12. Video Event Mining via Multimodal Content Analysis and Classification

Min Chen, Shu-Ching Chen, Mei-Ling Shyu, and Chengcui Zhang

Summary. As digital video data become more and more pervasive, the issue of mining information from video data becomes increasingly important. In this chapter, we present an effective multimedia data mining framework for event mining with its application in the automatic extraction of goal events in soccer videos. The extracted goal events can be used for high-level indexing and selective browsing of soccer videos. The proposed multimedia data mining framework first analyzes the soccer videos by using multimodal features (visual and audio features). Then the data prefiltering step is performed on raw video features with the aid of domain knowledge, and the cleaned data are used as the input data in the data mining process using the Nearest Neighbor with Generalization (NNG) scheme, a generalized Instance-Based Learning (IBL) mechanism. The proposed framework fully exploits the rich semantic information contained in visual and audio features for soccer video data, and incorporates a data mining process for effective detection of soccer goal events. This framework has been tested using soccer videos with different styles as produced by different broadcasters. The results are promising and can provide a good basis for analyzing the high-level structure of video content.

12.1 Introduction

With the increasing amount of digital video data, mining information from video data for efficient searching and content browsing in a time-efficient manner becomes increasingly important. Motivated by the strong interest of automatic annotation of the large amount of live or archived sports videos from broadcasters, research toward the automatic detection and recognition of events in sports video data has attracted a lot of attention in recent years. Soccer video analysis and events/highlights extraction are probably the most popular topics in this research area.

The major challenges in soccer event detection lie in the following four aspects. First, the value of sports video drops significantly after a short period of time [5], which poses the requirement of real-time (or close to real-time) processing. Second, unlike some of the other sports, such as baseball, tennis, etc., where the presence of canonical scenes (e.g., the pitching scene in baseball, the serve scene in tennis, etc.) could greatly simplify the technical challenges, soccer videos possess a relatively

loose structure. Third, the important video segments (events or highlights) in a sports video constitute only a minor portion of the whole data set. Consequently, the limited number of training data points increases the difficulties in detecting these so-called *rare events*, especially in the present of noisy data introduced during the production process. Last, but not least, the video data obtained from various sources might be inconsistent due to different production styles and postproduction effects. In other words, although some basic production rules might apply, the overall presentations vary greatly.

In the literature, many researches have been devoted to address these issues from the media content analysis [3, 12, 14, 25, 26, 33–35, 37] to the supervised classification techniques [1, 20, 21, 27, 30, 31]. An overview of the related work will be detailed in Section 12.2. However, few approaches possess the capabilities of tackling all the above-mentioned challenges. In response to these issues, in this paper, an effective multimedia data mining framework is proposed with its application on the soccer goal event detection, which seamlessly integrates the multimodal content analysis and the Nearest Neighbor with Generalization (NNG) scheme which is a generalized Instance-Based Learning (IBL) mechanism. Here, an event is defined in the shot level as the shot is widely regarded as a self-contained unit with an unbroken sequence of frames taken from one camera.

In our proposed framework, multiple cues from different modalities including audio and visual features are fully exploited and used to capture the semantic structure of soccer goal events. Then the NNG scheme is applied for event detection. Currently, most existing classification techniques adopted in the event detection area are called *model-based* approaches as they compute the global approximation (or called *model*) of the target classification function, which is then used to classify the unseen testing data. In contrast, the IBL mechanism is called *lazy* method in the sense that the generalization of the observed (training) data delays until each new query instance is encountered [24]. Therefore, it can use the query instance for selecting a local approximation to the target classification function [13] each time when a query instance is given. The importance of incorporating the query instance lies in the fact that the current production style is one of the key factors in determining the pattern of a targeted event. Therefore, by adopting the IBL mechanism, we direct our focus on the instances themselves rather than on the rules that govern their attribute values. In addition, in response to the requirements of real-time processing and rare event detection, a data prefiltering step is integrated in the IBL mechanism to perform on the raw video features with the aid of the specific domain knowledge. We have evaluated the performance of the proposed framework by using a large amount of soccer video data with different styles and different broadcasters. The experimental results demonstrate the effectiveness and the generality of our proposed framework.

The contributions of the proposed framework are summarized as follows:

- First, an advanced video shot detection method is adopted in this work, which can not only output the shot boundaries, but also generate some important visual

features during the process of shot detection. Moreover, since object segmentation is an embedded subcomponent in video shot detection, the higher level semantic information, such as the grass areas, which serves as an important indication in soccer goal detection, can be derived from the object segmentation results. Therefore, just a small amount of work needs to be done in order to extract the visual features for each shot, which distinguishes our framework from most of the other existing approaches.

- Second, when choosing the proper data mining technique, we take into consideration the data inconsistency posed by various production types. In other words, with widely varied production styles and preferences, it is difficult to achieve a global approximation accurately with regard to the event patterns as targeted by the model-based approaches. In contrast, the IBL mechanism defers the decision-making process until the presence of the new query instance, where the local optimization for this particular instance is considered.
- Third, the proposed data prefiltering step is critical to apply the IBL mechanism to this specific application domain when considering the real-time processing requirement, the influence of the noisy data, and the small percentage of the positive samples (goal shots) compared to the huge amount of negative samples (nongoal shots) in soccer video data. To our best knowledge, there is hardly any work addressing this issue.

The chapter is organized as follows. Section 12.2 gives an overview of the related work. In Section 12.3, the proposed multimedia data mining framework is discussed in details. Experimental results are presented and analyzed in Section 12.4. Finally, Section 12.5 concludes our study and presents some future research directions.

12.2 Related Work

Research work has been conducted to study the respective roles of visual [14, 26], auditory [25, 33], and textual [3] modalities in sports video analysis. Recently, the approaches using multimodal analysis have drawn increasing attentions [12, 37] as the content of a video is intrinsically multimodal and its meaning is conveyed via multiple channels. For instance, in [12], a multimodal framework using combined audio/visual/text cues was presented, together with a comparative analysis on the use of different modalities for the same purpose. However, the use of the textual transcript is not always available although it contains rich semantic information for event identification. In addition, to boost robustness against the variations in low-level features and to improve the adaptability of event detection schemes, mid-level representation has also been used in event detection, including the camera view types (global, medium, or close-up) [32], audio key words [15, 33], etc. Therefore, in our framework, multimodal content analysis is carried out in the audio and visual channels, where both low-level and mid-level features are explored.

Despite numerous efforts in video content analysis, it remains a major challenge in terms of effectively integrating the multiple physical features to infer the semantic events due to the well-known semantic gap. In response to this issue, some research efforts have been directed to extend the basic content analysis methods with the facilitation of more supervised approaches, such as heuristic method [21], E-R model [27], and Hidden Markov Model (HMM) [2]. In [21], a set of fixed rules is derived on the basis of the multimodal cues. However, the derivation process becomes infeasible with the increment of the number of multimodal features. In addition, the fixed thresholds adopted in the rules are not general enough for a large number of video samples. In [27], Tovinkere and Qian proposed a hierarchical E-R model on the basis of 3D data of the locations of players and ball, trying to model the semantic meaning and domain knowledge for soccer games. A set of rules is thereafter generated to determine the occurrence of the event. However, the generalization of this work is highly limited as the 3D information is not generally available in the video data. In [2], a method to detect and recognize soccer highlights using Hidden Markov Model (HMM) was proposed, in which each model is trained separately for each type of event. As shown in their preliminary results, this method can detect and recognize free kick and penalty event. However, it has the problem to deal with long video sequences.

More recently, data mining approaches, with their promising capabilities in discovering interesting patterns from large data sets, have been evolved to support fully automated event detection. In our earlier studies, the PRISM classification rule algorithm [4, 8] and decision-tree learning method [10] were applied for event detection. However, the rules induced by the PRISM algorithm may not exclude each other and possess no execution priority order. Such situations are called conflicts and are difficult to cope with in the real application. Alternatively, the decision-tree learning algorithm was applied in our recent work [10], which avoids the conflicts by adopting the divide-and-conquer approach. Meanwhile, in [38], the multilevel sequential association mining is introduced to explore associations among the audio and visual cues, classify the associations by assigning each of them with a class label, and use their appearances in the video to construct video indices. However, the source video clips are required to have all the commercials removed.

