International Journal of Image and Graphics, Vol. 1, No. 3 (2001) 445–468

RECENT ADVANCES IN CONTENT-BASED VIDEO ANALYSIS

CHONG-WAH NGO* and TING-CHUEN PONG†

Department of Computer Science, Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong

HONG-JIANG ZHANG‡

Microsoft Research China 5/F, Beijing Sigma Center, Haidian District, Beijing, PRC

In this paper, we present major issues in video parsing, abstraction, retrieval and semantic analysis. We discuss the success, the difficulties and the expectations in these areas. In addition, we identify important opened problems that can lead to more sophisticated ways of video content analysis.

For video parsing, we discuss topics in video partitioning, motion characterization and object segmentation. The success in video parsing, in general, will have a great impact on video representation and retrieval. We present three levels of abstracting video content by scene, keyframe and key object representations. These representation schemes in overall serve as a good start for video retrieval. We then describe visual features, in particular motion, and similarity measures adopted for retrieval. Next, we discuss the recent computational approaches in bridging the semantic gap for video content understanding.

Keywords: Video Parsing; Video Abstraction; Video Retrieval; Semantic Analysis.

1. Introduction

In the past few decades, computer vision community has developed theories and techniques for acquiring, manipulating, transmitting and storing video data. Nevertheless, the methodology for searching and representing visual information is still in its infancy. Since the last decade, the problems of content-based video and image retrieval have been actively researched and formally addressed by researchers from various communities.[1] Numerous attempts have been made to represent and describe the visual world — a world without language, or more formally, a world with inherent meaning, far more complex than words. Traditional work on textual annotation and retrieval of video and image information are being questioned and challenged.

*E-mail: cwngo@cs.ust.hk
†E-mail: tcpong@cs.ust.hk
‡E-mail: hjzhang@microsoft.com

To date, with the vast amount of available video sources, tools of extracting, representing and understanding video structures for content browsing and retrieval have become of timely importance and interest. On one hand, we need a machine that can automatically analyze and summarize the content of videos. Hence, for instance, instead of browsing through a one hour video from beginning to end, we could review the video in few minutes without overlooking important content. The tedious task of summarizing video content done in a traditional way by film producers, however, emerges as a difficult problem to overcome, even with the advance of machine learning and artificial intelligence techniques. The initial goal, therefore, is not to produce a system that can automatically and semantically imitate what film producers do, but a system that is capable of, at least semi-automatically, analyzing video content in a syntactical and intelligent way.

On the other hand, we need reliable methods for extracting salient features such as color, motion, texture, shape and face descriptors to index and retrieve visual information. These features are expected to describe the similarity or dissimilarity of visual data. There are two main difficulties with this research direction: (1) segmentation, which is always imperfect is required prior to feature extraction; (2) how does one match the similarity measure of machine to human, where perception changes from person to person? In this paper, we will mainly review techniques for the decomposition of a video into smaller units such as scenes, shots, keyframes and objects, and the indexing of visual features from these units for abstracting, retrieving and understanding video content.

Figure 1 depicts the structural content of a typical video. Generally speaking, a video is composed of scenes, which may convey distinct story units. Each scene is formed by one or more shots that are taken place at the same site, and each shot is further composed of frames with smooth and continuous motions. The goal of content-based video analysis, in a converse way, is to structure the content of videos in a bottom-up manner, as illustrated in Fig. 1, while abstracting the main content from frames. To accomplish this goal, we need techniques in parsing videos, extracting features and organizing video content. Figure 2 illustrates the major flow of these processes.

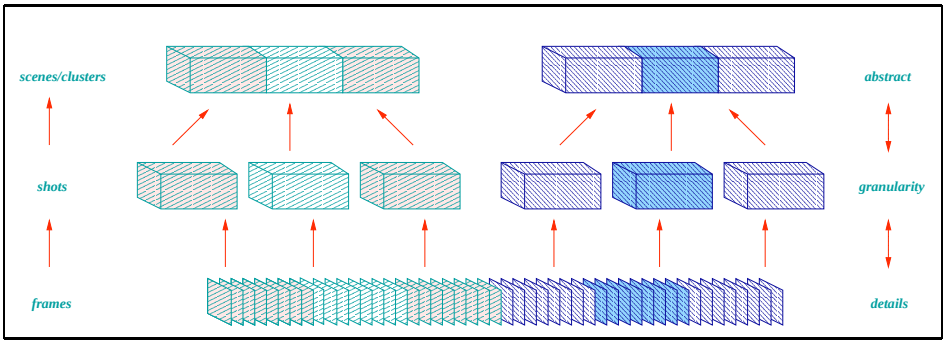


Fig. 1. Video structure.

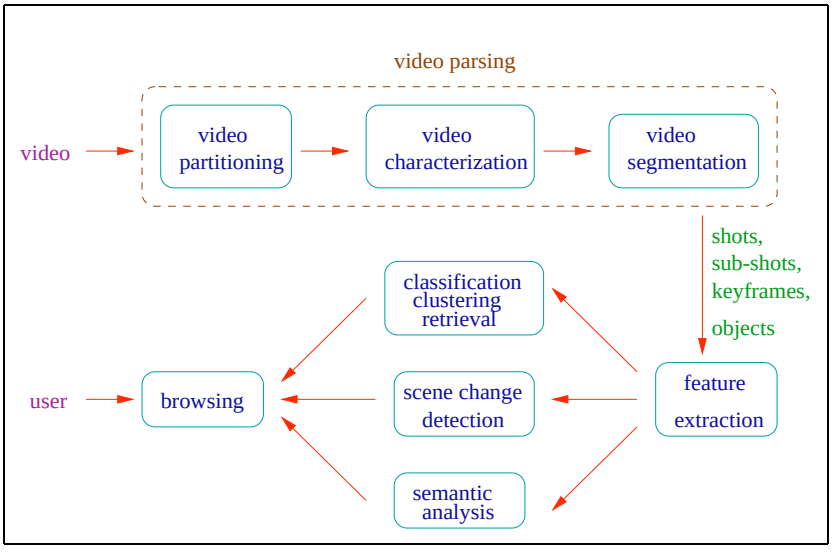


Fig. 2. Extracting, representing and utilizing video structure.

Figure 2 illustrates a general scheme for content-based video analysis. A video is first partitioned into shots (video partitioning). A shot can be further divided into finer units called sub-shots by characterizing the underlying camera and object motion (motion characterization). Based on the annotated motions, either the objects in a shot are segmented and tracked over frames (video segmentation), or keyframes are selected and constructed, to represent the content of a video. The visual features of these elementary units (shots, sub-shots, keyframes and objects) are further extracted for indexing. Subsequently, through the similarity measure of visual features, shots are semantically or intelligently organized for various purposes such as scene change detection, clustering and retrieval. Finally, a user can query and browse video content in a convenient and easy way.

Developing the components in such a system requires knowledge of image and video processing, computer vision, information retrieval and pattern recognition. Researchers from various communities have participated actively in the content-based video analysis since last decade. Besides inventing new algorithms for analyzing and processing videos, the following considerations are also taken into account:

- *Compressed versus uncompressed.* Processing digital videos requires intense computational time and huge memory space. In order to save both time and space, techniques which manipulate videos directly in compressed domain have become a common practice.[2–7]
- *Automatic versus semi-automatic.* A fully automatic and practical system requires high-level knowledge. Since most works are mainly reliant on low-level visual information, filling the gap between low-level and high-level knowledge is a nontrivial issue. As a result, most systems require manual adjustment and

interruption. One typical example is the relevancy feedback mechanism,[8] which requires feedback from users in order to fine tune system parameters.

