three-parallel-line* rule that uniquely specifies the penalty box area in a soccer field.
- Finally, we propose an efficient and effective framework for soccer video analysis and summarization that combines these algorithms in a scalable fashion. It is *efficient* in the sense that there is no need to compute object-based features when cinematic features are sufficient for the detection of certain events, e.g. goals in soccer. It is *effective* in the sense that the framework can utilize *object-based features* when needed to increase accuracy (at the expense of more computation). Hence, the proposed framework is adaptive to the requirements of the desired processing.

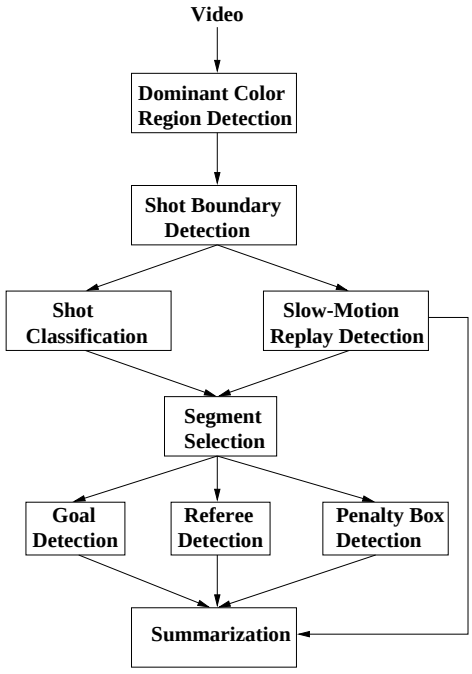


Figure 1. The flowchart of the system

We describe the proposed low-level algorithms for dominant color region detection, shot boundary detection, and shot classification in the next section. Sec. 3 presents proposed higher-level methods for goal detection, referee

detection, and penalty box detection. Experimental results over more than 13 hours of soccer video from different regions of the world and the temporal performance of the system are discussed in Sec. 4.

2. LOW-LEVEL ANALYSIS FOR CINEMATIC FEATURE EXTRACTION

This section explains extraction of low-level cinematic features, such as shot boundary detection and shot classification. Since both the shot boundary detector and shot classifier rely on accurate detection of soccer field region in each frame, we start by presenting our robust dominant color region detection algorithm. We use a slightly modified version of the algorithm in[17] for slow-motion replay detection; hence, we skip its discussion in this paper.

2.1. Robust Dominant Color Region Detection

A soccer field has *one distinct dominant color* (a tone of green) that may vary from stadium to stadium, and also due to weather and lighting conditions within the same stadium. Therefore, the algorithm does not assume any specific value for the dominant color of the field, but learns the statistics of this dominant color at start-up, and automatically updates it to adapt to temporal variations. The dominant field color is described by the mean value of each color component, which are computed around their respective histogram peaks. The computation involves determination of the peak index, i_{peak}, for each histogram, which may be obtained from one or more frames. Then, an interval, $[i_{min}, i_{max}]$, about each peak is defined, where i_{min} and i_{max} refer to the minimum and maximum of the interval, respectively, that satisfy the conditions in Eqs. 1-3, where H refers to color histogram. The conditions define the minimum (maximum) index as the smallest (largest) index to the left (right) of, including, the peak that has a predefined number of pixels. In our implementation, we fixed this minimum number as 20% of the peak count, i.e., $K = 0.2$. Finally, the mean color in the detected interval is computed for each color component.

$$H[i_{min}] \geq K * H[i_{peak}] \quad \text{and} \quad H[i_{min} - 1] < K * H[i_{peak}] \tag{1}$$

$$H[i_{max}] \geq K * H[i_{peak}] \quad \text{and} \quad H[i_{max} + 1] < K * H[i_{peak}] \tag{2}$$

$$i_{min} \leq i_{peak} \quad \text{and} \quad i_{max} \geq i_{peak} \tag{3}$$

Field colored pixels in each frame are detected by finding the distance of each pixel to the mean color by the *robust* cylindrical metric.[18] Since the algorithm works in the HSI space, achromaticity must be handled with care. If the estimated saturation and intensity means fall in the achromatic region, only intensity distance in Eq. 4 is computed for *achromatic* pixels. Otherwise, both Eq. 4 and Eq. 5 are employed for *chromatic* pixels in each frame.

$$d_{intensity}(j) = |I_j - I_{mean}| \tag{4}$$

$$d_{chromaticity}(j) = \sqrt{(S_j)^2 + (S_{mean})^2 - 2S_jS_{mean}cos(\theta)} \tag{5}$$

$$d_{cylindrical}(j) = \sqrt{(d_{intensity})^2 + (d_{chromaticity})^2} \tag{6}$$

$$\theta = \begin{cases} |Hue_{mean} - Hue_j| & \text{if} \quad |Hue_{mean} - Hue_j| < 180^o \\ 360^o - |Hue_{mean} - Hue_j| & \text{if} \quad |Hue_{mean} - Hue_j| > 180^o \end{cases} \tag{7}$$

In the equations, Hue, S, and I refer to hue, saturation and intensity, respectively, j is the j^{th} pixel, and θ is defined in Eq. 7. The field region is defined as those pixels having $d_{cylindrical} < T_{color}$, where T_{color} is a pre-defined threshold value that is determined by the algorithm given the rough percentage of dominant colored pixels in the training segment. The adaptation to the temporal variations is achieved by collecting color statistics of each pixel that has $d_{cylindrical}$ smaller than $a * T_{color}$, where $a > 1.0$. That means, in addition to the field pixels, the close non-field pixels are included to the field histogram computation. When the system needs an update, the collected statistics are used to estimate the new mean color value is computed for each color component.