To our best knowledge, almost all the classification methods adopted in the video event detection area are model-based approaches, which present some good qualities when the video data are with a high level of consistency. For instance, they are generally with an explicit model representation and computationally efficient for new data classification. However, they inevitably suffer from the data inconsistency problem in the sense that they are targeted to achieve the global approximation. The IBL mechanism, on the other hand, is capable of dealing with this issue at the possible cost of the high computational requirement, implicit rule representation, and noise sensitivity. Therefore, in our framework, the Nearest Neighbor with Generalization (NNG) [29], a generalized IBL mechanism with a nearest neighbor like algorithm using nonnested generalized exemplars, is adopted together with the proposed prefiltering process to overcome these obstacles.

12.3 Goal Shot Detection

12.3.1 Instance-Based Learning

Instance-based learning (IBL) is a conceptually intuitive approach to approximate the real-valued target functions, which in our case are the classification functions. In brief, the learning in IBL starts with the storage of the presented training instances $S = \{<I_i, L_i>, i = 1, 2, \ldots, N\}$. Here, $<I_i, L_i>$denotes a training instance, where I_i is represented by an attribute set $\{a_{ij}\}$ with the size of T (i.e., $j = 1, 2, \ldots, T\}$ and L_i is the class label, and N indicates the number of instances in the training set. In our case, $L_i \in L$, where $L = \{yes, no\}$ denotes the two class labels of the instances. The class label "*yes*" indicates a goal event and "*no*" indicates it is not a goal event. As opposed to the model-based approach where S is used to construct a (parametric) model and is then discarded, the training instances contribute in a more direct way to the inference result. More specifically, when a new query instance $Q = \{a_{qj}, j = 1, 2, \ldots, T\}$ is encountered, the relationship of the query instance Q with the instances from S is examined to assign a target function value (class label L_q) for Q. As illustrated in Figure 12.1, the simplest way to define the relationship is to apply a certain distance metric (e.g., Euclidean distance, Manhattan or city-block metric, etc.) and the class of Q is set to L_x, where L_x is the label of the closest instance $I_x \in S$ in terms of Q.

However, several key aspects or issues have to be taken into considerations in this simple scheme.

1) How to define the attribute set $\{a_{ij}\}$ for each instance?
2) A basic distance metric, say Euclidean distance, between two instances Q and I_x is defined as follows:

$$|Q, I_x| = \sqrt{\left(a_{q1} - a_{x1}\right)^2 + \left(a_{q2} - a_{x2}\right)^2 + \cdots + \left(a_{qT} - a_{xT}\right)^2} \tag{12.1}$$

As can be seen from Eq. (12.1), each attribute will have exactly the same influence on the decision making, which is generally not the case for video event detection. Therefore, how to derive suitable attribute weights from the training set becomes an essential problem improve the distance metric.

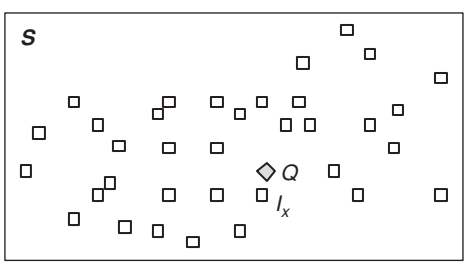


Fig. 12.1. The basic IBL mechanism.

3) The scheme is sensitive to the noisy data. For instance, if instance I_x is corrupted by the noise, instance Q might be misclassified. Therefore, a solution must be sought to dramatically reduce the effect of the noise.
4) In the current scheme, it would be quite time-consuming for a data set with a realistic size because all the training instances $I_i \in S$ need to be scanned in order to classify each test instance Q. The computational cost must be greatly reduced for the sake of real-time processing.
5) Different from the model-based approaches that present the knowledge explicitly by the constructed models, such as the rules in PRISM and the tree structure in the decision tree algorithm, the knowledge is expressed implicitly in IBL, which usually impedes the problem understanding.

In the following sections, all the above-mentioned issues will be detailed. More specifically, we will discuss the video feature extraction in Section 12.3.2 to address the first aspect, i.e., the construction of the attribute set $\{a_{ij}\}$. Then the remaining issues are tackled by the prefiltering process and the generalized IBL scheme in Sections 12.3.3 and 12.3.4, respectively.

12.3.2 Multimodal Analysis of Soccer Video Data

12.3.2.1 Shot-Based Video Event Mining

There are two widely used schemes for modeling and mining videos—shot-based approach and object-based approach. The shot-based approach divides a video sequence into a set of collections of video frames with each collection representing a continuous camera action in time and space, and sharing the similar high-level features (e.g., semantic meaning) as well as similar low-level features like color and texture [16]. In the object-based modeling approach, temporal video segments representing the life-span of the objects as well as some other object-level features are used as the basic units for video mining. Object-based modeling is best suitable where a stationary camera is used to capture a scene (e.g., video surveillance applications). In such a setting, shot-based modeling is not applicable since there is one and only one long shot according to the traditional shot definition [16]. In contrast, a soccer video sequence typically consists of hundreds of shots, with their durations ranging from seconds to minutes. Although an event boundary does not necessarily coincide with a shot boundary in soccer videos, there are several good reasons for using shot-based event detection for soccer videos:

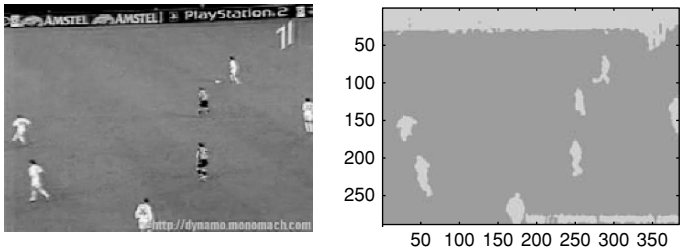
1) First of all, the occurring of certain soccer events is often indicated by the visual/audio clues in a shot with some temporal constraints. For example, a corner kick event typically starts a new shot in which both the player and a corner of the playfield are present followed by a camera pan during the first seconds of that shot. As another example, a goal shot is usually followed by another shot showing the excitement of the commentator and the crowd. A foul event is usually accompanied by a close-up shot of a referee.

2) Shot-based modeling, while it conforms to the hierarchical (i.e., scene/shot hierarchy) semantic modeling of video data for easy browsing and searching, also provides important visual cues critical for event mining during the process of shot detection. In addition, based on our experience with approximately 30 soccer videos with different production styles, audio cues, especially the crowd noise level, tend to be more consistent within a shot.

For the above reasons, in this chapter, we focus on the shot-based approach for event mining in soccer videos. Although shot detection has a long history of research, it is not a completely solved problem [19], especially for sports videos. According to [17], due to the strong color correlation between soccer shots, a shot change may not be detected since the frame-to-frame color histogram difference is not significant. Second, camera motions and object motions are largely present in soccer videos to track the players and the ball, which constitute a major source of false positives in shot detection. Third, the reliable detection of gradual transitions, such as fade in/out, is also needed for sports videos. We also need to take into consideration the requirements of real-time processing as it is essential for building an efficient sports video management system.