- *Mixed media.* To analyze semantically the real content of videos, audio and textual information are useful cues that can be used along with visual information. These works have been frequently considered in recent literature.[9–12]

In this paper, we present an overview of the current state-of-the-art in content-based video analysis, in particular, we focus our attention to the most recent published papers from year 1999 to 2000. Other review papers can be found in Refs. 2, 3, 13–19, which supplement the content of this paper. In the meantime, the latest related special issues include Refs. 20–22 will be discussed, while the recent work in Ref. 23 will be used as examples to further illustrate the ideas of some important topics. The paper is organized as follows. Section 2 presents a review of video parsing techniques. These include the partitioning, characterization and segmentation of video content. Section 3 discusses various schemes for video abstraction, which include keyframe, key object and scene representation. Section 4 describes the problems encountered and the approaches adopted in video retrieval. Section 5 gives an overview of the video semantic analysis. Finally, Sec. 6 concludes this paper.

2. Video Parsing

Video Parsing serves as the preliminary step to structure the content of videos. Elementary units such as shots, motion annotated sub-shots and video objects are obtained through this step. The success in this step can have great impact on video abstraction and retrieval. To date, most works on video parsing are devoted to video partitioning. Although motion characterization and video segmentation are as well important, researchers would like to skip these components since good results in term of speed and quality cannot be easily acquired. In fact, a success video application without lying on these steps seems more attractive. In any case, it is worth to notice that since the concern in video parsing is qualitative rather than quantitative information, the related algorithms that have been developed in the computer vision community should be simplified before applying directly for content analysis.

2.1. *Video partitioning*

Based on the transitional properties of video edits, there are three major types of camera breaks: *cut*, *wipe* and *dissolve*. A camera cut is an instantaneous change from one shot to another; a wipe is a moving transition of a frame (or a pattern) across the screen that enables one shot to gradually replace another; a dissolve superimposes two shots, where one shot gradually appears while the other fades out slowly.[a] Wipes and dissolves involve gradual transitions with no drastic changes

[a]Fade-in and fade-out can be considered as special cases of dissolve by replacing one of the shots as a constant image sequence (black image sequence for instance).

between two consecutive frames, and hence, are relatively difficult to identify. While cuts can be identified by comparing two adjacent frames, wipes and dissolves require the investigation of frames along a larger temporal scale.

Most methods on video partitioning utilize color histogram,[24] edge,[25] motion[26] and statistical hints[4,27] to identify camera breaks. To date, the identification of cuts has been somehow successfully tackled, detection of gradual transition, nevertheless, remains a hard problem. Several works[28–31] have been conducted to evaluate the performance of various algorithms. Cut detection algorithms, in general, perform reliably except in the cases such as illumination changes and foreground moving objects of large size or fast speed. Among these algorithms, color histogram based approaches give superior performance.[29] Gradual transition detector, on the other hand, can handle only simple examples. Besides the incapabilities in handling complicated transitions such as fancy wipe patterns, most approaches fail when motion is coupled with gradual transition. The number of false alarms can easily exceed the correct detections when processing real videos.[31]

2.1.1. *Gradual transition detection*

Most recent published works are dedicated to handle the wipe and dissolve detections.[26,32–41] Among them, many algorithms[32,34,35,37,38,40] are proposed based on the video production model. In brief, the model describes a gradual transition from shot g to shot h during a period (t_1, t_2) as:

$$f(x,y,t) = (1-\alpha(x,y,t))g(x,y,t) + \alpha(x,y,t)h(x,y,t)\,, \quad t_1 < t < t_2\,, \tag{1}$$

where $f(x,y,t)$ denote the value of a pixel (x,y) in sequence f at time t. The transition function $\alpha(x,y,t)$ characterizes, linearly or nonlinearly, how $f(x,y,t)$ is changed over time as a result of mixing $g(x,y,t)$ and $h(x,y,t)$. Typically, $\alpha(x,y,t) \in \{0,1\}$ for a wipe pattern, and $0 < \alpha(x,y,t) < 1$ for a dissolve, with the condition $\alpha(x,y,t) \leq \alpha(x,y,t+1)$. Since Eq. (1) is irreversible, apparently, detecting and classifying the type of gradual transitions (including various wipe patterns) is a hard problem.

To simplify the problem of detection, dissolve specifically, some algorithms[32,34,35,37,38,40] take $\alpha(x,y,t) = \alpha(t)$, $g(x,y,t) = g(x,y)$ and $f(x,y,t) = f(x,y)$. As a consequence, Eq. (1) becomes

$$f(t) = (1-\alpha(t))G + \alpha(t)H\,, \quad t_1 < t < t_2\,, \tag{2}$$

and the shots g and h contain only static frames G and H. By taking the frame difference $f(t) - f(t+k)$, we have

$$f(t) - f(t+k) = \rho(t,k)\{f(t-k) - f(t)\}\,, \quad t_1 + k < t < t_2 - k\,, \tag{3}$$

where

$$\rho(t,k) = \frac{\alpha(t+k) - \alpha(t)}{\alpha(t) - \alpha(t-k)} > 1\,.$$

Notice that the correlation between $f(t) - f(t+k)$ and $f(t-k) - f(t)$ is 1. Furthermore, plateau effect[6] will be exhibited if $k > t_2 - t_1 + 1$. These two properties are exploited in Refs. 6 and 35 and shown to give reasonably good results.

On the other hand, some approaches[34,38,40] further assume $\alpha(t)$ as a linear function, $\alpha(t) = (t - t_1)/(t_2 - t_1)$ for instance, as a result, Eq. (2) can be re-formulated in term of variance as:

$$\sigma_f(t) = (\sigma_G + \sigma_H)\alpha^2(t) - 2\sigma_G\alpha(t) + \sigma_G\,, \tag{4}$$

where $\sigma_f(t)$, σ_G and σ_H are the variances of $f(t)$, G and H, respectively. Since $\sigma_f(t)$ is a concave upward parabola curves, a simple algorithm for dissolve detection is by locating parabola curves. Nevertheless, this algorithm suffers from two fundamental problems: curves are not easily detected for dissolves involve only several frames; the valley of parabola curve will be too shallow for detection if the values of σ_G and σ_H are small. The approach in Ref. 34 further exploits the linearity property by detecting

$$\frac{d^2\sigma_f(t)}{dt^2} = \frac{2(\sigma_G + \sigma_H)}{(t_2 - t_1)^2} \quad \text{and} \quad \frac{df(t)}{dt} = \frac{(H - G)}{t_2 - t_1}\,,$$

which are constant values during dissolve period. However, this approach is vulnerable to noise and not suitable for compressed domain processing.

In summary, most gradual transition detectors based on the video production model can only achieve limited success. Besides the linearity assumption, the major obstacle is due to the assumption that the two superimposed shots g and h are static sequences. This deficiency, nevertheless, can be diminished by compensating the dominant motion between two adjacent frames prior to the detection of gradual transitions.[30,35]

There are also few interesting works on the gradual transition detection that do not based on the video production model. Bouthemy *et al.*[26] proposed a unified framework to detect cuts, wipes and dissolves based on a 2D parametric motion model. Cuts can be detected by observing the size of dominant motion support layers between two adjacent frames. A significant drop of the size in a frame clearly marks the presence of a cut. Gradual transitions, however, cannot be easily detected in this way, as a result, Hinkley test is employed to detect wipes and dissolves by investigating the size of support layer over a period of time. Although the proposed approach is computationally intensive and cannot be targeted for real-time applications, the estimated dominant motion is a useful by-product that is readily to be utilized for motion characterization. In addition, Jun *et al.*[42] proposed a fast and efficient dissolve detector by utilizing the macroblock information of MPEG videos. The authors empirically found the typical patterns of macroblocks during dissolves. By investigating the ratio of forward macroblocks in B-frames, and the spatial distribution of forward and backward macroblocks, promising results are obtained whentesting the proposed method on news and sports videos. Kim[36] and Ngo[40,41]

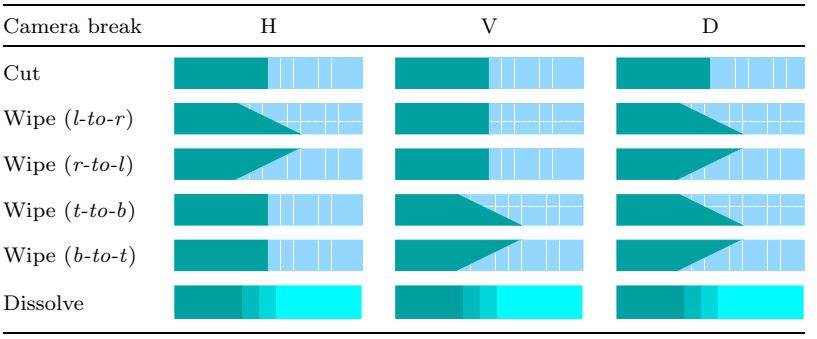


Fig. 3. The spatio-temporal slice patterns generated by various types of camera breaks: *l-to-r* (left-to-right); *r-to-l* (right-to-left); *t-to-b* (top to bottom); *b-to-t* (bottom-to-top).

proposed novel detection algorithms based on the analysis of spatio-temporal slices. These slices are a set of 2D images extracted along the temporal dimension of image sequences. In Refs. 40 and 41, Ngo *et al.* presented various transitional patterns in slices generated by cut, wipe and dissolves. One typical example is given in Fig. 3. These transitions are detected by a proposed spatio-temporal energy model, which takes the color and texture patterns of slices into account.

