# Algorithms and System for High-Level Structure Analysis and Event Detection in Soccer Video

Peng Xu, Shih-Fu Chang, Columbia University
Ajay Divakaran, Anthony Vetro, Huifang Sun, Mitsubishi Electric Advanced Television Lab

**Abstract**

In this report, we first present a general framework for video structure and content analysis. In this framework, frame-based low-level features are extracted. Each frame is represented by the values of features or labels converted from the features. So video sequence is transformed into multiple label sequences or real number sequences. Each of such sequence is associated with one of the extracted low-level feature. The feature sequences are analyzed together to extract high-level semantic features. Based on this framework, we describe an application system specifically for soccer video indexing and summarization. We use a distinctive feature to capture the high-level structure of the soccer video (e.g., play boundaries) and use a unique feature, grass orientation, together with camera motion to detect interesting events such as play strategy. The unique features of the system include compressed-domain feature extraction for real-time performance, use of domain specific features for detecting high-level events, and integration of multiple features for content understanding.

## Introduction

As digital video becomes more pervasive, efficient way of mining the information inside the video becomes necessary and important. Video itself contains huge amount of data and complexity that make the analysis very difficult. The first and very important analysis is to understand the structure of the video, which can provide the basis for further detailed analysis. Previous works have tried different approaches [1,2,3,4]. Video is first segmented into shots; key frames are extracted in each shots and then grouped into scenes. Scene transition graph and hierarchy tree are used to represent the structure [1, 4]. The problem with these approaches is the mismatch between the low-level shot information and the high-level scene information. It can only work when interesting content changes correspond to the shot changes. In many applications such as soccer videos, interesting events such “plays” cannot be defined by shot changes. Each play may contain multiple shots that have similar color distributions. Transitions between plays are hard to find by simple clustering of shot features.

And in many situations, when camera has a lot of motion, shot detection algorithm tends to have many false alarms. This kind of segmentation is from the low-level feature without considering the domain-specific syntax and content model of the video. Thus, it is difficult to bridge the gap between low-level features and high-level features based on shot-level segmentation. Moreover, too much information is lost during the shot segmentation process. We propose a framework in which at the first step, all the information of the low-level features are kept, and video sequences are represented by feature sequences that are represented by either symbolic labels or numerical numbers. Then according to the domain-specific syntax and content model, high level structure is

extracted from the video and multiple feature sequences are integrated to do event detection, statistical analysis and so on.

Videos in different domains have very different characteristics and structures. Domain knowledge can greatly facilitate the analysis process. For example, in sports videos, there are always a fixed number of cameras, views, camera control rules, and transition syntax imposed by the rules of the game (e.g., play by play in soccer, serve by serve in tennis, and inning by inning in baseball). In our framework, domain knowledge can be integrated from the initial steps of feature extraction to the latter steps of high-level semantic analysis.

There have been successful works in videos of news, baseball et al. [6, 7]. But there are few works in high-level structure analysis of soccer video. The challenge is that soccer game itself has relatively loose structure compared to other videos like news. Except the play-by-play structure, the content flow could be quite unpredictable and happen randomly. There are a lot of motion and view changes in the soccer. Based on our framework, we first focus on the play-by-play structure of the soccer game and identify unique features useful in the soccer domain. We use a unique feature to detect the play structure automatically. Then we use multiple features together to understand the activity in each play. All the features are extracted from the compressed domain using simple computation, and can be performed in the real time on the fly. Once the activity within each play is analyzed, we can browse and summarize the entire video sequence using the statistics of the constituent plays and types of activities in each play.