2.2. Shot Boundary Detection

Shot boundary detection is usually the first step in generic video processing. Although it has a long research history, it is not a completely solved problem.[19] Sports video is arguably one of the most challenging domains for robust shot boundary detection due to following observations: 1) There is strong color correlation between sports video shots that usually does not occur in generic video. The reason for this is the possible existence of a single dominant color background, such as the soccer field, in successive shots. Hence, a shot change may not result in a significant difference in the frame histograms. 2) Sports video is characterized by large camera and object motions. Thus, shot boundary detectors that use change detection statistics are not suitable. 3) A sports video contains both cuts and gradual transitions, such as wipes and dissolves. Therefore, reliable detection of all types of shot boundaries is essential.

In the proposed algorithm, we take the first observation into account by introducing a new feature, *the absolute difference of the ratio of dominant (grass) colored pixels to total number of pixels between two frames* denoted by G_d. Computation of G_d between the i^{th} and $(i-k)^{th}$ frames is given by Eq. 8, where G_i represents the grass colored pixel ratio in the i^{th} frame. As the second feature, we use *the difference in color histogram similarity,* H_d, which is computed by Eq. 9. The similarity between two histograms is measured by histogram intersection in Eq. 10, where the similarity between the i^{th} and the $(i-k)^{th}$ frames, $HI(i,k)$, is computed. In the same equation, N denotes the number of color components, and is three in our case, B_m is the number of bins in the histogram of the m^{th} color component, and H_i^m is the *normalized* histogram of the i^{th} frame for the same color component. The algorithm uses different k values in Eqs. 8-10 to detect cuts and gradual transitions. Since cuts are instant transitions, $k = 1$ will detect cuts, and other values will indicate gradual transitions.

$$G_d(i,k) = |G_i - G_{i-k}| \tag{8}$$

$$H_d(i,k) = |HI(i,k) - HI(i-k,k)| \tag{9}$$

$$HI(i,k) = \frac{1}{N}\sum_{m=1}^{N}\sum_{j=0}^{B_m-1} min(H_i^m[j], H_{i-k}^m[j]) \tag{10}$$

A shot boundary is determined by comparing H_d and G_d with a set of thresholds. A novel feature of the proposed method, in addition to **the introduction of G_d as a new feature,** is **the adaptive change of the thresholds on H_d.** When a sports video shot corresponds to out-of-field or close-up views (the definitions of both will be given in Sec. 2.3), the number of field colored pixels will be very low and the shot properties will be similar to a generic video shot. In such cases, the problem is the same as generic shot boundary detection; hence, we use only H_d with a high threshold. In the situations where the field is visible, we use both H_d and G_d, but using a lower threshold for H_d. Thus, we define four thresholds for shot boundary detection: $T_H^{Low}, T_H^{High}, T_G$, and $T_{lowGrass}$. The first two thresholds are the low and high thresholds for H_d, and T_G is the threshold for G_d. The last threshold is essentially a rough estimate for low grass ratio, and determines when the conditions change from field view to non-field view. The values for these thresholds is set for each sport type after a non-supervised learning stage. Once the thresholds are set, the algorithm needs only to compute local statistics and runs in *real-time.* Furthermore, the proposed algorithm is robust to spatial downsampling since both G_d and H_d are size-invariant. In Sec. 4, we will present our results on 4x4 spatially downsampled video.

2.3. Shot Classification

The type of a shot conveys interesting semantic cues; hence, we classify soccer shots into three classes: 1) Long shots, 2) In-field medium shots, and 3) Out-of-field or close-up shots. The definitions and characteristics of each class are given below[20]:

- **Long shot:** A long shot displays the global view of the field as shown in Fig 2 (a) and (b); hence, a long shot serves for accurate localization of the events on the field.

- **In-field medium shot:** A medium shot, where a whole human body is usually visible, is a zoomed-in view of a specific part of the field as in Fig 2 (c) and (d).
- **Close-up or Out-of-field Shot :** A close-up shot usually shows above-waist view of one person (Fig 2 (e)). The audience, coach, and other shots are denoted as out-of-field shots (Fig 2 (f)). We analyze both out of field and close-up shots in the same category due to their similar semantic meaning.