2.1.2. *Cut detection*

Besides wipe and dissolve detections, several works have also been done to improve previous algorithms on cut detection. For instance, Vasconcelos and Lippman[27] proposed an adaptive threshold setting method based on the Bayesian formulation by modeling the shot duration and shot activities. Better experimental results was yielded compared to the fixed threshold strategy commonly practiced for cut detection. Lee *et al.*[43] proposed a method to speed up the processing time of feature based cut detection algorithms e.g., Ref. 4 by extracting edge features directly from the DCT coefficients of MPEG videos. Experimental results show that the speed can be comparable to that of the algorithm presented by Yeo and Liu,[6] who employed the frame differencing of DC sequence to detect cuts. Nang *et al.*[39] proposed a method that takes the relationship of macroblocks among two adjacent B-frames into account for cut detection. The relationship are hand-crafted in two decision tables for measuring the similarity between two frames. Their algorithm is comparatively robust to those cut detectors that utilize the ratios on the numbers of forward and backward motion vectors in frames. In addition, a method similar to step-variable[44] was proposed by Hua *et al.*[45] to improve the processing efficiency by temporally sub-sampling of video frames. Intuitively, instead of sequential search for camera cuts from one frame to another, an adaptive[45] or binary search[44,46] can be carried out to locate cuts while skipping unnecessary frame comparisons. Such strategy can push the speed of detection about 16 times faster than sequential search.

2.2. *Motion characterization*

Characterization of camera and object motion plays a critical role in content-based video indexing. It is an essential step towards creating a compact video representation automatically. We can imagine a camera as a narrative eye, it describes by showing: a camera panning imitating an eye movement to either track an object or to examine a wider view of a scene; freeze frames give the impression that an image should be remembered; closeups indicate the intensity of an impression. In order to capture these impressions in a compact representation, a panning sequence could be represented by a mosaic image; a static sequence is well represented by one frame; a zoom sequence is well described by the frames before and after zoom, while the focus of a tracking sequence should be the targeted objects. By motion characterization, a shot can be further temporally segmented into sub-shots with each sub-shot consists of a coherent camera motion.[26,47] In this way, keyframe extraction and motion retrieval can be directly and easily conducted on the basis of these motion annotated sub-shots.

MPEG motion vectors,[24] spatio-temporal patterns in slices[47] and 2D parametric motion model[26] are commonly utilized for motion characterization. In ideal situations, for instance, the absence of large homogeneous regions and moving foreground objects, the annotation of camera motion can be done effectively through the dominant motion estimation. In principle, to tackle the effect of motion parallax, 3D parametric motion model, instead of 2D model, should be applied for analysis.[48] Unlike camera motion, which is rigid, the analysis of object motion is far more difficult. For most approaches, foreground objects are segmented after compensating the camera (dominant) motion in frames. The target objects, either in rigid or deformable forms, are tracked while the corresponding motions are modeled and indexed from time to time.

2.3. *Video segmentation*

Video segmentation is normally viewed as the first step towards high-level (semantic) video representation and retrieval. For instance, suppose background and foreground objects can always be segmented, team classification in most sport videos can be done effectively by utilizing the color information of foreground objects, while scene change detection can be conducted by comparing the similarity of background regions in shots.[23] However, video segmentation is always a hard problem and perfect segmentation is not easily attainable in an automatic fashion. In most cases, user interaction is required to guide the segmentation process. For example, in VideoQ system,[49] the initial sketch of a segmented object needs to be specified manually before the tracking, indexing and retrieval of video objects.

Automatic video segmentation usually carried out in a hierarchical manner.[50,51] Starting from an over segmentation of regions, regions are merged hierarchically to form pseudo-objects based on the hand crafted heuristics or visual similarity measures. For instance, Nguyen *et al.*[51] proposed a motion similarity measure to

merge the over segmented regions, while Chang *et al.*[50] utilized edge information to merge smaller regions with coherent color. In addition, segmentation bases solely on the motion feature has also been applied.[23,48,52] The basic assumption is that each motion can correspond to one moving object and hence objects can be obtained directly through motion segmentation. Although deformable and articulate objects cannot be handled under this assumption, the background objects that are induced by camera motion and the foreground objects that exhibit different motion from the camera can always be acquired.

3. Video Abstraction

Conventional video browsing are mostly done by forwarding the content sequentially or linearly from beginning to end. Video abstraction, in contrast, provides a nonlinear way of browsing video content. This is typically done by identifying few representative frames to summarize a shot (keyframe representation), detecting the abrupt changes of a moving object to summarize the life span of an object (key object representation), and grouping similar shots together to summarize the narrative flow of video content (scene representation).

3.1. *Keyframe representation*

Keyframe representation is a simple yet effective way of summarizing the content of videos for browsing and retrieval. The main challenge lies in how to represent videos with keyframes in a perceptually compact and meaningful way. Basically, there are two different approaches to achieve this purpose, namely keyframe selection and keyframe formation. Keyframe selection is the process of selecting frames directly from shots to represent the video content. Keyframe formation, on the other hand, is the process of forming new images given image sequences. The former approach has attracted a lot of attention from researchers[53–61] since practically it is efficient, and statistically it is highly possible to minimize the redundant information existing in shots. The latter approach, although more intuitive, requires effort in motion annotation and segmentation, which are regarded as difficult problems. For effective browsing and retrieval, the selected and constructed keyframes in a video can be further organized as flat partition,[59] hierarchical tree,[53,55] or video poster.[60,61]

Recent advances in keyframe selection utilize graph theory,[53] curve splitting,[55,62] clustering[56,57,59] and singular value decomposition.[54,58] In general, these approaches represent a frame as a point in the high dimensional feature space. The idea is to select a subset of points that can either cover the remaining points within a feature distance or to capture the significant changes of content in a shot. Chang *et al.*[53] viewed a shot as a proximity graph and each frame is a vertex in the graph. The problem of keyframe selection can be equivalent to the vertex cover problem. Specifically, we want to find a minimal cover of vertices that minimizes the total feature distance between the vertices and their neigboring points. Nevertheless, this problem is NP-complete. Thus, Chang *et al.* proposed sub-optimal solutions based on

the greedy approach and rate-distortion performance. Unlike Chang, Dementhon[55] and Zhao[62] turned the keyframe selection as the curve splitting problem. A shot is viewed as the feature trajectory or curve of high dimensional points. The task is to detect the junctions or break points of a curve as keyframes. Based on this idea, Dementhon further proposed a hierarchical view of keyframes by recursively identifying curve junctions while simplifying a curve from fine to coarse levels.