In this chapter, the visual feature extraction is based on video shots. Thus, a three-level filtering architecture is used for shot detection, namely *pixel-histogram comparison*, *segmentation map comparison*, and *object tracking*. The pixel-level comparison basically computes the differences in the values of the corresponding pixels between two successive frames. This can, in part, solve the strong color-correlation problem because the spatial layout of colors also contributes to the shot detection. However, though simple as it is, it is very sensitive to object and camera motions. Thus, to address the second concern of camera/object motions, the histogram-based comparison is added to pixel-level comparison to reduce its sensitivity to small rotations and slow variations. However, the histogram-based method also has problems. For instance, two successive frames will probably have the similar histograms but with totally different visual contents. On the other hand, it has difficulty in handling the false positives caused by the changes in luminance and contrast. The reasons of combining the pixel-histogram comparison in the first level filtering are twofolds. (1) Histogram comparison can be used to exclude some false positives due to the sensitivity of pixel comparison, while it would not incur much extra computation because both processes can be done in one pass for each video frame. Note that the percentage of changed pixels (denoted as *pixel_change_percent*) and the histogram difference (denoted as *histo_change*) between consecutive frames, obtained in pixel-level comparison and histogram comparison respectively, are important indications for camera and object motions and can be used to extract higher level semantics for event mining. (2) Both of them are computationally simple. By applying a relatively loose threshold, we can ensure that most of the correct shot boundaries will be included, and in the meanwhile, a much smaller candidate pool of shots is generated at a low cost.

We take the third observation into account by introducing two other filters, namely *segmentation map comparison* and *object tracking*, which are implemented on the



(a) (b)

Fig. 12.2. An example segmentation mask map. (a) An example soccer video frame; (b) the segmentation mask map for (a).

basis of an unsupervised object segmentation and tracking method proposed in our previous work [6, 7]. A novel feature introduced is the *segmentation mask map* of a video frame, which can be automatically extracted and contains the segmentation result of that frame. In other words, a *segmentation mask map* contains the significant objects or regions of interests extracted from that video frame. Thus, the pixels in each frame have been grouped into different classes (e.g., 2 classes), corresponding to the foreground objects and background areas, respectively. Then two frames can be compared by checking the differences between their segmentation mask maps. An example segmentation mask map is given in Figure 12.2. The segmentation mask map comparison is especially effective in handling the fade in/out effects with drastic luminance changes and flash light effects [9]. In addition, to better handle the situation of camera panning and tilting, the object tracking technique based on the segmentation results is used as an enhancement to the basic matching process. Since the segmentation results are already available, the computation cost for object tracking is almost trivial compared to those manual template-based object tracking methods. It needs to be pointed out that there is no need to do object segmentation for each pair of consecutive frames. Instead, only the shots in the small candidate pool will be fed into the segmentation process. The performance of segmentation and tracking is further improved by using incremental computation together with parallel computation [36]. As a result, the combined speed-up factor can achieve 100–200. The time for segmenting one video frame ranges from 0.03 to 0.12 second depending on the size of the video frames and the computer processing power.

12.3.2.2 Visual Feature Analysis and Extraction

In the proposed framework, multimodal features (visual and audio) are extracted for each shot based on the shot boundary information obtained in the shot detection step. The proposed video shot detection method can not only detect shot boundaries, but also produce a rich set of visual features associated with each video shot. For examples, the pixel-level comparison can produce the percentage of changed pixels between consecutive frames, while the histogram comparison provides us with the histogram differences between frames, both of which

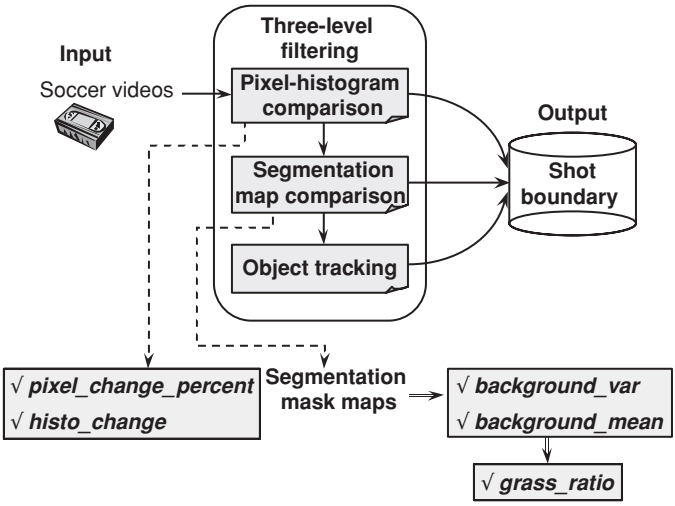


Fig. 12.3. Visual feature analysis and extraction.

are very important indications for camera and object motions. In addition, the object segmentation can further provide us with the higher level semantic information such as the object locations and foreground/background areas. By taking these advantages brought by video shot detection, we include the following five visual features in our multimedia data mining framework for soccer goal detection, namely *pixel_change_percent*, *histo_change*, *background_mean*, *background_var*, and *grass_ratio*. Here, *pixel_change_percent* denotes the average percentage of the changed pixels between the consecutive frames within a shot. Similarly, *histo_change* represents the mean value of the frame-to-frame histogram differences in a shot. Obviously, as illustrated in Figure 12.3, *pixel_change_percent* and *histo_change* can be obtained simultaneously and at a low cost during the video shot detection process. As mentioned earlier, both features are important indications of camera motion and object motion. For example, a close-up shot with a high *pixel-change* value and a low *histo_change* value usually indicates the object motion but a slow camera motion. Usually in global shots, the visual effects of object motion or camera motion are not that significant, and thus, low values for both *pixel_change* and *histo_change* can be observed.

While *pixel_change_percent* and *histo_change* can be easily obtained, the *grass_ratio* feature is derived from the *background_var* and *background_mean* features which can be obtained via object segmentation (see Figure 12.3). *grass_ratio* is an important domain-specific feature for soccer highlights detection [12]. As we can see from Figure 12.4 (a) and (b), a large amount of grass areas are present in global shots (including **goal** shots), while less or hardly any grass areas are present in the mid- or the close-up shots (including the cheering shots following the goal shots). Another observation is that the global shots usually have a much longer duration than the close-up shots. In this study, the mean value of *grass-ratio* within a shot is used to indicate the shot type (global, close-up, etc.). However, it is a challenge to

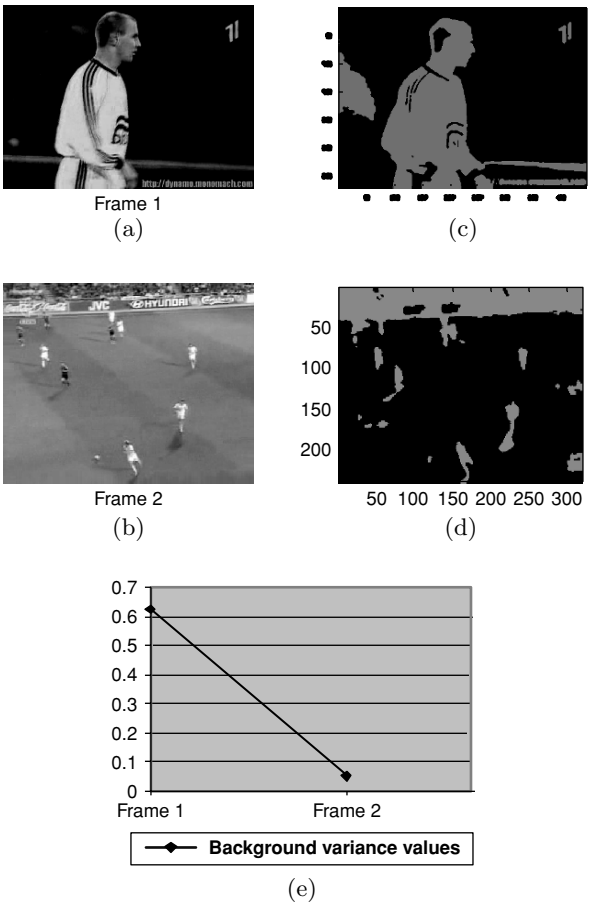


Fig. 12.4. (a) A sample frame from a goal shot (global view); (b) a sample frame a close-up shot; (c) object segmentation result for (a); (d) object segmentation result for (b); and (e) background variance values for frame 1 and frame 2.

distinguish the grass colors from others because the color values may change under different lighting conditions, different play fields, different shooting scales, etc. The method proposed in [18] relies on the assumption that the play field is always green to extract the grass areas, which is not always true for the reasons mentioned above. The methods based on the dominant color for grass area detection are more robust [17]. Our proposed method also does not assume any specific value for the play field color. The proposed grass area detection and feature extraction process is conducted in the following steps.