## Video content analysis framework

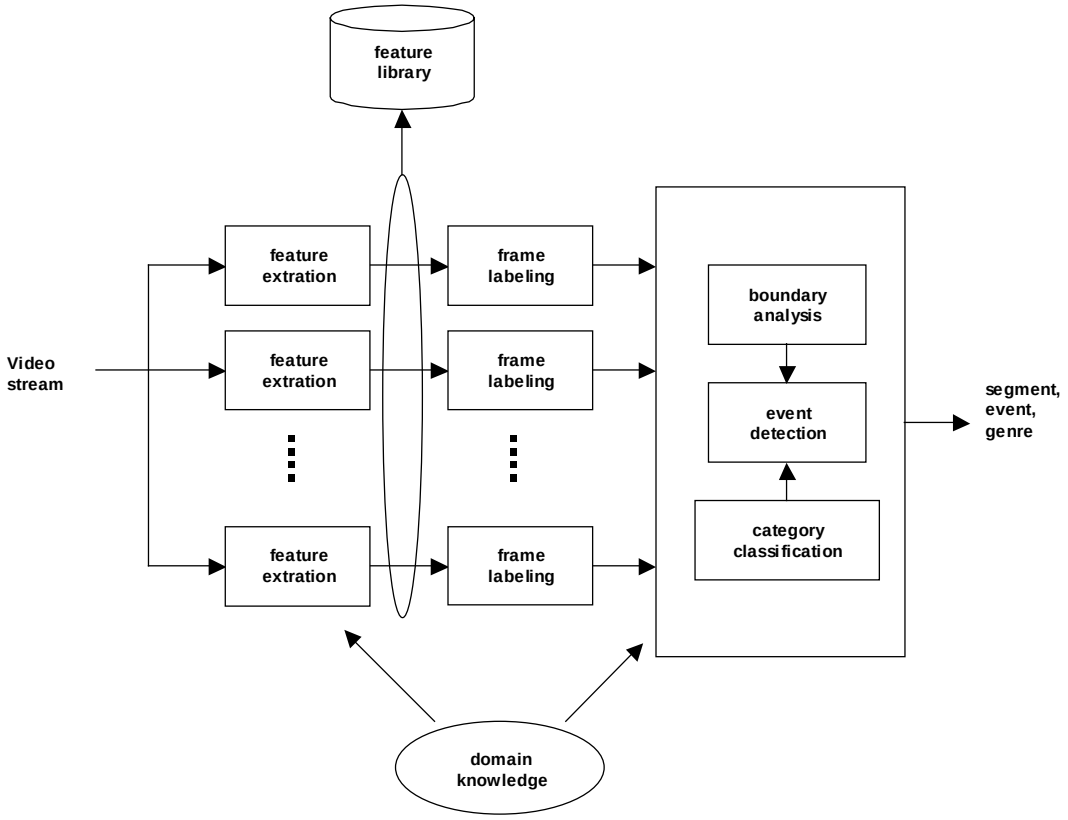


Fig 1. System architecture of the framework

In Figure 1, we describe the general video content analysis framework. Based on the characteristics of each domain, appropriate frame-based features are selected and extracted. If needed, features could be converted to labels that are the identifiers of the classes, or quantized discrete numbers, or just used with their real values. Then the multiple label sequences are integrated together for analyzing the structure of the video, detecting significant events, statistical analysis, or subject classification. Event boundaries from different label sequences may be synchronous or asynchronous. Synchronous boundaries can indicate significant content change, and asynchronous boundaries may indicate complementary information. Event transitions can be modeled using Hidden Markov Model or other machine learning techniques. In [9,10], multiple Hidden Markov Models are combined together to detect special events. In [11], domain-specific Hidden Markov Models have been trained to segment and recognize classes of videos like news, weather, and commercial.

## Soccer video analysis system

In this section, we describe application of the above framework in specific domain, i.e., soccer videos. We will first discuss observations of structure and syntax in this specific domain. We then present unique features and effective algorithms for extracting such features, detecting play boundaries, and analyzing the activities in each play.