A more traditional way of keyframe selection is through the clustering of video frames. Hanjalic and Zhang[59] employed a partitional clustering algorithm with cluster-validity analysis to select the optimal numbers of clusters for shots. The resulting clusters are optimal in term of inter and intra cluster distance measures. The frame, which is the closest to a cluster centroid is selected as keyframe. While clustering is efficient, the curse of dimensionality need to be handled carefully. In principle, the number of frames to be clustered have to be several times larger than the dimensions of feature space. To be effective, feature dimensionality reduction can be done prior to clustering. For instance, Chang[54] and Gong[58] encoded the shot information in a matrix A and decomposed the matrix with singular value decomposition (SVD), $A = U\Sigma V^T$. Each column in matrix A is the feature vector of a frame. The left singular matrix U consists of eigen images of a shot that can facilitate video retrieval, while the right singular matrix V quantitatively measures the content richness of each frame. Since frames in a shot are highly correlated, normally only few principle components are selected. This not only reduces the feature space dimension but also suppresses undesired noise. As a result, clustering can be performed more effectively and efficiently in the projected feature space. Despite these advantages, SVD is computationally intensive.

To a certain extent, compact video representation requires not only keyframe selection, but also keyframe formation. For instance, a camera panning or tilting, intuitively, should be summarized by a panoramic image,[48] instead of selecting few frames to describe the scene. Ngo *et al.*[23,63] proposed a motion-based keyframe representation based on the analysis of spatio-temporal slices. The idea is to represent shots compactly and adaptively through motion characterization. Figure 4 illustrates the idea of keyframe representation, shows together with the extracted spatio-temporal slices for motion analysis. In brief, one frame is arbitrarily selected to summarize the content of a sequence with static motion, a panoramic image is formed for a sequence with camera panning or tilting, while two frames before and after zoom are picked for a sequence with camera zoom. For multiple motion, background image is constructed while foreground object is tracked over time. Such representation is found to be more useful for retrieval, browsing and scene change detection. Nevertheless, the success of this scheme is highly dependent on the robustness of motion analysis algorithm.

3.2. *Key object representation*

While keyframes provide a summary of video content, key objects give a visual summary of the lifespan of an individual object in a video. Basically, key objects

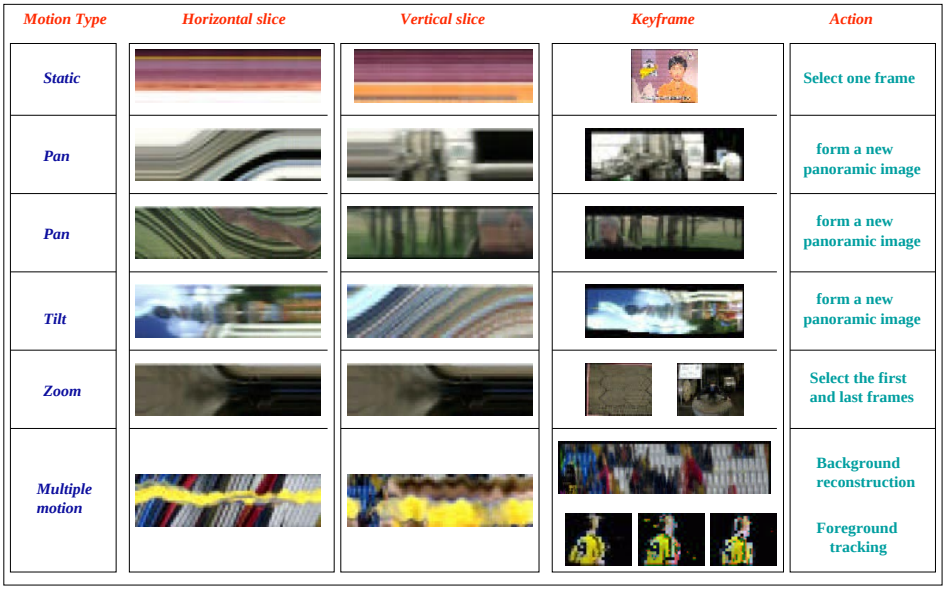


Fig. 4. Motion-based keyframe representation.

are the snapshots of a tracked object, where motion discontinuities and shape variation occur. In other words, each key object describes the temporal segment of an object lifespan as a sequence with coherent object motion and shape information. These key objects are selected mainly to support object-based video queries. Zhang *et al.*[64] first proposed a key object based framework for video representation and retrieval. The motion segmentation algorithm described by Wang and Adelson[65] was employed to automatically segment key objects. Recently, due to the emergence of MPEG-4, there have been few works[66–68] conducted by making use of the video object planes (VOP) information on MPEG-4 platform for indexing. In general, shape information can be easily extracted from VOP while shape similarity can be measured by matching contour points through Hausdorff distance.[66,67] In Ref. 67, each video object is represented by a 2D adaptive triangular mesh. This mesh representation models the motion changes of an object, in addition, provides object animation capability for browsing.

3.3. *Scene representation*

A scene is composed of a series of consecutive shots that are coherent from the narrative point of view. These shots are either shoot in the same place or they share similar thematic content. The major problem in scene representation is how to group shots that are narratively similar as a scene, or equivalently, how to identify scene changes given a series of consecutive shots. Scene change detection, intuitively, can be tackled from two aspects: recovering and comparing the similarity of background scenes in shots; analyzing the content of audio features. Nevertheless,

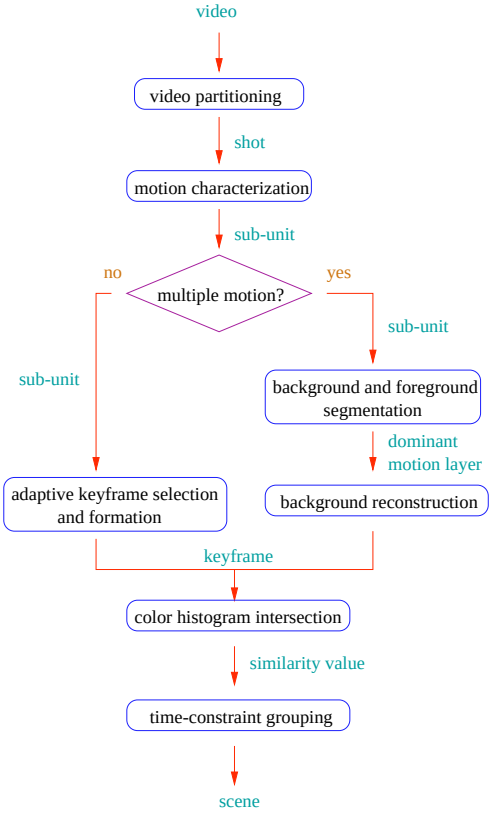
there are several research problems along this thought: (i) background and foreground segmentation; (ii) background and foreground identification; (iii) similarity matching, and (iv) word spotting from audio signal. The first problem can be solved satisfactorily only when the background and foreground objects have different motion patterns. The second problem requires high-level knowledge and, in most cases, necessitates manual feedback from human. The third problem has been addressed seriously since last decade, a good piece of work can be found in Refs. 69 and 70. The last problem is still regarded as hard since video soundtracks are complex and often mixed with many sound sources.

Based on the problems discussed above, scene change detection, in general, is considered as a difficult task. A fully automatic system cannot be easily realized. Research works on scene change detection includes Refs. 9, 23, 63, 71–79. Basically, there are two major approaches: one adopts the time-constraint clustering algorithm to group shots, which are visually similar and temporally closed as a scene;[23,63,71,72,74–77,79] the other employs audiovisual characteristics to detect scene breaks.[9,73,78] The former aims at representing a video in a compact yet semantically meaningful way, while the latter attempts to mimic human perception capability to detect scene breaks.