First, the object segmentation component from the *segmentation map comparison* filter (see Figure 12.3) is used to segment the video frames drawn at the 50-frame interval into the background areas (grass, crowd, etc.) and foreground areas (player,

ball, etc.). It is worth noting that the segmentation is conducted in the HSV color space since it is a proven perceptual color space particularly amenable to color image analysis [11]. As shown in Figure 12.4 (c) and (d), the foreground areas are marked with the gray color and the background areas are marked with the black color. It can be observed that the grass field tends to be much smoother in terms of its color and texture distributions. Thus, for each frame, the color variance of each class is captured using the standard deviation of its pixels' values. The class with a smaller color variance is called background, and the mean and variance of background pixels are recorded for each frame. As the segmentation mask maps shown in Figure 12.4, in the global view frames (see Figure 12.4(b)), the grass area tends to be detected as the background with low background variance values (see Figure 12.4(e)). On the other hand, in close-up frames (see Figure 12.4(a)), the background is very complex and may contain crowd, signboard, etc., resulting in higher background variance values, as can be seen from Figures 12.4(c) and 12.4(e). Therefore, the background is considered as a candidate grass area if its *background_var* is less than a small threshold. The *grass_ratio* of that frame is then set temporarily to the ratio of the background area within that frame.

The second step is to select reference frames to learn the field colors. All the frames containing candidate grass areas identified in previous step are considered as the reference frames. The *background_mean* value of a reference frame actually represents the mean color value of a candidate grass area. Thus, their corresponding *background_mean* values are collected, and the color histogram is then calculated over the pool of the possible field colors collected for a single video clip. However, prior to the histogram calculation, a prefiltering step is needed to filter out the outliers in the candidate pool by taking out those shots that are too short and those shots whose *background_mean* values are out of a reasonable scope of the average *background_mean.* Another way to improve this could be to select those reference frames with their *grass_ratio* values greater than a threshold, and such frames are more likely to come from global shots. Based on our observations on a large set of video data, there are two possible situations in the histogram: (1) there is a single peak in the histogram, indicating a good video quality and stable lightning conditions, and (2) there are multiple peaks in the histogram, which correspond to the variations in grass colors caused by camera shooting scale and lightning condition. For example, Figure 12.5 depicts the histogram distribution of *background_mean* values of the reference frames from a 20-minute long soccer video sequence. It can be observed from this figure that most of the *background_mean* values of the reference frames fall into two major histogram bins. By carefully studying the data for this video, we found that the reference frames from the close-up shots form the left peak in Figure 12.5; while the right peak mainly consists of reference frames from the global shots. We can also tell from this figure that the number of close-up reference frames is much smaller than that of the global reference frames. This conforms to the observation that the global shots usually have a much longer duration than the close-up shots. In situation (1), the single peak is selected as the grass pixel detector to calculate the actual *grass_ratio* for each sample frame; while in situation (2), multiple peaks within a reasonable range are all selected as grass detectors. The threshold for selecting the histogram peaks can be adjusted.

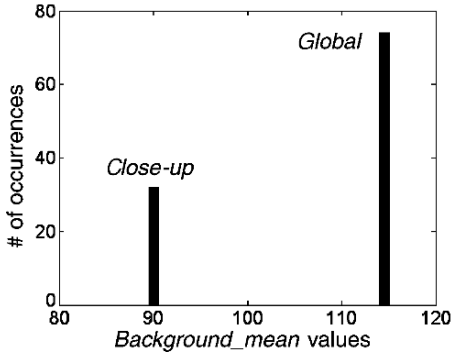


Fig. 12.5. The histogram of the candidate grass values for a 20-minute long soccer video. Two peaks correspond to two major types of shooting scales in the video data—global and close-up.

Figure 12.6 shows the detected grass areas for two sample frames from different types of shots (close-up, global, etc.), and the results are very promising.

Thus, the shot-level features *background_var*, *background_mean*, and *grass_ratio* are computed as the mean values of the corresponding frame-level features within the shot. It is worth noting that the proposed grass area detection method is unsupervised and the grass values are learned through unsupervised learning within each video sequence, which is invariant to different videos.

The major theoretical advantages of our approach are summarized as follows.

1) The proposed method allows the existence of multiple dominant colors, which is flexible enough to accommodate variations in grass colors caused by different camera shooting scales and lightning conditions.

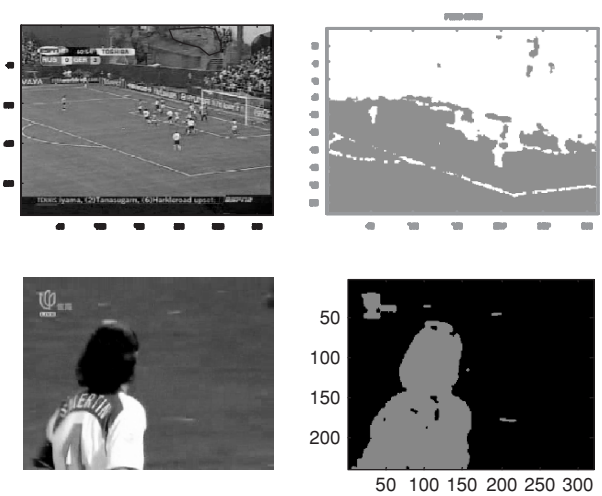


Fig. 12.6. Detected grass areas (black areas on the right column) for two sample video frames.

2) In the learning process, the proposed method adopts an automated and robust approach to choose the appropriate reference frames for the learning process.

While the existing dominant color-based methods tend to ignore the fact that the nongrass areas may have the similar color (e.g., signboards, player clothes, etc.) to that of the filed and thus introduce false positives, the propose method uses an advanced strategy to obtain the *grass_ratio* value at the region level instead of the pixel level to alleviate this problem with minor extra effort. For example, as can be seen from the upper-left sample frame in Figure 12.6, the green area at the top of the frame (the area inside the blue polygon) is correctly identified as nongrass area due to the good localization properties of the segmentation method [7].

12.3.2.3 Audio Feature Analysis and Extraction

Extracting effective audio features is essential in achieving a high distinguishing power in audio content analysis for video data. A variety of audio features have been proposed in the literature for audio track characterization [22, 28]. Generally, they fall into two categories: time domain and frequency domain. With respect to the requirements of specific applications, the audio features may be extracted at different granularities such as frame level and clip level. In this section, we describe several features that are especially useful for classifying audio data.

The proposed framework exploits both time-domain and frequency-domain audio features. To investigate the semantic meaning of an audio track, the high-level features representing the characteristics of a comparable longer period are necessary. In our case, we explore both clip-level features and shot-level features, which are obtained via the analysis of the finer granularity features such as frame-level features. In this framework, the audio signal is sampled at 16,000 Hz, i.e., 16,000 audio samples are generated for a 1-s audio track. The sample rate is the number of samples of a signal that is taken per second to represent the signal digitally. An audio track is then divided into clips with a fixed length of 1 s. Each audio feature is first calculated on the frame level. An audio frame is defined as a set of neighboring samples which last about 10–40 ms. Each frame contains 512 samples shifted by 384 samples from the previous frame as shown in Figure 12.7. A clip thus includes around 41 frames. The audio feature analysis is then conducted on each clip (e.g., an audio feature vector is calculated for each clip).

12.3.2.4 Audio Feature Analysis

The generic audio features utilized in our framework can be divided into three groups, namely, *volume related features*, *energy related features*, and *Spectrum Flux related features*.