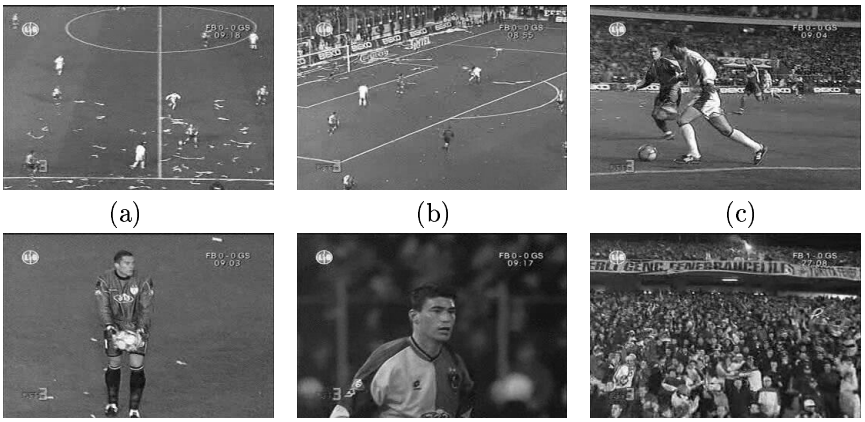


Figure 2. View types in soccer: (a,b) Long view, (c,d) in-field medium view, (e) close-up view, and (f) out of field view

Classification of a shot into one of the above three classes is based on spatial features. Therefore, shot class can be determined from a single key frame or from a set of frames selected according to a certain criteria. In order to find the frame view, frame grass colored pixel ratio, G, is computed. In,[9] an intuitive approach is used, where a low G value in a frame corresponds to a non-field view, while high G value indicates a long view, and in between, a medium view is selected. Although the accuracy of that approach is sufficient for play-break application in,[9] it is not sufficient for our application that extracts higher level semantics. By using only grass colored pixel ratio, medium shots with high G value will be mislabeled as long shots. The error rate due to this approach depends on the broadcasting style and it usually reaches intolerable levels for the employment of higher level algorithms in Sec. 3. Therefore, another feature is necessary for accurate classification of the frames with a high number of grass colored pixels.

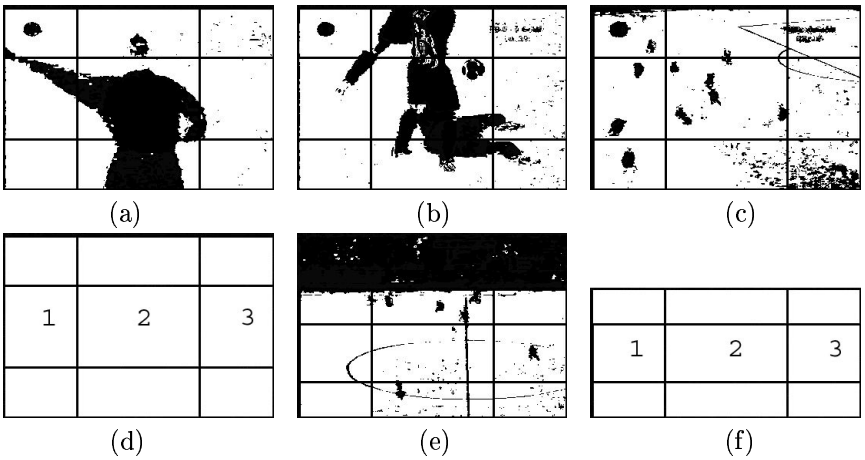


Figure 3. Examples of Golden Section spatial composition in (a-b) medium and (c-e) long views, the resulting grass region boxes and the regions are shown in (d) and (f) for (a-c) and (e), respectively

We propose a compute-easy, yet efficient, cinematographic measure for the frames with high G values. We define regions by using *Golden Section* spatial composition rule,[21,22] which suggests dividing up the screen in 3:5:3 proportion in both directions, and positioning the main subjects on the intersection of these lines. We have revised this rule for soccer video, and divide the *grass region box* instead of the whole frame. *Grass region box* can be defined as the minimum bounding rectangle (MBR), or a scaled version of it, of grass colored pixels. In Fig. 3, the examples of the regions obtained by *Golden Section* rule is displayed on several medium and long views. In the regions R_1, R_2, and R_3 in Fig. 3 (d) and (f), we found the two features below the most distinguishing: G_{R_2}, the grass colored pixel ratio in the second region, and R_{diff}, the average of the sum of the absolute grass color pixel differences between R_1 and R_2, and between R_2 and R_3, found by $R_{diff} = \frac{1}{2}\{|G_{R_1} - G_{R_2}| + |G_{R_2} - G_{R_3}|\}$. Then, we employ a Bayesian classifier using the above two features.

The flowchart of the proposed shot classification algorithm is shown in Fig. 4. The first stage uses G value and two thresholds, $T_{closeUp}$ and T_{medium}, to determine the frame view label. These two thresholds are roughly initialized to 0.1 and 0.4 at the start of the system, and as the system collects more data, they are updated to the minimum of the grass colored pixel ratio, G, histogram as suggested in.[9] When $G > T_{medium}$, the algorithm determines the frame view by using our novel cinematographic features.