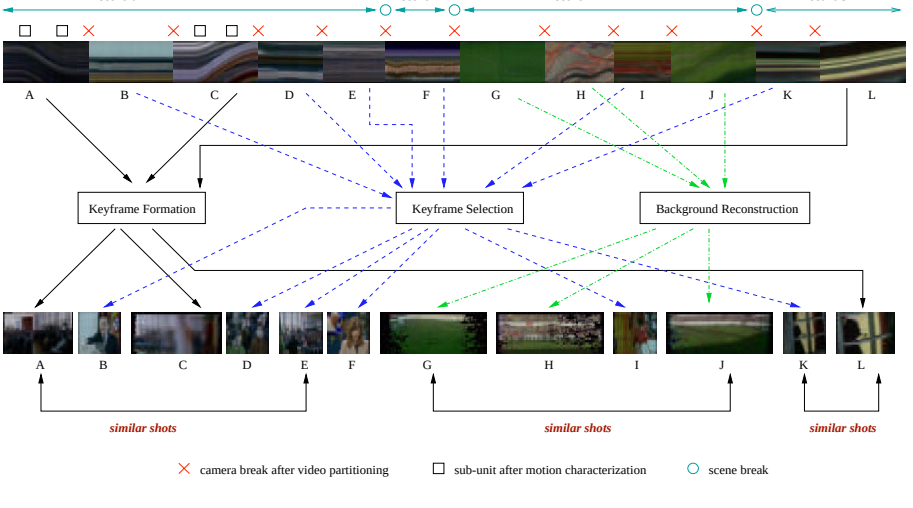
Here, we take the work of Ngo[23] as an example to show how scene change detection is attempted based on the visual information only. Basically, the problem is tackled from four different aspects: (i) represent shots adaptively and compactly through motion characterization; (ii) reconstruct background in the multiple motion case; (iii) reduce the distraction of foreground objects by histogram intersection;[80] (iv) impose time-constraint to group shots that are temporally closed.

Figure 5(a) depicts the basic framework developed in Ref. 23 for scene change detection. An input video is first partitioned into shots. Those shots that have more than one camera motion are temporally segmented into motion coherent sub-units (sub-shots), and each sub-unit is characterized according to its camera motion. A test is then conducted to check if a sub-unit has more than one motion (e.g., both camera and object motion). For multiple motion case, those sub-units are further spatially decomposed into motion layers. The dominant motion layer of a sub-unit is subsequently reconstructed to form a background image. For other cases, keyframe selection and formation are adaptively performed based on the annotated motion to compactly represent the content of a shot. Finally, scene change is detected by grouping shots with similar color content.

Figure 5(b) illustrates an example together with the techniques used. A spatio-temporal slice (the horizontal axis is time while the vertical axis is image space) is extracted from a video along the time dimension. This slice is partitioned into twelve regions (shots) using the video partitioning algorithm.[40,41] By employing the motion characterization technique,[47] shots A and C, for instance, are segmented into sub-units with static and panning motions. Based on the motion annotated information, keyframes are adaptively selected (shots B, D, E, F, I and K) and formed (shots A, C, L), in addition, background are reconstructed (shots G, H, J) for multiple motion case. Finally, color features are extracted from each keyframes for similarity



(a) Framework for scene change detection



(b) Example

Fig. 5. A framework for scene change detection.

measure through histogram intersection. As indicated in the figure, shots A and E (G and J) are considered similar, as a results, all shots from A to E (G to J) are grouped as one scene based on the time-constraint grouping algorithm.

4. Video Retrieval

While video browsing, keyframe based browsing in particular, has been achieved for practical applications, video retrieval is still at its preliminary state. Despite the fact that videos, in addition to image information, consist of one extra dimensional information i.e., scene dynamic to model scene composition, video retrieval is still regarded as a hard problem. Besides the lack of effective tools for representing and modeling spatio-temporal information, video retrieval confronts the same difficulties as image retrieval. The difficulties lie on the fact that utilizing low-level features for retrieval does not match human perception well in the general domain. Three problems that have been attempted are: retrieve similar videos;[81] locate similar video clips in a video;[82,83] retrieve similar shots.[5,23,49,50,84–87] Two major concerns for these problems are feature extraction and similarity measure.

4.1. *Feature extraction*

To date, techniques for video retrieval are mostly extended directly or indirectly from image retrieval techniques. Examples include first selecting keyframes from shots and then extracting image features such as color and texture features from those keyframes for indexing and retrieval. The success from such extension, however, is doubtful since the spatio-temporal relationship among video frames is not fully exploited. Recently, more works have been dedicated to address this problem,[23,49,50,77,84,85] more specifically, to exploit and utilize the motion information along the temporal dimension for retrieval.

Motion features that have been used for retrieval include the motion trajectories and motion trails of objects,[49,50] principle components of MPEG motion vectors,[77] and temporal texture.[23,85] Motion trajectories and trails are used to describe the spatio-temporal relationship of moving objects across time. The relationship can be indexed as 2D or 3D strings to support spatio-temporal search. Principle components are utilized to summarize the motion information in a sequence as several major modes of motion. In this way, clustering and retrieval can be done in an effective and efficient manner. Temporal texture are employed to model more complex dynamic motion such as the motion of river, swimming and crowds. As image texture, temporal texture can be modeled as co-occurrence matrix,[84,88] autoregressive model,[89] wold decomposition,[90] Gibbs random field[85] and tensor histogram.[23] The input to temporal texture can be optical flow field[84,85,88] or spatio-temporal slices.[23]

An important issue needed to be addressed is the decomposition of camera and object motion prior to feature extraction. Ideally, to fully explore the spatio-temporal relationship in videos, both camera and object motion need to be fully exploited in order to index the foreground and background information separately.

Motion segmentation is required especially when the targets of retrieval are objects of interest. In such applications, camera motion is normally canceled by global motion compensation and foreground objects are segmented by inter-frame subtraction.[85] However, such task is always turned up to be difficult, and most importantly, poor segmentation will always lead to poor retrieval results. Although motion decomposition is a preferable step prior to the feature extraction of most videos, it may not be necessary for certain videos. If we imagine a camera as a narrative eye, the movement of eye not only tells us what to be seen but also the different ways of observing events. Typical examples include the sport events that are captured by cameras, which are mounted at the fixed locations of a stand. These camera motion are mostly regular and driven by the pace of games and the type of events that are taken place. For these videos, camera motion is always an essential cue for retrieval. Furthermore, fixed motion patterns can always be observed when camera motion are coupled with the object motion of a particular event. In these applications, motion decomposition is not necessarily performed yet encouraging retrieval results can still be acquired, as demonstrated by the experiments conducted in Refs. 23 and 77.

4.2. *Similarity measure*

In general, similarity measure can be done by matching features either locally or globally. Local matching requires aligning and matching frames (or keyframes) across time. For instance, Tan *et al.*[83] employed dynamic programming to align two video sequences of different temporal length. Global matching, on the other hand, measures the similarity between two shots by computing the distance between the two representative features of shots. For retrieving similar videos, besides computing similarity among shots, the temporal order of similar shots between two videos are also taken into account.[81]

More sophisticated ways of similarity measure include spatio-temporal matching[50,91] and nearest feature line matching.[62] The spatio-temporal matching was proposed by Chang *et al.*[50] to measure the similarity of video objects, which are represented as trajectories and trails in the spatial and temporal domains. Recently, Dağtas[91] further presented various trajectory and trail based models for motion-based video retrieval. Their proposed models emphasize both the spatial and temporal scale invariant properties for object motion retrieval. In addition, Zhao[62] described shot similarity measure by employing the concept of nearest feature line. Initially, all frames in a shot are viewed as a curve. Frames that located at the corners are extracted as keyframes. Those keyframes are connected by lines and form a complete graph. Given a frame as a query, the distance from the frame to the graph is the nearest perpendicular projected distance among the frame to lines.

4.3. *Cluster-based retrieval*

Clustering is always a solution to abbreviate and organize the content of videos, in addition, provides an efficient indexing scheme for video retrieval since similar

shots are grouped under the same cluster. The proposed approaches that employ clustering structure for retrieval include Refs. 23, 53 and 92. For instance, Ngo *et al.* proposed a two-level hierarchical clustering structure to organize the content of sport videos. The top level is clustered by color features while the bottom level is clustered by motion features. The top level contains various clusters including wide-angle, medium-angle and close-up shots of players from different teams. The shots inside each cluster are partitioned to form sub-clusters in the bottom level according to their motion similarity. In this way, for example, the sub-cluster of a close-up shot can correspond either to "players running across the soccer field" or "players standing on the field". Such organization facilitates not only video retrieval and browsing, but also some high-level video processing tasks. For instance, to perform player recognition, only those shots in the cluster that correspond to close-up shots of players are picked up for processing. To perform motion-based background reconstruction, the sub-cluster corresponds to "players running across the soccer field" is further selected for processing. Through empirical results, Ngo shown that the cluster-based retrieval, in addition to speed up retrieval time, will generally give better results especially when a query is located at the boundary of two clusters.