- *Feature 1: Volume.* Volume is one the most frequently used and the simplest audio features. As an indication of the loudness of sound, volume is very useful for soccer video analysis. Volume values are calculated for each audio frame. Figure 12.8 depicts samples of two types of sound tracks: speech and music. For speech, there are local minima which are close to zero interspersed between high values. This

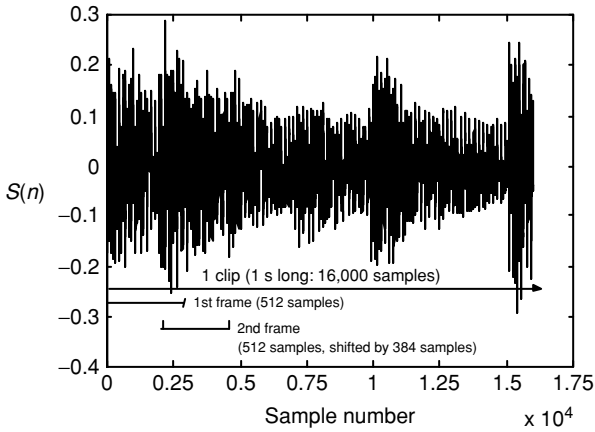
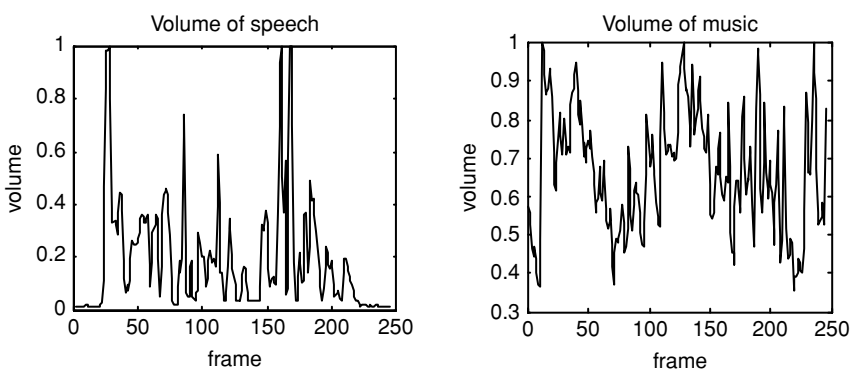


Fig. 12.7. Clip and frames used in feature analysis.

is because when we speak, there are very short pauses in our voice. Consequently, the normalized average volume of speech is usually lower than that of music. Thus, the volume feature will help not only identify exciting points in the game but also distinguish commercial shots from regular soccer video shots. According to these observations, four useful clip-level features related to volume can be extracted: (1) average *volume (volumn_mean)*; (2) *volumn_std,* the standard deviation of the volume, normalized by the maximum volume; (3) *volumn_stdd*, the standard deviation of the frame to frame difference of the volumes, and (4) *volume_range*, the dynamic range of the volume, defined as $(\max(v) - \min(v))/\max(v)$.

- *Feature 2: Energy.* Short-time energy means the average waveform amplitude defined over a specific time window. In general, the energy of an audio clip with music content has a lower dynamic range than that of a speech clip. The energy of a speech



(a) Speech (b) Music

Fig. 12.8. Volume of audio data.

clip changes frequently from high peaks to low peaks. Since the energy distribution in different frequency bands varies quite significantly, energy characteristics of subbands are explored as well. Four energy subbands are identified, which cover respectively the frequency interval of 1Hz-(fs/16)Hz, (fs/16)Hz-(fs/8)Hz, (fs/8)Hz-(fs/4)Hz, and (fs/4)Hz-(fs/2)Hz, where *fs* is the sample rate. Compared to other subbands, subband1 (1Hz-(fs/16)Hz) and subband3 ((fs/8)Hz-(fs/4)Hz) appear to be most informative. Several clip-level features over subband1 and subband3 are extracted as well. Thus, the following energy-related features are extracted from the audio data: (1) *energy_mean,* the average RMS (Root Mean Square) energy; (2) The average RMS energy of the first and the third subbands, namely *sub1_mean* and *sub3_mean*, respectively; (3) *energy_lowrate,* the percentage of samples with the RMS power less than 0.5 times of the mean RMS power; (4) The energy-lowrates of the first subband and the third band, namely *sub1_lowrate* and *sub3_lowrate*, respectively; and (5) *sub1_std*, the standard deviation of the mean RMS power of the first subband energy.

- *Feature 3: Spectrum Flux.* Spectral Flux is defined as the two norms of the frame-to-frame spectral amplitude difference vector. Spectrum flux is often used in quick classification of speech and nonspeech audio segments. In this study, the following Spectrum Flux related features are explored: (1) *sf_mean*, the mean value of the Spectrum Flux; (2) the clip-level features *sf_std,* the standard deviation of the Spectrum Flux, normalized by the maximum Spectrum Flux; (3) *sf_stdd*, the standard deviation of the difference of the Spectrum Flux, which is also normalized; and (4) *sf_range*, the dynamic range of the Spectrum Flux.

Please note that the audio features are captured at different granularities: frame-level, clip-level, and shot-level, to explore the semantic meanings of the audio track. Totally 15 generic audio features are used (4 volume features, 7 energy features, and 4 Spectrum Flux features) to form the 15 out of 17 components of an audio feature vector for a video shot. Another two audio features are directly derived from the volume related features. For each shot, the feature *sumVol* keeps the summation of the peak volumes of its last 3-s audio track and its following shot's first 3-s track (for short, nextfirst3). Then the mean volume of its nextfirst3 forms another audio feature *vol_nextfirst3*.

Once the proper video features and audio features have been extracted, they are ready to be fed into the prefiltering step which is critical to the performance of the proposed multimedia data mining framework for the reasons as discussed in next subsection.

12.3.3 Prefiltering

As mentioned earlier, the obtained feature set may contain noisy data that were introduced during the video production process. Moreover, the data amount is typically huge and among them the number of goal shots only accounts for less than 1% in our case. For the sake of accuracy and efficiency, it is judicious and essential to reduce the density of exemplars that lie well inside the class boundaries whereas keep the data points near the boundaries [29]. The basic idea is illustrated in Figure 12.9.

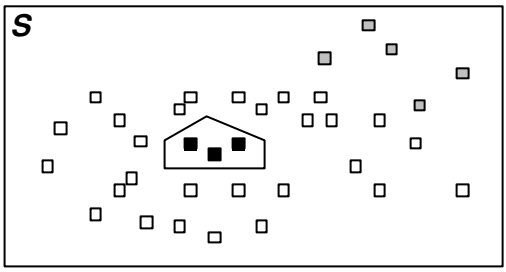


Fig. 12.9. Prefiltering.

In Figure 12.9, let the black cubes be the positive instances whereas the white and gray points represent the negative exemplars which are close to or far away from the class boundary (the black polygon), respectively. It is intuitive that with the removal of the gray points, the classification results for any unseen data remain the same while the computational cost can be reduced.

Therefore, a prefiltering process is proposed to clean the data and to select a small set of training exemplars using domain knowledge. Here, domain knowledge is defined as the empirically verified or proven information specific to the application domain that is served to reduce the problem or search space [29]. In general cases, both positive and negative exemplars can be reduced on the basis of the above-mentioned principle. However, as discussed earlier, the interested events (i.e., goal events in our study) in the soccer video are quite scarce and are of great importance. Consequently, the intention of the proposed prefiltering process is to greatly reduce the irrelevant negative exemplars (nongoal events). Furthermore, as discussed in the Introduction section, although some basic rules exist, the overall presentations of the videos are generally varied from each other. Therefore, it would normally be a major challenge in terms of achieving an effective trade-off between the generality and specialty possessed by the source data, which is in fact one of the key reasons that a global approximation might fail to capture. Fortunately, the ultimate goal of this process is to remove the negative exemplars far away from the class boundaries (for short, they will be called far-negative exemplars from now on). Therefore, in this section, we present this prefiltering process using some computable observation rules with loose thresholds on the soccer videos, which can be classified into two categories, namely audio rules and visual rules.

12.3.3.1 Audio Rules

In the soccer videos, the sound track mainly includes the foreground commentary and the background crowd noise. According to the observation and prior knowledge, the commentator and crowd become excited at the end of a goal shot. In addition, different from other sparse happenings of excited sound or noise, normally this kind of excitement will last to the following shot(s). Thus, the duration and intensity of sound can be used to capture the candidate goal shots and remove the far-negative exemplars as defined in the following rule:

- **Rule 1**: As a candidate goal shot, the audio track of its last 3 (or less) seconds and that of the first 3-s (or less) of its following shot should both contain at least one exciting point.