**Structure definition**

Soccer game has a relatively loose structure compared to other types of videos such as news. It has two equal periods of 45 minutes. During the match, the ball is either in play or out of play. The ball is out of play when the ball is out of the field or the referee stops the play. At all other times, the ball is in play. According to this property, we define the structure of the soccer game as a sequence of plays and breaks. We define play as the period during which the ball is in play, and define break as the period during which the ball is out of play. Play starts when the ball is thrown in or kicked off. It can be a throw-in, free-kick, gate-kick, corner-kick or penalty-kick. It ends when the ball is out of the field or there is a goal, a fault or misconduct. Between plays there are breaks. During the break, the players prepare a kick, celebrate their goal, or get cautioned from referee and so on. For the viewer, activities within plays are more important than breaks, although exceptions may exist due to special needs sometimes.

**View and grass area**

Based on the observation, in the soccer video, there are three kinds of views, as shown in figure 2.

Figure 2a

Figure 2b

Figure 2c

Figure 2. Three kinds of views in soccer video
(2a: global, 2b: zoom-in, 2c: close-up)

The first one is the *global view*, which is shot from the top of the side. It gives the whole-picture view of the current play of the game. In this view, the grass field covers a large area and players appear small. The second is *zoom-in view*. The camera zooms to a small area in the field that is the focus of the game. It shows clearly the control of the ball by one player or fight between several players. In this view, there is still some area occupied by the grass. The last one is the *close-up view*. It shows players, coaches, referees, audience or so on. In this kind of view, there is very little area covered by grass area. It happens most time during break. We can do play-break segmentation based on these three different views. Close-up view typically corresponds to break. Zoom-in views can happen between plays or within a play, but these two cases are different. Between plays, they are typically replays, which replay the interesting parts that happen in the previous play; while during a play, zoom-in views typically give the on-going views of the game, although occasionally some replays occur inside a play. So one approach is to use view classification techniques combined with replay detection techniques such as those described in [8] to achieve the play-break segmentation.

Since the grass area is a very unique feature in distinguishing the three different views. We use it to do view classification and play-break segmentation. We only use I frames

and the computation is performed on the color thumbnail images of I frames from the DC coefficients without decompression. Grass area is identified by its unique green color, and the perceived green is best identified by its hue value, between 0.2 and 0.3. So we detect the grass area using the hue of the grass green. The hue of the pixel in the thumbnail image can be calculated by the color space transformation from YCbCr to RBG to HSV. Typically, the hue of grass green is quite consistent across different videos. But if we want to achieve a higher accuracy, this value can also be calibrated for different videos. According to the statistics, there are more than 80% frames in the soccer video belonging to the first two kinds of views containing grass area. A number of frames are selected randomly from a segment that is long enough to contain several plays. The cumulative histogram on hue is computed from the selected frames. The peak of the histogram between 0.2 and 0.3 gives the grass hue value, $h_g$. Then in the thumbnail image pixels having the hue in [$h_g$ - 0.05, $h_g$ + 0.05] will be considered as grass. The frames are classified according to the number of the grass pixels. The number of grass pixels is very distinctive for the three classes. Therefore, the threshold is easy to find by trails and can be used across different soccer videos. In this work, if more than half of the frame is covered by the grass, it is classified as global view, if less that 100 pixels out of 44*15 pixels are grass, it is classified as close-up view, other values are for zoom-in view. After classification, each I frame is labeled as 0, 1 and 2, corresponding to the three views. Further noise removal and label merge is processed. From our experience, if a view last less than 2 seconds in the soccer video, it doesn't give too much perceptual impression to viewers. We define segment as continuous label sequence with the same label. After classification, we merge segments with less than 4 labels to longer neighbor segments. Video is then segmented into play and break. Figure 3 shows the result of a five minutes soccer video from the MPEG-7 test videos. Figure 3a shows the number of detected grass pixels in each frame. Figure 3b shows the labels of views after classification. Figure 3c includes the recognition labels after noise reduction and neighbor merging processes are further applied. It can be seen that whole sequence is segmented into plays (labeled as 1 or 2) interlaced by breaks (labeled as 0).