5. Semantic Analysis

Despite the fact that most of the proposed frameworks and solutions are on the basis of low-level features, users would generally prefer to retrieve and browse video content at the semantic (or high) level. Nonetheless, building up a generic framework for bridging the gap between low and high levels, at the current state-of-the-art, is always impossible. This is mainly due to the limit of computational power and the lack of tools for representing and modeling human knowledge. To date, works on the semantic analysis of video content are either tailored to specific domain[5,10–12,27,77,93–98] or dependent on user interaction to improve performance.[49,99] These works are mainly devoted to the areas of video categorization,[27,77,93,95] highlight detection,[5,12,94,98] concept modeling[11,99] and semantic video parsing.[10,96,97] The features being utilized for accomplishing different applications include low-level visual and audio features, caption information, types of camera break, shot duration and activities, and video structure. Machine learning tools such as hidden Markov model (HMM), support vector machine (SVM) and vector quantization (VQ) are frequently employed for modeling semantic concept.

Video categorization, even at the coarse semantic level, is always useful for a video library for instance. To date, the domains being explored are mainly in movie,[27,93] commercial[95] and sport videos.[77] Besides low-level visual features, editing clues such as shot duration and types of camera break are always taken into account to derive abstract concept for categorization. For instance, Vasconcelos and Lippman[27] demonstrated their empirical results that the classification of movie clips into romance or comedy, action and other categories can be done directly in a 2D

feature space of shot duration and activity. Colombo *et al.*[95] further used the types of camera breaks, in addition to shot duration and activity, to map the relationship between the so-called emotional features and low-level perceptual features in commercial videos. The idea is that the frequent occurrence of cuts can infer the emotion of active, excitement and suspense, while dissolves can infer quietness. Their approach can classify commercial clips into practical, utopia, critical and playful videos. In addition, Sahouria and Zakhor[77] shown that the categorization of basketball, hockey and volleyball videos can be done by training VQ or HMM solely with the principle components of MPEG motion vectors as input features.

Highlight detection is normally application tailored since domain knowledge is required in most cases. Two specific domains that have captured researchers' attention are sport videos[5,12,94] and scene surveillance.[98] In Ref. 5, Tan *et al.* heuristically defined rules to detect the shots at the basket by using motion information encoded in MPEG videos. In Ref. 12, Rui *et al.* proposed a method to detect the highlight of baseball videos by audio features. The audio information being utilized are announcers' excited speech and the special sound effect like baseball hits. For certain types of sport videos, highlight can be identified by detecting replay sequences. By previewing the replay sequences, intuitively the highlight of sport videos are summarized. There are basically three types of replay sequence: (a) same shot that is repetitively shown for more than one time; (b) same shot that is shown in slow motion manner; (c) same scene but captured by different cameras at different viewpoints. The first two cases can be easily identified, the last case, however, is more difficult to be detected since domain specific knowledge such as background scene and the location of cameras is likely required. The problem can be abbreviated anyway if the replay sequences are sandwiched between a specific wipe pattern. This has been commonly practiced especially during the live production of sport videos. Noboru *et al.*[94] have made use of this fact to detect the highlight of TV programs of American football by semi-automatically detecting the wipe pattern embedded in the programs.

In surveillance applications, functionalities like detecting, indexing and retrieving abnormal or dangerous objects for alarm generation require video content analysis. Stringa and Regazzoni[98] presented a surveillance system to detect abandoned objects and to highlight the people who left them in an indoor environment. The system is equipped with video indexing and retrieval capabilities such that the human operator can quickly access the video clip of interest while scanning a large video library. In this system, since the background scene information is known and static while the camera is mounted at fixed location, foreground objects can be easily segmented. A multilayer perceptron is trained off-line to classified the foreground objects as abandoned objects, person, lighting or structural changes. By motion analysis, the person who left an abandoned object will be highlighted while the shape and color information of the object will be indexed for future reference.

Recently, machine learning and human-machine interaction techniques have also been incorporated to learn the object behavior and to model the semantic concept

among different objects. Two such examples are multiject-multinet framework[11] and semantic visual template.[99] Naphade and Huang[11] proposed a HMM model to semantically label certain multimedia objects (multijects) by fusing visual, audio and caption information. These multijects are further combined to form a multimedia network (multinet) modeled by Bayesian framework. Under this framework, complex queries like "explosion on a beach" can be more effectively answered. In addition, Chang *et al.*[99] proposed the use of semantic visual templates to describe the semantic concept of events. This approach emphasizes two way interaction and learning between human and machine. Initially, a user sketches the visual template of a concept. By relevancy feedback mechanism, the user checks the returned results by machine, while the machine generates more templates for retrieval based on the user's feedback. Ideally, templates of different concepts can be further combined to synthesize new visual semantic templates.

The structure of videos, news videos in particular, has also been explored for parsing of semantic events.[10,96,97] News videos consist of several headline stories, each of which is usually summarized by an anchor person prior to the detailed report. In addition, commercials are interleaved between different headline stories. With this video structure, the primary steps for semantic parsing of news stories are the anchor person identification and the separation of news and commercials. The former is normally achieved by template matching while the latter by audio signal analysis. Textual caption detection techniques[7,100] are also incorporated for more sophisticated indexing and browsing capabilities.

6. Conclusion

We have presented many efforts and techniques addressing video parsing, abstraction, retrieval and semantic analysis. It is clear that the number as well as the scope of research issues is rather large. More and more researchers from different fields are expected to explore these issues. Currently, more effort are driven towards the semantic modeling of video content through semi-automatic and application-oriented approaches. It should be aware that most existing works are served as tools to aid rather than to automate the analysis of video content. In addition, general solutions for video content analysis are always a long term research effort. In brief, to push the current success one step farther, more attempts are required to address the following opened problems:

- Detection of gradual transitions.
- Motion analysis for video segmentation, indexing, representation and retrieval.
- Integration and fusion of multimedia data such as video, audio and text.
- Modeling the spatio-temporal relationship of video objects for more sophisticated way of video retrieval.
- Knowledge representation and learning for semantic modeling of video content.

Acknowledgments

This work is supported in part by RGC Grants HKUST661/95E and HKUST6072/97E.

References

1. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkhani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The QBIC system," *IEEE Comput.* **28**(9), 23 (1995).
2. M. K. Mandal, F. Idris, and S. Panchanathan, "A critical evaluation of image and video indexing techniques in the compressed domain," *Image and Vision Comput.* **17**, 513 (1999).
3. C. W. Ngo, T. C. Pong, and R. T. Chin, "A survey of video parsing and image indexing techniques in compressed domain," in *Symposium on Image and Speech and Signal Processing and Robotics*, Vol. 1, 1998, pp. 231–236.
4. N. V. Patel and I. K. Sethi, "Compressed video processing for cut detection," *IEE Proc. Vision and Image and Signal Processing* **143**, 315 (1996).
5. 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," *IEEE Trans. Circuits Syst. Video Technology* **10**(1), 133 (2000).
6. B. L. Yeo and B. Liu, "On the extraction of dc sequence from MPEG compressed video," *IEEE Trans. Circuits Syst. Video Technology* **5**(6), 533 (1995).
7. Y. Zhong, H. J. Zhang, and A. K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. Pattern Analysis and Machine Intelligence* **22**(4), 385 (2000).
8. Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega, "A relevance feedback architecture for content-based multimedia information retrieval systems," in *IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997, pp. 82–89.
9. J. Huang, Z. Liu, and Y. Wang, "Integration of audio and visual information for content-based video segmentation," in *Int. Conf. on Image Processing*, Vol. 3, 1998, pp. 526–529.
10. Q. Huang, A. Puri, and Z. Liu, "Multimedia search and retrieval: New concepts and system implementation and application," *IEEE Trans. Circuits Syst. Video Technology* **10**(5), 679 (2000).
11. M. R. Naphade and T. S. Huang, "Semantic video indexing using a probabilistic framework," in *Int. Conf. on Pattern Recognition*, Vol. 3, 2000, pp. 83–88.
12. Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," in *ACM Multimedia*, 2000.
13. G. Ahanger and T. D. C. Little, "A survey of technologies for parsing and indexing digital video," *J. Visual Commun. Image Representation* **7**(1), 28 (1996).
14. P. Aigrain, H. J. Zhang, and D. Petkovic, "Content-based representation and retrieval of visual media: A state of the art review," *Multimedia Tools Appl.* **3**, 179 (1996).
15. R. Brunnelli, O. Mich, and C. M. Modena, "A survey on the automatic indexing of video data," *J. Visual Commun. Image Representation* **10**, 78 (1999).
16. S. F. Chang, T. S. Huang, Q. Huang, A. Puri, and B. Shahraray, "Multimedia search and retrieval," in *Advances in Multimedia*: *Systems*, *Standards*, *and Networks*, eds. B. Furht and T. Chen (Marcel Dekker, New York, 1999).