Here the exciting point is defined as a 1-s period whose volume is larger than 60% of the highest 1-s volume in this video. It is worth mentioning that actually this volume threshold can be assigned to an even higher value for most of the videos. However, based on our experiments, 60% is a reasonable threshold since the number of the candidate goal shots can be reduced to 28% of the whole search space while including all the goal shots. In addition, this rule performs as a data cleaning step to remove some of the noise data because, though normally the noise data has a high volume, it will not last for long.

12.3.3.2 Visual Rules

As mentioned earlier, we have two basic types of shots, close-up shots and global shots, for soccer videos based on the ratio of the green grass area. We observe that the goal shots belong to the global shots with a high grass ratio and are always closely followed by the close-up shots that include cutaways, crowd scenes, and other shots irrelevant to the game without grass pixels, as shown in Figure 12.10. Figure 12.10(a)–(c) capture three consecutive shots starting from the goal shot (Figure 12.10(a)), and Figure 12.10(d)–(f) show another three consecutive shots where Figure 12.10(d) is the goal shot. As can be seen from this figure, within two consecutive shots that follow the goal shot, usually there is a close-up shot (Figures 12.10(b) and (f), respectively).

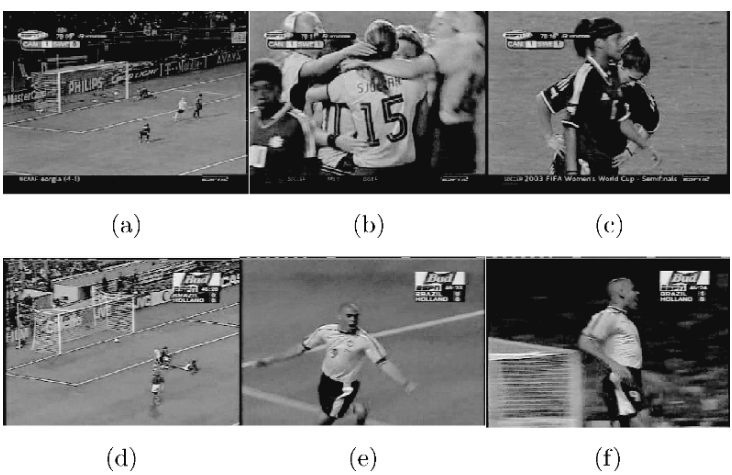


Fig. 12.10. Goal shots followed by close shots: (a)–(c) are three consecutive shots in a goal event, where (a) is the goal shot and (b) is the close shot that follows the goal shot; (d)–(f) are another goal event and its three consecutive shots, where (d) is the goal shot and (f) is the close shot follows the goal shot.

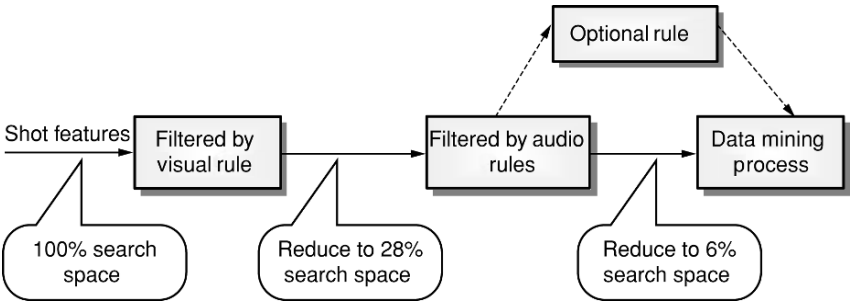


Fig. 12.11. Prefiltering process.

According to these observations, two rules are defined as follows:

- **Rule 2**: A goal shot should have a grass ratio larger than 40%.
- **Rule 3**: Within two succeeding shots that follow the goal shot, at lease one shot should belong to the close-up shots.

Note that the threshold defined in Rule 2 can be altered to a higher value for most of the videos. However, our experiments show that about 80% of the candidate pool obtained after applying Rule 1 can be reduced using Rule 2 and Rule 3, which means that only less than 6% of the whole search space remains as the input for the data mining process. In addition, according to the prior knowledge, a goal shot normally lasts more than 3 s, which can be used as an optional filter called Optional Rule. In our case, since the search space has been dramatically reduced, this rule has small effects. In summary, the workflow as well as the performance of the prefiltering process is illustrated in Figure 12.11.

In summary, the prefiltering process is mainly targeted to solve the fourth problem mentioned above. In other words, by reducing the number of far-negative exemplars for the data mining process, the computational cost can be dramatically decreased. In addition, some noisy data can also be filtered out by the audio rule.

12.3.4 Nearest Neighbor with Generalization (NNG)

As mentioned earlier, the basic IBL mechanism is sensitive to the noisy data. Although the prefiltering process is capable of removing part of the noisy data, a generalized IBL, called Nearest Neighbor with Generalization (NNG) [23], is adopted in our framework to overcome this limitation.

Let T be the number of attributes in the attribute set, the basic idea for the NNG algorithm is to create a group of T-dimensional rectangles or so-called hyper-rectangles, where each of them covers a portion of examples without overlap. To simplify the idea, Figure 12.12 shows a two-dimensional instance space with two classes of instances (represented by the black and white cubes, respectively), and the possible rectangular regions are created as the results of the generalization process. For more detailed information about the generalization process, the interested readers can refer to [23].

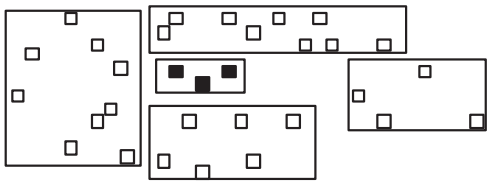


Fig. 12.12. The basic idea of the NGG algorithm.

In brief, NNG represents a trade-off between the specificity of the basic IBL scheme and the generality of the model-based approach. More specifically, the hyperrectangles can be considered as rules, which, when extracted, can facilitate knowledge understanding. Given an unknown instance Q, it will be assigned the corresponding class label if it falls within one of these rectangles, which is similar to the general rule induction approaches. Otherwise, the usual nearest neighbor rule applies except that the distances are calculated between Q and each of the rectangles (instead of the instances). Formally, the difference between the instance Q and a rectangle R with regard to its ith attribute (denoted as a_{qi} and R_i, respectively) is defined as follows:

$$\left|a_{qi} - R_i\right| = \begin{cases} a_{qi} - R_i^{\max} & \text{if } a_{qi} > R_i^{\max} \\ R_i^{\min} - a_{qi} & \text{if } a_{qi} < R_i^{\min} \\ 0 & \text{otherwise} \end{cases} \tag{12.2}$$

Here, $R_i^{\max}$ and $R_i^{\min}$ denote the boundaries of the hyperrectangle R with regard to the ith feature. As the distance is measured to a group of data points instead of one single instance, it will greatly reduce the adverse effect of the noisy data, as long as the noisy data accounts for a reasonable small portion of the instance set.

As far as the distance metric is concerned, as mentioned earlier, the basic Euclidean function considers all the attributes with an equal relevance in decision making. However, it is generally the case in event detection (and many of the other domains) that the significance of the attributes with regard to the outcome varies from each other. Therefore, an intuitive improvement toward the Euclidean function is to introduce the attribute weights as $w_1, w_2, \ldots$, and w_T, and Eq. (12.1) can be redefined to calculate the distance between instance Q and the hyperrectangle R as follows:

$$|Q, R| = \sqrt{w_1^2 \left|a_{q1} - R_1\right|^2 + w_2^2 \left|a_{q2} - R_2\right|^2 + \cdots + w_T^2 \left|a_{qT} - R_T\right|^2} \tag{12.3}$$

In our work, the dynamic feature weighting scheme proposed in [1] is adopted to define the weights.