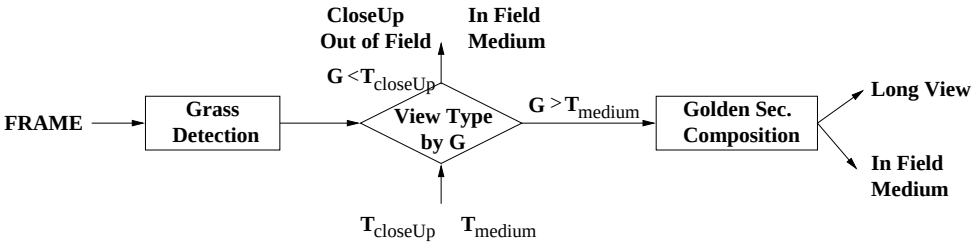


Figure 4. The flowchart of the shot type (view) classification algorithm

3. SOCCER EVENT AND OBJECT DETECTION

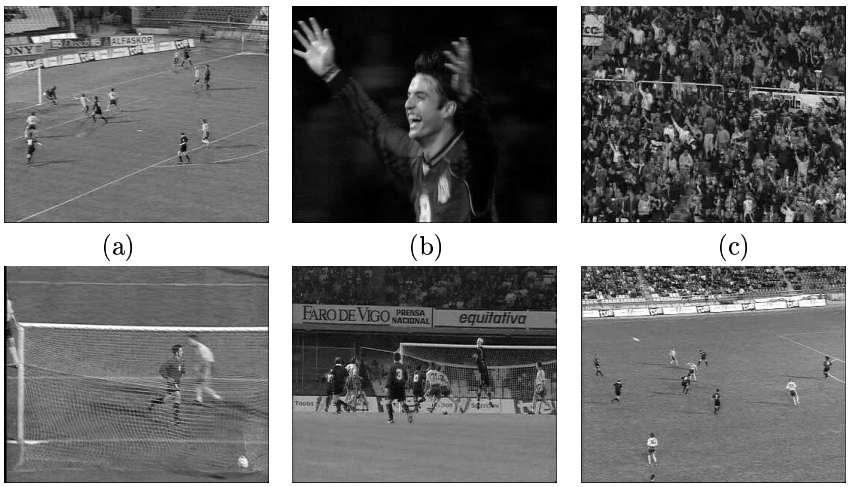
Detection of certain events and objects in a soccer game enables generation of more concise and semantically rich summaries. Since goals are arguably the most significant event in soccer, we propose a novel goal detection algorithm in Sec. 3.1. The proposed goal detector employs *only cinematic features* and runs in *real-time*. Goals, however, are not the only interesting events in a soccer game. Controversial decisions, such as red-yellow cards and penalties (medium and close-up shots involving referees), and plays inside the penalty box, such as shots and saves, are also important for summarization and browsing. Therefore, we also develop novel algorithms for referee and penalty box detection that are presented in Sec. 3.2 and 3.3, respectively.

3.1. Goal Detection

A goal is scored when the whole of the ball passes over the goal line, between the goal posts and under the crossbar.[23] Unfortunately, it is difficult to verify these conditions automatically and reliably by video processing algorithms. However, occurrence of a goal is generally followed by a special pattern of cinematic features, which is what we exploit in our proposed goal detection algorithm. 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. The emotions are captured by one or more close-up views of the actors of the goal event, such as the scorer and the goalie, and by frames of the audience celebrating the goal. For a better visual experience, several slow-motion replays of the goal event from different camera positions are shown. Then, the restart of the game is usually captured by a long shot. Between the long shot resulting in the goal event and the long shot that shows the restart of the game, we define *a cinematic template* that should satisfy the following requirements:

- *Duration of the break:* A break due to a goal lasts no less than 30 and no more than 120 seconds.
- *The occurrence of* **at least one** *close-up/out of field shot:* This shot may either be a close-up of a player or out of field view of the audience.
- *The existence of* **at least one** *slow-motion replay shot:* The goal play is always replayed one or more times.

- *The relative position of the replay shot:* The replay shot(s) follow the close-up/out of field shot(s).



(a) (b) (c)

(d) (e) (f)

Figure 5. The broadcast of the first goal in *Spain1*: (a) long view of the actual goal play, (b) player close-up, (c) audience, (d) the first replay, (e) the third replay, and (f) long view of the start of the new play

In Fig. 5, the instantiation of the template is demonstrated for the first goal in *Spain1* sequence of MPEG-7 data set, where the break lasts for 54 sec. The search for goal event templates start by detection of the slow-motion replay shots. For every slow-motion replay shot, we find the long shots that define the start and the end of the corresponding break. These long shots must indicate a play that is determined by a simple duration constraint, i.e. long shots of short duration are discarded as breaks. Finally, the conditions of the template are verified to detect goals. The proposed "cinematic template" models goal events very well, and the detection runs in real-time with a very high recall rate.