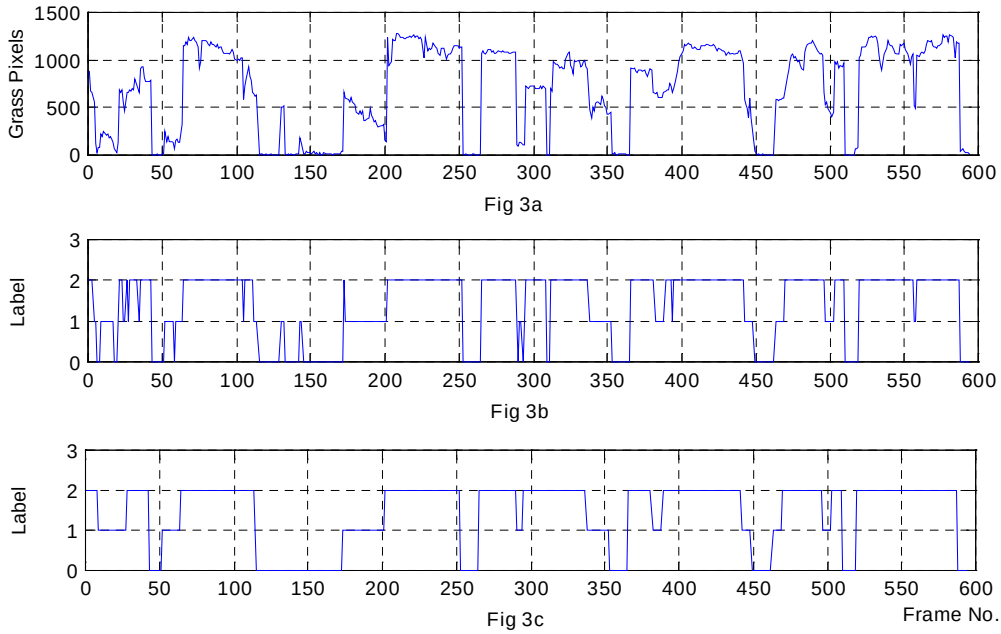


Figure 3. Grass area feature
(Fig 3a. grass area, Fig 3b. grass area label, 3c. noise removal and label merge)

After the segmentation, different analysis processes can be performed depending on whether it is play or break. During the break, the close-up gives the very close view of the player, coach or audiences. Pixel-domain techniques can be used to do more detailed analysis. For example, face recognition or character recognition can be done to identify the person in the view. So it can be inferred who is the center of the last play. During the play, replays can first be detected on the segment with zoom-in views, while in global views, other features, such as grass orientation, motion magnitude, camera motion can be used to analyze information about the side switch, fight intensity, etc. We will describe techniques for the latter type of processes in the following section.

**Grass orientation**

Usually the grass field has stripes. In the global view, the stripes are very clear and have different orientations depending on the view angle of the camera. The view angle of the camera depends on the location of the shooting in the field. So the location of the play can be inferred from the stripe orientation. When the camera shoots the right side of the field, the angle of the stripe is greater than 90 degree; when it shoots the middle of the field, the angle is around 90 degree; when it shoots the left side, the angle is less than 90 degree. The change of the angle indicates the change of the location of the activity. When there is side switch, there is angle change from greater than 90 degree to less than 90 degree or vice versa. The plays can be grouped according to the number of side switch.