12.4 Experimental Results and Discussions

12.4.1 Video Data Source

Soccer game video files used in our experiments were collected from a wide range of sources via the Internet. After excluding those video files that either have poor

digital quality or do not contain any goal scene, there are 25 video files left, with different styles and produced by different broadcasters. Those video files last from several minutes to more than half an hour. The corresponding sound tracks have been extracted from the original video files.

12.4.2 Video Data Statistics and Feature Extraction

12.4.2.1 Information Collecting

Information such as the total number of video frames and the durations is necessary and can be obtained by using video editing software. The average duration and number of frames are about 22 min and 35,000 frames, respectively.

To facilitate the extraction of audio and visual features that represent the media contents, shot boundaries need to be located, which is achieved by parsing the video files using the proposed shot detection algorithm. Because of the good performance of the shot detection algorithm (with >92% precision and >98% recall value), only little effort is needed to correct the shot boundaries. On an average, those soccer video files contain about 184 shots. The detailed statistics of all the video files are listed in Table 12.1.

Table 12.1. Detailed statistics of all the video data files

Files	Frame #	Shot #	Duration ([hours:] minutes: seconds)	Goal #
File 1	30,893	148	20:36	2
File 2	20,509	83	13:40	1
File 3	23,958	93	15:58	4
File 4	42,083	194	23:24	2
File 5	48,153	346	32:6	2
File 6	14,612	96	9:44	1
File 7	13,122	106	8:45	1
File 8	51,977	418	34:21	1
File 9	49,959	238	33:18	1
File 10	41,817	212	27:53	1
File 11	46,472	230	30:59	3
File 12	27,624	149	18:25	2
File 13	35,283	150	23:31	2
File 14	22,230	95	14:49	1
File 15	15,198	129	10:8	1
File 16	40,973	322	27:19	3
File 17	19,149	119	12:46	1
File 18	33,943	137	18:53	1
File 19	43,320	173	24:5	2
File 20	65,677	294	36:31	2
File 21	32,788	125	18:14	1
File 22	17,288	81	9:37	1
File 23	21,518	95	11:58	1
File 24	73,138	371	40:40	1
File 25	42,335	197	23:33	1
Total	**874,019**	**4601**	**9:1:13**	**39**

12.4.2.2 Feature Extraction and Instances Filtering

Both visual and audio features are computed for each video shot via the multimodal content analysis process presented in Section 12.3.2 and are contained in each feature vector. Prefiltering techniques are applied to reduce the noise and outliers in the original data set, which generates the candidate shots pool for the data mining stage. The resulting pool size after prefiltering is 258.

12.4.3 Video Data Mining for Goal Shot Detection

These 258 candidate shots are randomly selected to serve as either the training data or the testing data. In our experiments, two experiments are designed and the rigorous fivefold cross-validation scheme is adopted to test the effectiveness of the proposed framework. In other words, the whole data set is randomly divided into a training data set and a testing data set in five times. Consequently, five different models are constructed and tested by the corresponding testing data set.

The first experiment is designed to testify the effects of the proposed prefiltering process in term of the accuracy and efficiency of the whole framework. Therefore, the classifications are conducted with and without the prefiltering step and the performance is summarized in Table 12.2. It is worth noting that the classification program [39] we adopted was written in Java and is running on a 3.06 GHz Pentium 4 personal computer with 1 GB RAM. As can be seen from this table, the classification results achieved with the integration of both NNG and the prefiltering process is quite promising, where by average the recall and precision values reach around 90% and 84%, respectively. In addition, the prefiltering step plays an important role in improving the overall classification performance. More specifically, the accuracy rates are more than double and the running time is reduced more than 95% by comparing to the ones without the prefiltering process.

The intention of the second experiment is to compare the performance of this proposed framework with our earlier approach where the C4.5 decision tree algorithm

Table 12.2. Classification performance of NNG on the testing data sets with and without the prefiltering step

Model	# of goals	Prefiltering	Identified	Missed	Misidentified	Recall (%)	Precision (%)	Running time (s)
1	15	Without	1	14	2	6.7	33.3	1.0
		With	12	3	2	80.0	85.7	0.05
2	15	Without	6	9	7	40.0	46.2	1.84
		With	14	1	3	93.3	82.4	0.05
3	13	Without	4	9	7	30.8	36.4	1.03
		With	12	1	3	92.3	80.0	0.06
4	13	Without	4	9	6	30.8	40.0	1.26
		With	12	1	2	92.3	85.7	0.05
5	12	Without	3	9	7	25.0	30.0	1.14
		With	11	1	2	91.7	84.6	0.06
Average		Without				26.7	37.2	1.25
		With				89.9	83.7	0.05

Table 12.3. Classification performance of C4.5 decision tree on the testing data sets with and without the prefiltering step

Model	# of goals	Prefiltering	Identified	Missed	Misidentified	Recall (%)	Precision (%)
1	15	Without	4	11	3	26.7	57.1
		With	12	3	3	80.0	80.0
2	15	Without	6	9	4	40.0	60.0
		With	14	1	3	93.3	82.4
3	13	Without	3	10	8	23.1	27.3
		With	12	1	3	92.3	80.0
4	13	Without	3	10	3	23.1	50.0
		With	12	1	3	89.9	80.2
5	12	Without	4	8	4	33.3	50.0
		With	11	1	3	91.7	78.6
Average		Without				29.2	48.9
		With				89.9	80.2

is adopted [10]. Here, the comparison with the PRISM scheme [8] is not performed because of its possible conflict problem discussed earlier. As an example, the PRISM approach induces 31 and 19 rules for *No* and *Yes* classes, respectively, in one training model, where 6 of them are conflicting rules. Therefore, the PRISM approach is excluded in our comparative experiment as many extra efforts are required to refine the classification results. For the purpose of comparison, the same attribute set is extracted and the same fivefold cross-validation scheme is adopted to test the decision tree based classification framework. In addition, the results are recorded by applying the classification algorithms with and without the prefiltering scheme, which has exactly the same criteria as the ones used in the first experiment. Table 12.3 shows the classification results, from which we have the following observations. First, the prefiltering process is critical for both NNG and decision tree classification algorithms in the sense that it greatly reduces the outliers and the noisy data. Second, without the facilitation of the prefiltering process, both the recall and precision rates achieved by the NNG algorithm are much lower than the ones obtained by the decision tree algorithm, which indicates that NNG is more sensitive to noise. However, after the prefiltering process, the accuracy of NNG is higher than that of the decision tree, which shows that NNG is more capable of dealing with the data inconsistency problem. It is worth mentioning that since the C4.5 program we use was coded in C language, the running time is not listed for comparison. Third, with the facilitation of the proposed multimodal analysis and the prefiltering process, we are able to achieve quite encouraging results by adopting various classification algorithms such as NNG and C4.5 decision tree, which fully demonstrates the effectiveness and generality of the proposed framework.

12.5 Conclusions

In this chapter, we have presented an effective multimedia data mining framework for the detection of soccer goal shots by using combined multimodal content analysis,

data prefiltering process, and generalized Instance-Based Learning (IBL) scheme. The proposed framework has many implications in video indexing and summarization, video database retrieval, semantic video browsing, etc. The proposed method allows effective and efficient mining of soccer goals by using a selective mixture of low-level features, middle-level features, and object-level features. By using the object-segmentation results (segmentation mask maps) produced during shot detection, some high-level features such as the grass ratio can be derived at a low cost, which are further used in the detection of the goal events. The proposed framework takes into account the various production styles of soccer videos by adopting an Instance-Based Learning scheme known for its focus on the local optimization for a particular query instance. In particular, a data prefiltering step is performed on the raw video features with the aid of domain knowledge, in order to alleviate the problem of noise sensitivity of IBL. The basic IBL is further generalized by using the so-called Nearest Neighbor with Generalization (NNG) for two main purposes, that is, to further reduce the adverse effect of noise data, and to expedite the classification process. Experiments have been conducted to examine the effect of the prefiltering process and the performance of the proposed multimedia data mining framework when compared with other popular model-based methods like decision trees. Our experiments over diverse video data from different sources have demonstrated that the proposed framework is highly effective in classifying the goal shots for soccer videos. Our future work will be conducted in the following three directions: (1) to extend the proposed framework to detect other soccer events (e.g., fouls, free kicks, etc.), (2) to identify more high-level semantic features that can be directly or indirectly derived from the existing object-level features, and (3) to investigate more effective methods for temporal data modeling and mining.