3.2. Referee Detection

Referees in soccer games wear distinguishable colored uniforms from those of the two teams on the field. Therefore, a variation of our dominant color region detection algorithm in Sec. 2.1 can be used to detect referee regions. We assume that there is, if any, a single referee *in a medium or out-of-field/close-up shot* (we do not search for a referee in a long shot). Then, the horizontal and vertical projections[24] of the feature pixels can be used to accurately locate the referee region. The peak of the horizontal and the vertical projections and the spread around the peaks are used to compute the rectangle parameters surrounding the referee region, hereinafter "MBR_{ref}." MBR_{ref} coordinates are defined to be the first projection coordinates at both sides of the peak index without enough pixels, which is assumed to be 20% of the peak projection. In Fig. 6, an example frame, the referee pixels in that frame, the horizontal and vertical projections of the referee region, and the resulting referee MBR_{ref} are shown.

The decision about the existence of the referee in the current frame is based on the following size-invariant *shape* descriptors:

- *The ratio of the area of the MBR_{ref} to the frame area:* A low value indicates that the current frame does not contain a referee.
- *MBR_{ref} aspect ratio (width/height):* It determines if the MBR_{ref} corresponds to a human region.
- *Feature pixel ratio in the MBR_{ref}:* This feature approximates the compactness of the MBR_{ref}, higher compactness values are favored.
- *The ratio of the number of feature pixels in the MBR_{ref} to that of the outside:* It measures the correctness of the single referee assumption. When this ratio is low, the single referee assumption does not hold, and the frame is discarded.

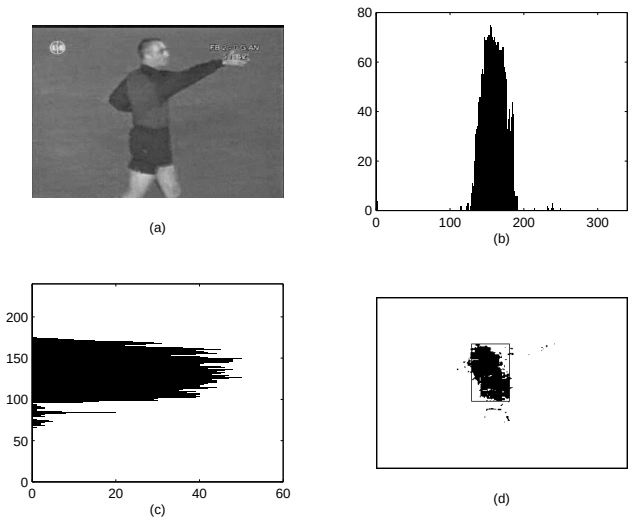


Figure 6. The referee in the input frame (a) is detected by using the horizontal (b) and the vertical (c) projections of the binary referee mask image (d)

The proposed approach for referee detection runs very fast, and it is robust to spatial downsampling. We have obtained comparable results for original (352x240 or 352x288), and for 2x2 and 4x4 spatially downsampled frames.

3.3. Penalty Box Detection

Field lines *in a long view* can be used to localize the view and/or register the current frame on the standard field model. In this section, we reduce the penalty box detection problem to the search for three parallel lines. In Fig. 7 (a), a view of the whole soccer field is shown, and **three parallel field lines,** shown in bold in Fig. 7 (b), become visible when the action occurs around one of the penalty boxes. This observation yields a robust method for penalty box detection, and it is arguably more accurate than the goal post detection proposed in[3] for a similar analysis, since goal post views are likely to include cluttered background pixels that cause problems for Hough transform.

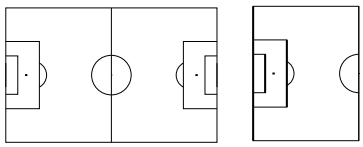


Figure 7. (a) Soccer field model, (b) three highlighted parallel lines around goal area

To detect three lines, we use the grass detection result in Sec.2.1. To limit the operating region to the field pixels, we compute a mask image from the grass colored pixels, displayed in Fig. 8 (b). The mask is obtained by first computing a scaled version of the grass MBR, drawn on the same figure, and then, by including all field regions that have enough pixels inside the computed rectangle. As shown in Fig. 8 (c), non-grass pixels may be due to lines and players in the field. To detect line pixels, we use edge response, defined as the pixel response to the 3x3 Laplacian mask in Eq. 11. The pixels with the highest edge response, the threshold of which is automatically determined from the histogram of the gradient magnitudes, are defined as line pixels. The resulting line pixels after the Laplacian mask operation and the the image after thinning are shown in Fig. 8 (d) and (e), respectively.

$$h = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{vmatrix} \tag{11}$$

Then, three parallel lines are detected by Hough transform that employs *size, distance* and *parallelism* constraints. As shown in Fig. 7 (b), the line in the middle is *the shortest line,* and it has *a shorter distance to the goal line (outer*

line) than to the penalty line (inner line). The detected three lines of the penalty box in Fig. 8 (a) are shown in Fig. 8 (f).