We use the following algorithm to get the grass stripe orientation. First we use the Sobel gradient masks $S_x, S_y$ to get the gradient vectors $(g_{x,} g_y)$ from the intensity of the thumbnail image:

$$S_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad S_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Pixels that have larger magnitude of the gradient than the threshold are kept as the edge points. The gradient orientation for an edge point is calculated by:

$$\alpha = \arctan \frac{g_y}{g_x}$$

Then the grass orientation histogram is calculated by the grass orientations of all the edge points inside the grass area. The grass orientation at the edge point is:

$$\beta = 90^\circ - \alpha$$

$\beta$ is from 0 to 180 degree. The angle corresponding to the peak of the histogram is the angle for the orientation of the grass stripe. Although the orientation calculated by this method is not very accurate, but it is enough to show the approximate location of current play. Figure 4 shows one of the results:

Figure 4a. Thumbnail image

Figure 4b. Binary grass image

Figure 4c. Edge image

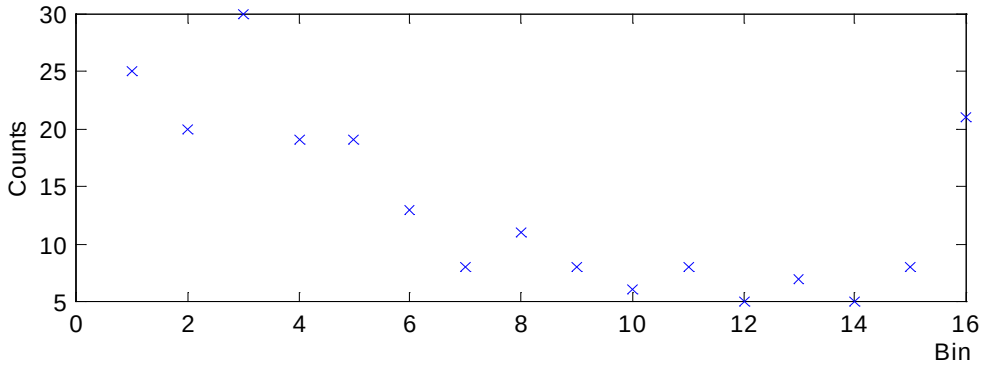


Figure 4d. Grass angle histogram

**Camera motion estimation**

In the global view, camera motion causes the global motion in the image frame. Because the camera tends to follow the movement of the ball, camera motion typically

indicates the motion of the ball. Players are relatively small in the global view so that there is not much local motion. In this case, motion vectors are good to use to estimate the camera motion. Because we only use I frame to get the grass area and orientation feature, we also reduce the computation here to the P frames immediately following I frames. Therefore for each I frame, we use motion vectors in the following P frame to estimate the camera motion occurring at the I frame. In the global view, camera motion is very simple. Most of the time, it has translation (pan/tilt). Occasionally it has zoom (zoom in/zoom out). People use different camera models for the estimation [5, 6]. Here a simple 3 parameters $\{k, p_x, p_y\}$ camera motion model is sufficient to estimate camera motion.

The camera motion can be considered as two-step operation. First camera translates to the new center. Second camera zooms at the center. The coordinates $(x_c', y_c')$ of the center are always the same, $(x_c', y_c') = (w/2, h/2)$, $w, h$ are the width and height of the frame. So

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = k \begin{pmatrix} x - p_x - x_c' \\ y - p_y - y_c' \end{pmatrix} + \begin{pmatrix} x_c' \\ y_c' \end{pmatrix}$$

According to the definition of motion vector,

$$\begin{pmatrix} u_{x'} \\ u_{y'} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Where $(x, y)$ is the coordinate of a point in the reference frame (I frame), $(x', y')$ is the corresponding location in the current frame (P frame). $k$ is the zooming factor, $(p_x, p_y)$ is the translation factor. Using the least squares method, one can find the best $\{k, p_x, p_y\}$ that minimizes the squared error between the estimated motion vectors and the actual motion vectors obtained from the MPEG stream. That is, to find $\{k, p_x, p_y\}$, so that $S(k, p_x, p_y)$ is minimized.

$$S(k, p_x, p_y) = \sum_{x'} \sum_{y'} [(u_{x'} - \hat{u}_{x'})^2 + (u_{y'} - \hat{u}_{y'})^2]$$