17. P. Correia and F. Pereira, "The role of analysis in content-based video coding and indexing," *Signal Processing* **66**, 125 (1998).
18. F. Idris and S. Panchanathan, "Review of image and video indexing techniques," *J. Visual Commun. Image Representation* **8**(2), 146 (1997).
19. H. J. Zhang, "Content-based video browsing and retreival," in *Handbook of Multimedia Computing*, ed. B. Furht (CRC Press, 1999), pp. 255–280.
20. B. S. Manjunath, T. S. Huang, A. M. Tekalp, and H. J. Zhang, "Introduction to the special issue on image and video processing for digital libraries," *IEEE Trans. Image Processing* **9**(1), 1 (2000).
21. F. Pereira, S. F. Chang, R. Koenen, A. Puri, and O. Avaro, "Introduction to the special issue on object-based video coding and description," *IEEE Trans. Circuits Syst. Video Technology* **9**(8), 1144 (1999).
22. W. Wolf, "Video libraries," *Multimedia Systems* **7**(5), 349 (1999).
23. C. W. Ngo, "Analysis of spatio-temporal slice for video content representation," Ph.D. Thesis, Hong Kong University of Science & Technology, 2000.
24. H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," in *ACM Multimedia System*, Vol. 1, 1993, pp. 10–28.
25. R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks," in *ACM Multimedia*, Nov. 1995, pp. 189–200.
26. P. Bouthemy, M. Gelgon, and F. Ganansia, "A unified approach to shot change detection and camera motion characterization," *IEEE Trans. Circuits Syst. Video Technology* **9**(7), 1030 (1999).
27. N. Vasconcelos and A. Lippman, "Statistical models of video structure for content analysis and characterization," *IEEE Trans. Image Processing* **9**(1), 3 (2000).
28. L. F. Cheong, "Scene-based shot change detection and comparative evaluation," *J. Comput. Vision and Image Understanding* **79**(2), 224 (2000).
29. U. Gargi, R. Kasturi, and S. H. Strayer, "Performance characterization of video-shot-change detection methods," *IEEE Trans. Circuits Syst. Video Technology* **10**(1), 1 (2000).
30. V. Kobla, D. Dementhon, and D. Doermann, "Special effect edit detection using video trails: A comparison with existing techniques," in *SPIE Proc. Storage and Retrieval for Image and Video Database VII*, 1999, pp. 302–313.
31. R. Lienhart, "Comparison of automatic shot boundary detection algorithm," in *SPIE Proc. Storage and Retrieval for Image and Video Database VII*, 1999, pp. 290–301.
32. M. S. Drew, S. N. Li, and X. Zhong, "Video dissolve and wipe detection via spatio-temporal images of chromatic histogram differences," in *Int. Conf. on Image Processing*, Vol. 3, 2000, pp. 929–932.
33. W. A. C. Fernando, C. N. Canagarajah, and D. R. Bull, "Wipe scene change detection in video sequences," in *Int. Conf. on Image Processing*, Vol. 3, 1999, pp. 294–298.
34. W. A. C. Fernando, C. N. Canagarajah, and D. R. Bull, "Fade and dissolve detection in uncompressed and compressed video sequences," in *Int. Conf. on Image Processing*, Vol. 3, 1999, pp. 299–303.
35. R. A. Joyce and B. Liu, "Temporal segmentation of video using frame and histogram space," in *Int. Conf. on Image Processing*, 2000.
36. H. Kim, S.-J. Park, J. Lee, W. M. Kim, and S. M. Song, "Processing of partial video data for detection of wipes," in *SPIE Proc. Storage and Retrieval for Image and Video Database VII*, 2000, pp. 280–289.
37. Z. N. Li, Z. Tauber, and M. S. Drew, "Spatio-temporal joint probability images for video segmentation," in *Int. Conf. on Image Processing*, Vol. 2, 2000, pp. 295–298.
38. H. B. Lu, Y. J. Zhang, and Y. R. Yao, "Robust gradual scene change detection," in *Int. Conf. on Image Processing*, Vol. 3, 1999, pp. 304–308.

39. J. Nang, S. Hong, and Y. Ihm, "An efficient video segmentation scheme for MPEG video stream using macroblock information," in *ACM Multimedia*, 1999, pp. 23–26.
40. C. W. Ngo, T. C. Pong, and R. T. Chin, "Detection of gradual transitions through temporal slice analysis," in *Computer Vision and Pattern Recognition*, Vol. 1, 1999, pp. 36–41.
41. C. W. Ngo, T. C. Pong, and R. T. Chin, "A robust wipe detection algorithm," in *Asian Conference on Computer Vision*, Vol. 1, 2000, pp. 246–251.
42. S. B. Jun, K. Yoon, and H. Y. Lee, "Dissolve transition detection algorithm using spatio-temporal distribution of MPEG macro-block types," in *ACM Multimedia*, 2000.
43. S. W. Lee, Y. M. Kim, and S. W. Choi, "Fast scene change detection using direct feature extraction from MPEG compressed videos," *IEEE Trans. Multimedia* **2**(4), 240 (2000).
44. W. Xiong and C. M. Lee, "Efficient scene change detection and camera motion annotation for video classification," *J. Comput. Vision and Image Understanding* **17**(2), 161 (1998).
45. K. A. Hua and J. H. Oh, "Detecting video shot boundaries up to 16 times faster," in *ACM Multimedia*, 2000.
46. M. S. Drew, J. Wei, and Z. N. Li, "Illumination-invariant image retrieval and video segmentation," *Pattern Recognition* **32**(8) (1999).
47. C. W. Ngo, T. C. Pong, H. J. Zhang, and R. T. Chin, "Motion characterization by temporal slice analysis," in *Computer Vision and Pattern Recognition*, Vol. 2, 2000, pp. 768–773.
48. M. Irani and P. Anandan, "Video indexing based on mosaic representations," *Proc. IEEE* **86**, 905 (1998).
49. D. Zhong and S. F. Chang, "An integrated approach for content-based video object segmentation and retrieval," *IEEE Trans. Circuits Syst. Video Technology* **9**(8), 1259 (1999).
50. S. F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A fully automatic content-based video search engine supporting multi-object spatio-temporal queries," *IEEE Trans. Circuits Syst. Video Technology* **8**(5), 602 (1998).
51. H. T. Ngyuyen, M. Worring, and A. Dev, "Detection of moving objects in video using a robust motion similarity measure," *IEEE Trans. Image Processing* **9**(1), 137 (2000).
52. H. S. Sawhney and S. Ayer, "Compact representations of videos through dominant and multiple motion estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence* **18**(8), 814 (1996).
53. H. S. Chang, S. S. Sull, and S. U. Lee, "Efficient video indexing scheme for content-based retrieval," *IEEE Trans. Circuits Syst. Video Technology* **9**(8) (1999).
54. C.-Y. Chang, A. A. Maciejewski, and V. Balakrishnan, "Eigendecomposition-based analysis of video images," in *SPIE Proc. Storage and Retrieval for Image and Video Database VII*, 1999, pp. 186–191.
55. D. DeMenthon, V. Kobla, and D. Doermann, "Video summarization by curve simplification," in *ACM Multimedia*, 1998, pp. 211–218.
56. A. M. Ferman and A. M. Tekalp, "Efficient filtering and clustering methods for temporal video segmentation and visual summarization," *J. Visual Commun. Image Representation* **9**(4), 336 (1998).
57. M. S. Drew and J. Au, "Video keyframe production by efficient clustering of compressed chromaticity signatures," in *ACM Multimedia*, 2000, pp. 365–368.
58. Y. Gong and X. Liu, "Video summarization using singular value decomposition," in *Computer Vision and Pattern Recognition*, Vol. 2, 2000, pp. 174–180.