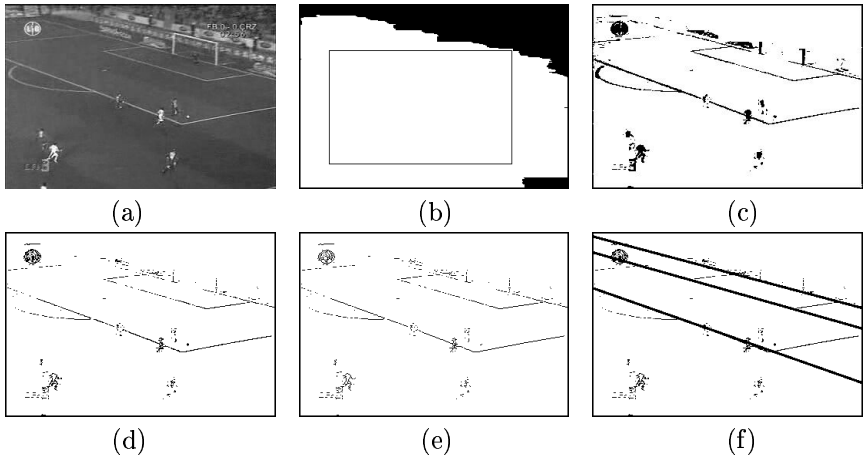


Figure 8. Penalty box detection, (a) the input frame, (b) the field mask, (c) grass/non-grass image in the field region, (d) the pixels in (c) with high gradient, (e) image after thinning, and (f) three detected lines

4. RESULTS

We have rigorously tested the proposed algorithms over a data set of more than 13 hours of soccer video. The database is composed of 17 MPEG-1 clips, 16 of which in 352x240 resolution at 30 fps and one in 352x288 resolution at 25 fps. We have used several short clips from two of the 17 sequences for training. The segments used for training are omitted from the test set; hence, neither sequence is used by the goal detector.

4.1. Results for Low-Level Algorithms

In this section, we present the performance of the proposed low-level algorithms. We define two ground truth sets, one for shot boundary detector and shot classifier, and one for slow-motion replay detector. The first set is obtained from three soccer games captured by *Turkish, Korean,* and *Spanish* crews, and it contains 49 minutes of video. The sequences are not chosen arbitrarily; on the contrary, we intentionally selected the sequences from different countries to demonstrate the robustness of the proposed algorithms to varying cinematic styles.

Each frame in the first set is downsampled, without low-pass filtering, by a rate of four in both directions to satisfy the real-time constraints, that is, 88x60 or 88x72 is the actual frame resolution for shot boundary detector and shot classifier. Overall, the algorithm achieves 97.3% recall and 91.7% precision rates for cut-type boundaries. On the same set at full resolution, a generic cut-detector,[25] which comfortably generates high recall and precision rates (greater than 95%) for non-sports video, has resulted in 75.6% recall and 96.8% precision rates. A generic algorithm, as expected, misses many shot boundaries due to the strong color correlation between sports video shots. The precision rate at the resulting recall value does not have a practical use. The proposed algorithm also reliably detects gradual transitions, which refer to *wipes* for *Turkish, wipes* and *dissolves* for *Spanish,* and *other editing effects* for *Korean* sequences. On the average, the algorithm achieves 85.3% recall and 86.6% precision rates. Gradual transitions are difficult, if not impossible, to detect when they occur between two long shots or between a long and a medium shot with a high grass ratio.

The accuracy of the shot classification algorithm, which uses the same 88x60 or 88x72 frames as shot boundary detector, is shown in Table 1. For each sequence, we provide two results, one by using only grass colored pixel ratio, G, and the other by using both G and the proposed features, G_{R_2} and R_{diff}. Our results for *Korean* and *Spanish* sequences by only G are very close to the reported results on the same set in.[9] By introducing two new features, G_{R_2} and R_{diff}, we are able to obtain 17.5%, 6.3%, and 13.8% improvement in *Turkish, Korean,* and *Spanish* sequences, respectively. The results clearly indicate the effectiveness and the robustness of the proposed algorithm for different cinematographic styles.

Sequence	Turkish		Korean		Spanish		All	
Method	G	P	G	P	G	P	G	P
# of Shots	188	188	128	128	58	58	374	374
Correct	131	164	106	114	47	55	284	333
False	57	24	22	14	11	3	90	41
Accuracy(%)	69.7	87.2	82.8	89.1	81.0	94.8	75.9	89.0

Table 1. View classification results for three test sequences, (Method (G) uses only grass measure, while method P is the proposed method)

The ground truth for slow-motion replays includes two new sequences making the length of the set 93 minutes, which is approximately equal to a complete soccer game. The slow-motion detector uses frames at *full resolution,* and has detected 52 of 65 replay shots, 80.0% recall rate, and incorrectly labeled 9 normal motion shots, 85.2% precision rate, as replays. These results are somewhat worse than the reported results, 100% recall without explicit precision rate, in.[17] Resolution and compressed format may be accounted for the difference since the detector is sensitive to resolution and precise pixel values. The content features, such as abrupt and fast camera motions in long shots and irregular object motion in close-ups, are the main reasons for false positives (In,[17] only one soccer game that is less than a minute is used). Overall, the recall-precision rates in slow-motion detection are quite satisfactory.