$(x', y')$ are the coordinates of all the macroblocks. $(u_{x'}, u_{y'})$ is the estimated motion vector for macroblock at $(x', y')$, $(\hat{u}_{x'}, \hat{u}_{y'})$ is the corresponding motion vector from the stream. After the estimation, real motion vectors that have large distance from the estimated motion vectors are filtered out. Estimation is repeated on the survived motion vectors. The estimation is iterated several times to refine the accuracy. At the last iteration, the average motion vector magnitude can be computed from the motion vectors used for the final estimation. Because the estimation is performed on the motion vectors of P frame that follows I frame, and all the I frames are from the segment of the play which has global view, most of the motion vectors in the P frame are consistent with the global motion.

After the camera motion estimate, in each play, accumulated $p_x$ can be computed at each I frame starting from the beginning of the play. It represents the total camera motion along horizontal direction, and reflects the position change of the ball in the field along horizontal direction. This information together with orientation feature, can give the more accurate information about the activity within each play. Also for each play, the average motion vector magnitude gives the level of the motion activity in this play, which indicates the intensity of the game in the play. Figure 5 shows the grass orientation and cumulative pan along x of one play in the video. It shows three side switches.

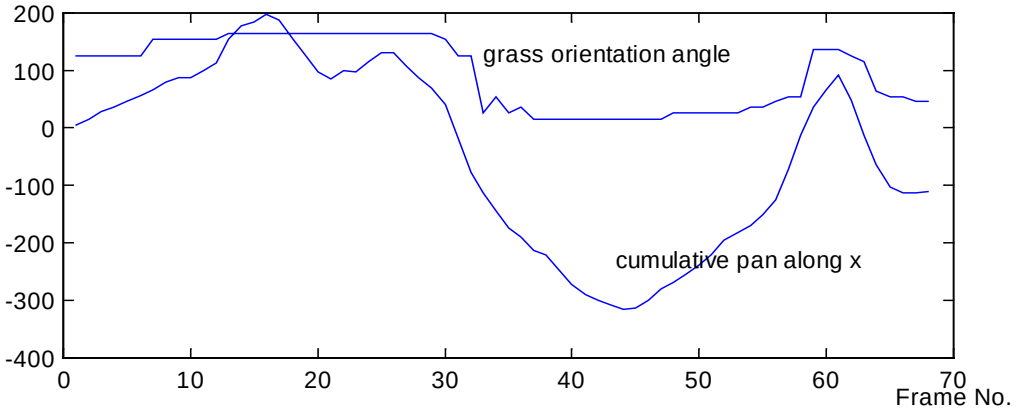


Fig 5. Grass orientation and cumulative pan along x in one play

**System architecture**

Based on our work, we can build an automatic structure analysis and summary system that can be used to browse and analyze soccer video. The video stream is fed into the feature extraction module. Video structure is represented by the cycle of play and break. In different part of game, different features are extracted and combined to get high-level features. In each play, feature of grass orientation can give information about side switches of the play, motion magnitude can give the fight intensity of the play, and camera motion can help to track the ball in the play. In the break, face recognition or character recognition can be performed.