Acknowledgments

For Shu-Ching Chen, this research was supported in part by NSF EIA-0220562 and NSF HRD-0317692. For Mei-Ling Shyu, this research was supported in part by NSF ITR (Medium) IIS-0325260. For Chengcui Zhang, this research was supported in part by SBE-0245090 and the UAB ADVANCE program of the Office for the Advancement of Women in Science and Engineering.

References

1. Aha D. Tolerating noisy, irrelevant, and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies* 1992;36(2):267–287.
2. Assfalg J, Bertini M, Bimbo AD, Nunziati W, Pala P. Soccer highlights detection and recognition using HMMs. In: *Proceedings of IEEE International Conference on Multimedia and Expo*; 2002, pp. 825–828.
3. Babaguchi N, Kawai Y, Kitahashi T. Event based indexing of broadcasted sports video by intermodal collaboration. *IEEE Transactions on Multimedia* 2002;4(1):68–75.

4. Cendrowska J. PRISM: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies* 1987;27(4):349–370.
5. Chang SF. The holy grail of content-based media analysis. *IEEE Multimedia* 2002; 9:6–10.
6. Chen S-C, Shyu M-L, Zhang C, Kashyap RL. Video scene change detection method using unsupervised segmentation and object tracking. In: *Proceedings of IEEE International Conference on Multimedia and Expo*; 2001. pp. 57–60.
7. Chen S-C, Shyu M-L, Zhang C, Kashyap RL. Identifying overlapped objects for video indexing and modeling in multimedia database systems. *International Journal on Artificial Intelligence Tools* 2001;10(4):715–734.
8. Chen S-C, Shyu M-L, Zhang C, Luo L, Chen M. Detection of soccer goal shots using joint multimedia features and classification rules. In: *Proceedings of the Fourth International Workshop on Multimedia Data Mining (MDM/KDD2003)*, in conjunction with the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2003, pp. 36–44.
9. Chen S-C, Shyu M-L, Zhang C. Innovative shot boundary detection for video indexing. In: Sagarmay Deb, editor. *Video Data Management and Information Retrieval*. Idea Group Publishing, ISBN: 1-59140546-7; 2005, pp. 217–236.
10. Chen S-C, Shyu M-L, Zhang C, Chen M. A multimodal data mining framework for soccer goal detection based on decision tree logic. *International Journal of Computer Applications in Technology*, Special Issue on Data Mining Applications. In press.
11. Cheng HD, Jiang XH, Sun Y, Wang J. Color image segmentation: advances and prospects. *Pattern Recognition* 2001;34(12):2259–2281.
12. Dagtas S, Abdel-Mottaleb M. Extraction of TV highlights using multimedia features. In: *Proceedings of IEEE International Workshop on Multimedia Signal Processing*; 2001, pp. 91–96.
13. Deshpande U, Gupta A, Basu A. Performance enhancement of a contract net protocol based system through instance-based learning. *IEEE Transactions on System, Man, and Cybernetics, Part B* 2005;35(2):345–358.
14. Duan LY, Xu M, Yu XD, Tian Q. A unified framework for semantic shot classification in sports videos. In: *Proceedings of ACM Multimedia*; 2002, pp. 419–420.
15. Duan LY, Xu M, Chua TS, Tian Q, Xu CS. A mid-level representation framework for semantic sports video analysis. In: *Proceedings of ACM Multimedia*; 2003, pp. 33–44.
16. Ekin A. Sports video processing for description, summarization, and search [dissertation]. Department of Electrical and Computer Engineering, School of Engineering and Applied Sciences, University of Rochester; 2003.
17. Ekin A, Tekalp AM, Mehrotra R. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing* 2003;12(7):796–807.
18. Gong Y, Sin LT, Chuan CH, Zhang H, Sakauchi M. Automatic parsing of TV soccer programs. In: *Proceeding of IEEE Multimedia Computing and Systems*; 1995, pp. 167–174.
19. Hanjalic A. Shot-boundary detection: unraveled and resolved. *IEEE Transactions on Circuits and Systems for Video Technology* 2002;12:90–105.
20. Leonardi R, Migliorati P, Prandini M. Semantic indexing of soccer audio-visual sequences: a multimodal approach based on controlled Markov chains. *IEEE Transactions on Circuits and Systems for Video Technology* 2004;14(5):634–643.
21. Li B, Pan H, Sezan I. A general framework for sports video summarization with its application to soccer. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*; 2003, pp. 169–172.

22. Liu Z, Wang Y, Chen T. Audio feature extraction and analysis for scene segmentation and classification. *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology* 1998;20(1/2):61–80.
23. Martin B. Instance-based learning: nearest neighbor with generalization [thesis]. Department of Computer Science, University of Waikato, New Zealand; 1995.
24. Mitchell T. *Machine Learning*. New York: McGraw-Hill; 1997.
25. Rui Y, Gupta A, Acero A. Automatically extracting highlights for TV baseball programs. In: *Proceedings of ACM Multimedia*; 2000, pp. 105–115.
26. Tan YP, Saur DD, Kulkarni SR, Ramadge PJ. Rapid estimation of camera motion from compressed video with application to video annotation. *IEEE Transactions on Circuits and Systems for Video Technology* 2000;10(1):133–146.
27. Tovinkere V, Qian RJ. Detecting semantic events in soccer games: towards a complete solution. In: *Proceedings of IEEE International Conference on Multimedia and Expo*; 2001, pp. 1040–1043.
28. Wang Y, Liu Z, Huang J. Multimedia content analysis using both audio and visual clues. *Signal Processing Magazine* 2000;17:12 – 36.
29. Witten IH, Frank E. *Data Mining-Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco: Morgan Kaufmann Publishers; 1999.
30. Xie L, Chang SF, Divakaran A, Sun H. Structure analysis of soccer video with Hidden Markov Models. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*; 2002, pp. 13–17.
31. Xie L, Chang SF, Divakaran A, Sun H. Unsupervised discovery of multilevel statistical video structures using Hierarchical Hidden Markov Models. In: *Proceedings of the IEEE International Conference on Multimedia and Expo*; 2003, pp. 29–32.
32. Xie L, Xu P, Chang SF, Divakaran A, Sun H. Structure analysis of soccer video with domain knowledge and Hidden Markov Models. *Pattern Recognition Letters* 2003;24(15):767–775.
33. Xu M, Maddage NC, Xu CS, Kankanhalli M, Tian Q. Creating audio keywords for event detection in soccer video. In: *Proceedings of IEEE International Conference on Multimedia and Expo*; 2003, pp. 281–284.
34. Xu P, Xie L, Chang SF, Divakaran A, Vetro A, Sun H. Algorithms and system for segmentation and structure analysis in soccer video. In: *Proceedings of IEEE International Conference on Multimedia and Expo*; 2001, pp. 928–931.
35. Yow D, Yeo BL, Yeung M, Liu B. Analysis and presentation of soccer highlights from digital video. In: *Proceedings of 2nd Asian Conference on Computer Vision*; 1995, pp. 499–503.
36. Zhang C, Chen S-C, Shyu M-L. PixSO: A system for video shot detection. In: *Proceedings of the Fourth IEEE Pacific-Rim Conference on Multimedia*; 2003, pp. 1–5.
37. Zhu W, Toklu C, Liou SP. Automatic news video segmentation and categorization based on closed-captioned text. In: *Proceedings of IEEE International Conference on Multimedia and Expo*; 2001, pp. 1036–1039.
38. Zhu X, Wu X, Elmagarmid AK, Feng Z, Wu L. Video data mining: Semantic indexing and event detection from the association perspective. *IEEE Transactions on Knowledge and Data Engineering* 2005;17(5):665–677.
39. Weka 3: data mining software in Java. Available from: http://www.cs.waikato.ac.nz/ml/weka/.