4.2. Results for High-Level Analysis and Summarization

Goals are detected in 15 test sequences in the database. Each sequence, in full length, is processed to locate shot boundaries, shot types, and replays. When a replay is found, goal detector computes the cinematic template features to find goals. The proposed algorithm runs *in real-time,* and, on the average, achieves 90.0% recall and 45.8% precision rates. We believe that the three misses out of 30 goals are more important than false positives, since the user can always fast-forward false positives, which also do have semantic importance due to the replays. Two of the misses are due to the inaccuracies in the extracted shot-based features, and the miss where the replay shot is broadcast minutes after the goal is due to the deviation from the goal model. The false alarm rate is directly related to the frequency of the breaks in the game. The frequent breaks due to fouls, throw-ins, offsides, etc. with one or more slow-motion shots may generate cinematic templates similar to that of a goal. The inaccuracies in shot boundaries, shot types, and replay labels also contribute to the false alarm rate.

In Sec. 3, we explained that the existence of *referee* and *penalty box* in a summary segment, which, by definition, also contains a slow-motion shot, may correspond to certain events. Then, the user can browse summaries by these *object-based features.* The recall rate of and the confidence with referee and penalty box detection are specified for a set of semantic events in Table 2 and 3, where *recall* rate measures the accuracy of the proposed algorithms, and the *confidence* value is defined as the ratio of the number of events with that object to the the total number of such events in the clips, and it indicates the applicability of the corresponding object-based feature to browsing a certain event. For example, the confidence of observing a referee in a free kick event is 62.5%, meaning that the referee feature may not be useful for browsing free kicks. On the other hand, the existence of both objects is necessary for a penalty event due to their high confidence values. In Table 2 and 3, the first row shows the total number of a specific event in the summaries. Then, the second row shows the number of events where the referee and/or the three penalty box lines are visible. In the third row, the number of detected events is given. Recall rates in the second columns of both Table 2 and 3 are lower than those of other events. For the former, the misses are due to referee's occlusion by other players, and for the latter, abrupt camera movement during a high activity prevents reliable penalty box detection. Finally, it should be noted that the proposed features and their statistics are used for browsing purposes, not for detecting such non-goal events; hence, precision rates are not meaningful.

The compression rate for the summaries varies with the requested format. On the average, 12.78% of a game is included to the summaries of all slow-motion segments, while the summaries consisting of all goals, including all false positives, only account for 4.68%, of a complete soccer game. These rates correspond to the summaries that are less than 12 and 5 minutes, respectively, of an approximately 90-minute game.

	Yellow/Red Cards	Penalties	Free-Kicks
Total	19	3	8
Referee Appears	19	3	5
Detected	16	3	5
Recall(%)	84.2	100	100
Confidence(%)	100	100	62.5

Table 2. The statistics about the appearance of referee for some semantic events

	Shots/Saves	Penalties	Free-Kicks
Total	50	3	8
Penalty Box Appears	49	3	8
Detected	41	3	8
Recall(%)	83.7	100	100
Confidence(%)	98.0	100	100

Table 3. The statistics about the appearance of penalty box (pbox) for some semantic events

4.3. Temporal Performance

RGB to HSI color transformation required by grass detection limits the maximum frame size; hence, 4x4 spatial downsampling rates for both shot boundary detection and shot classification algorithms are employed to satisfy the real-time constraints. The accuracy of slow-motion detection algorithm is sensitive to frame size; therefore, no sampling is employed for this algorithm, yet the computation is completed in real-time with 1.6 GHz CPU speed. A commercial system can be implemented by multi-threading where shot boundary detection, shot classification, and slow-motion detection should run in parallel. It is also affordable to implement the first two sequentially, as it was done in our system. In addition to spatial sampling, temporal sampling may also be applied for shot classification without significant performance degradation. In this framework, goals are detected with a delay that is equal to the cinematic template length, which may range from 30 to 120 seconds as explained in Sec. 3.1.

5. CONCLUSION

In this paper, a new framework for summarization of soccer video has been introduced. The proposed framework allows real-time event detection by cinematic features, and further filtering of slow-motion replay shots by object-based features for semantic labeling. The implications of the proposed system include real-time streaming of live game summaries, summarization and presentation according to user preferences, and efficient semantic browsing through the summaries, each of which makes the system highly desirable. The topics for future work include: i) integration of aural and textual features to increase the accuracy of event detection; and ii) extension of the proposed framework to different sports, such as football, basketball, and baseball, which require different event and object detection modules.

6. ACKNOWLEDGMENTS

This work has been supported in part by the National Science Foundation under grant number IIS-9820721 and Eastman Kodak Company.