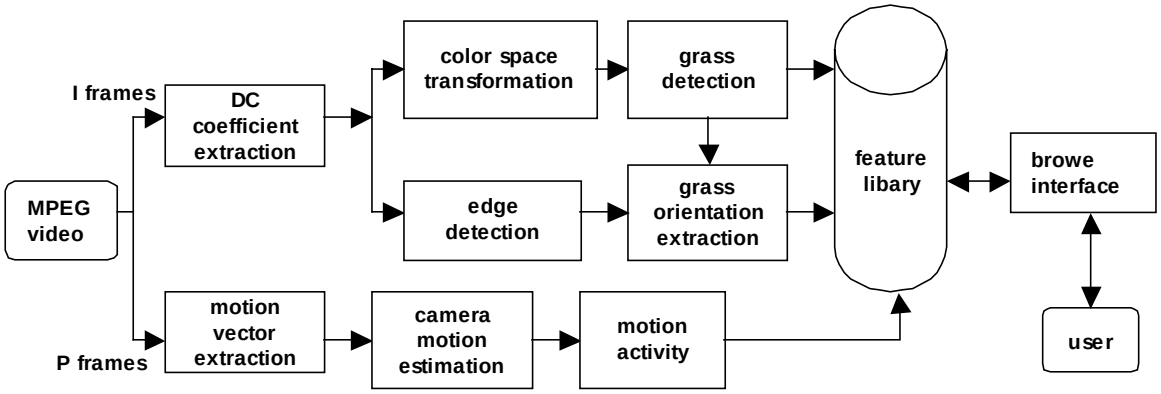


Figure 6. Soccer video analysis system architecture

**Conclusion**

We presented a general framework for video structure and high-level content analysis. Based on the framework, we build a system for soccer video indexing, summarization and browsing. Soccer video is a very challenging domain because of the characteristics of the game. Our unique approaches include

- Compressed-domain feature extraction and content analysis
- Domain-specific structure analysis and event detection using unique features
- Integration of multiple features
- Integration of segmentation, browsing, and event categorization

The current results indicated very promising potential. There are a few areas we could extend and achieve improved results, including comprehensive testing over more videos, improved algorithms for computing more accurate information about grass orientation, and a systematic way for identifying and integrating unique features in different domains.

Reference:

[1] M. M. Yeung, B. L. Yeo, W. Wolf, and B. Liu. Video Browsing using Clustering and Scene Transitions on Compressed Sequences. In Multimedia Computing and Networking 1995, Vol. SPIE 2417, pp. 399-413, Feb. 1995.

[2] M. J. Yeung and B. L. Yeo. Time-constrained Clustering for Segmentation of Video into Story Units. In ICPR, Vol. C. pp. 375-380 Aug. 1996.

[3] D. Zhong, H. J. Zhang and S. F. Chang. Clustering Methods for Video Browsing and Annotation. In SPIE Conference on Storage and Retrieval for Image and Video Databases, Vol. 2670, Feb. 1996.

[4] J. Y. Chen, C. Taskiran, E. J. Delp and C. A. Bouman. ViBE: A New Paradigm for Video Database Browsing and Search. In *Proc*. IEEE Workshop on Content-Based Access of Image and Video Databases, 1998.

[5] Y. P. Tan, S. R. Kulkarni and P. J. Ramadge. A New Method for Camera Motion Parameter Estimation. In Proc. IEEE International Conference on Image Processing, Vol. 2, pp. 722-726, 1995.
[6] Y. P. Tan, D. D. Saur, S. R. Kulkarni and P. J. Ramadge. Rapid estimation of camera motion from compressed video with application to video annotation. To appear in IEEE Trans. on Circuits and Systems for Video Technology, 1999.
[7] H. J. Zhang, S. Y. Tan, S. W. Smoliar and Y. H. Gong. Automatic Parsing and Indexing of News Video. In Multimedia Systems, Vol. 2, pp. 256-266, 1995.
[8] V. Kobla, D. DeMenthon and D. Doermann. Detection of slow-motion replay sequences for identifying sports videos. In Proc. IEEE Workshop on Multimedia Signal Processing, 1999.
[9] T. T. Kristjansson, B. J. Frey and T. S. Huang. Event-Coupled Hidden Markov Models. In ICME, Aug. 2000.
[10] M. R. Naphade, T. Kristjansson, B. Frey and T. S. Huang. Probabilistic multimedia Objects (Multijects): A Novel Approach to Video Indexing and Retrieval in Multimedia Systems. In ICIP, 1998.
[11] J. C. Huang, Z. Liu and Y. Wang. Joint Video Scene Segmentation and Classification Based on Hidden Markov Model. In ICME, Aug. 2000.