REFERENCES

1. S-F. Chang, "The holy grail of content-based media analysis," *IEEE Multimedia,* vol. 9, no. 2, pp.6-10, Apr.-June 2002.
2. Y. Fu, A. Ekin, A. M. Tekalp, and R. Mehrotra, "Temporal segmentation of video objects for hierarchical object-based motion description," *IEEE Trans. Image Processing,* vol. 11, no. 2, pp. 135-145, Feb. 2002.
3. D. Yow, B-L. Yeo, M. Yeung, and B. Liu, "Analysis and presentation of soccer highlights from digital video," in *Proc. Asian Conf. on Comp. Vision (ACCV),* 1995.

4. Y. Gong, L.T. Sin, C.H. Chuan, H-J. Zhang, and M. Sakauchi, "Automatic parsing of soccer programs," in *Proc. IEEE Int'l. Conf. on Mult. Comput. and Sys,* pp. 167-174, 1995.
5. S. Intille and A. Bobick, "Recognizing planned, multi-person action," *Computer Vision and Image Understanding,* vol. 81, no. 3, pp. 414-445, March 2001.
6. V. Tovinkere and R. J. Qian, "Detecting semantic events in soccer games: towards a complete solution," in *Proc. IEEE Int'l. Conf. on Mult. and Expo (ICME),* Aug. 2001.
7. G. S. Pingali, Y. Jean, and I. Carlbom, "Real time tracking for enhanced tennis broadcasts," in *Proc. IEEE Comp. Vision and Patt. Rec. (CVPR),* pp. 260-265, 1998.
8. A. Gueziec, "Tracking pitches for broadcast television," *IEEE Computer,* vol. 35, no. 3, pp. 38-43, March 2002.
9. P. Xu, L. Xie, S-F. Chang, A. Divakaran, A. Vetro, and H. Sun, "Algorithms and system for segmentation and structure analysis in soccer video," in *Proc. IEEE Int'l. Conf. on Mult. and Expo (ICME),* Aug. 2001.
10. B. Li and M. I. Sezan, "Event detection and summarization in American football broadcast video," in *Proc. of the SPIE conf. on Storage and Retrieval for Media Databases,* vol. 4676, pp. 202-213, Jan. 2002.
11. R. Leonardi and P. Migliorati, "Semantic indexing of multimedia documents," *IEEE Multimedia,* vol. 9, no. 2, pp. 44-51, Apr.-June 2002.
12. W. Zhou, A. Vellaikal, and C-C.J. Kuo, "Rule-based video classification system for basketball video indexing," in *ACM Mult. Conf.,* 2000.
13. D. Zhong and S-F. Chang, "Structure analysis of sports video using domain models," in *Proc. IEEE Int'l. Conf. on Mult. and Expo (ICME),* Aug. 2001.
14. K. A. Peker, R. Cabasson, and A. Divakaran, "Rapid generation of sports video highlights using the MPEG-7 motion activity descriptor," in *Proc. of the SPIE conf. on Storage and Retrieval for Media Databases,* vol. 4676, pp. 318-323, Jan. 2002.
15. N. Babaguchi, Y. Kawai, and T. Kitashi, "Event based indexing of broadcasted sports video by intermodal collaboration," *IEEE Trans. on Multimedia,* vol. 4, no. 1, pp. 68-75, March 2002.
16. Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," in *Proc. ACM Multimedia,* 2000.
17. H. Pan, P. van Beek, and M. I. Sezan, "Detection of slow-motion replay segments in sports video for highlights generation," in *Proc. IEEE Int'l. Conf. on Acoustics, Speech, and Signal Processing (ICASSP),* 2001.
18. K. N. Plataniotis and A. N. Venetsanopoulos, *Color image processing and applications,* Springer-Verlag, Berlin, Germany, pp. 25-32 and 260-275, 2000.
19. A. Hanjalic, "Shot-boundary detection: unraveled and resolved?," *IEEE Trans. Circuits and Sys. for Video Tech.,* vol. 12, no. 2, pp. 90-105, Feb. 2002.
20. A. Ekin and A. M. Tekalp, "A framework for analysis and tracking of soccer video," in *Proceedings of the IS&T/SPIE Conference on Visual Com. and Image Proc. (VCIP),* Jan. 2002.
21. G. Millerson, *The technique of television production,* 12^{th} Ed., Focal Publishers, March 1990.
22. A. M. Ferman and A. M. Tekalp, "A fuzzy framework for unsupervised video content characterization and shot classification," *J. of Electronic Imaging,* vol. 10, no. 4, pp. 917-929, Oct. 2001.
23. Laws of the Game and Decisions of the International Football Associations Board - 2001, www.fifa.com
24. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision,* 2^{nd} Ed., Brooks/Cole Publishing, pp. 256-260, 1999.
25. A. M. Ferman and A. M. Tekalp, "Efficient filtering and clustering methods for temporal video segmentation and visual summarization," *J. Vis. Comm. Image. Represent.,* vol. 9, no. 4, pp. 336-351, Dec. 1998.