59. A. Hanjalic and H. J. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *IEEE Trans. Circuits Syst. Video Technology* **9**(8), 1280 (1999).
60. S. Uchihashi and J. Foote, "Summarizing video using a shot importance measure and a frame-packing algorithm," in *Int. Conf. on Acoustics and Speech and Signal Processing*, Vol. 6, 1999, pp. 3041–3044.
61. S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky, "Video manga: Generating semantically meaningful video summaries," in *ACM Multimedia*, 1999, pp. 383–392.
62. L. Zhao, W. Qi, S. Z. Li, S. Q. Yang, and H. J. Zhang, "Key-frame extraction and shot retrieval using nearest feature line (NFL)," in *International Workshop on Multimedia Information Retrieval*, 2000.
63. C. W. Ngo, T. C. Pong, H. J. Zhang, and R. T. Chin, "Motion-based video representation for scene change detection," in *Int. Conf. Pattern Recognition*, Vol. 1, 2000, pp. 827–830.
64. H. J. Zhang, J. Y. A. Wang, and Y. Altunbasak, "Content-based video retrieval and compression: A unified solution," in *Int. Conf. on Image Processing*, 1997, pp. 13–16.
65. J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Trans. Image Processing* **3**(5), 625 (1994).
66. B. Erol and F. Kossentini, "Automatic key video object plane selection using the shape information in the MPEG-4 compressed domain," *IEEE Trans. Multimedia* **2**(2), 129 (2000).
67. B. Gunsel, A. M. Tekalp, and P. J. L. Beek, "Content-based access to video objects: Temporal segmentation, visual summarization and feature extraction," *Signal Processing* **66**, 261 (1998).
68. C. Kim and J. N. Hwang, "An integrated scheme for object-based video abstraction," in *ACM Multimedia*, 2000.
69. D. W. Jacobs, D. Weinshall, and Y. Gdalyahu, "Classification with nonmetric distances: Image retrieval and class representation," *IEEE Trans. Pattern Analysis and Machine Intelligence* **22**(6), 583 (2000).
70. S. Santini and R. Jain, "Similarity matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1999.
71. J. M. Corridoni and A. Del. Bimbo, "Structured representation and automatic indexing of movie information content," *Pattern Recognition* **31**(12), 2027 (1998).
72. A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video retrieval systems," *IEEE Trans. Circuits Syst. Video Technology* **9**(5), 580 (1999).
73. H. Jiang, T. Lim, and H. J. Zhang, "Video segmentation with the assistance of audio content analysis," in *Int. Conf. on Multimedia and Expo*, 2000.
74. T. Lin and H. J. Zhang, "Automatic video scene extraction by shot grouping," in *Int. Conf. on Pattern Recognition*, Vol. 4, 2000, pp. 39–42.
75. Y. Rui, T. S. Huang, and S. Mehrotra, "Exploring video structure beyong the shots," in *Proc. IEEE Conf. on Multimedia Computing and Systems*, 1998, pp. 237–240.
76. Y. Rui, T. S. Huang, and S. Mehrotra, "Constructing table-of-content for videos," *Multimedia Syst.* **7**(5), 359 (1999).
77. E. Sahouria and A. Zakhor, "Content analysis of video using principle components," *IEEE Trans. Circuits Syst. Video Technology* **9**(8), 1290 (1999).
78. H. Sundaram and S. F. Chang, "Determining computable scenes in films and their structure using audio-visual memory models," in *ACM Multimedia*, 2000.
79. M. M. Yeung and B. L. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content," *IEEE Trans. Circuits Syst. Video Technology* **7**(5), 771 (1997).

80. M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vision* **7**(1), 11 (1991).
81. Y. Wu, Y. Zhuang, and Y. Pan, "Content-based video similarity model," in *ACM Multimedia*, 2000.
82. A. K. Jain, A. Vailaya, and W. Xiong, "Query by video clip," *Multimedia Syst.* **7**(5), 368 (1999).
83. Y. P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge, "A framework for measuring video similarity and its application to video query by example," in *IEEE Int. Conf. on Image Processing*, 1999.
84. P. Bouthemy and R. Fablet, "Motion characterization from temporal co-occurrences of local motion-based measures for video indexing," in *Int. Conf. on Pattern Recognition*, 1998, pp. 905–908.
85. R. Fablet, P. Bouthemy, and P. Perez, "Statistical motion-based video indexing and retrieval," in *Int. Conf. on Content-based Multimedia Info. Access*, 2000, pp. 602–619.
86. J. Lee and B. W. Dickinson, "Hierarchical video indexing and retrieval for subband-coded video," *IEEE Trans. Circuits Syst. Video Technology* **10**(5), 824 (2000).
87. Z. Lei and Y. T. Lin, "3D shape inferencing and modeling for video retrieval," *J. Visual Commun. Image Representation* **11**(1), 41 (2000).
88. R. Nelson and R. Polana, "Qualitative recognition of motion using temporal texture," *CVGIP*: *Image Understanding* **56**(1), 78 (1992).
89. M. O. Szummer, "Temporal texture modeling," Ph.D. Thesis, MIT, 1995.
90. F. Liu and R. W. Picard, "Finding periodicity in space and time," in *Int. Conf. on Computer Vision*, 1998, pp. 376–383.
91. S. Dağtas, W. A. Khatib, A. Ghafoor, and R. L. Kashyap, "Models for motion-based video indexing and retreival," *IEEE Trans. on Image Processing* **9**(1), 88 (2000).
92. W. Zhou, A. Vellaikal, and C.-C. Jay Kuo, "Rule-based video classification system for basketball video indexing," in *International Workshop on Multimedia Information Retrieval*, 2000.
93. B. Adams, C. Dorai, and S. Venkatesh, "Study of shot length and motion as contributing factors to movie tempo," in *ACM Multimedia*, 2000.
94. N. Babaguchi, Y. Kawai, Y. Yasugi, and T. Kitahashi, "Linking live and replay scenes in broadcasted sport video," in *International Workshop on Multimedia Information Retrieval*, 2000.
95. C. Colombo, A. D. Bimbo, and P. Pala, "Semantics in visual information retrieval," *IEEE Multimedia* **6**(3) (1999).
96. A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Semi-automatic news analysis and indexing and classification system based on topics preselection," in *SPIE Proc. Storage and Retrieval for Image and Video Database VII*, 1999, pp. 86–97.
97. Q. Huang, Z. Liu, and A. Rosenberg, "Automatic semantic structure reconstruction and representation generation for broadcast news," in *SPIE Proc. Storage and Retrieval for Image and Video Database VII*, 1999, pp. 50–62.
98. E. Stringa and C. S. Regazzoni, "Real time video-shot detection for scene surveillance applications," *IEEE Trans. Image Processing* **9**(1) (2000).
99. S. F. Chang, W. Chen, and H. Sundaram, "Semantic visual templates: Linking visual features to semantics," in *Int. Conf. on Image Processing*, 1998.
100. H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video," *IEEE Trans. Image Processing* **9**(1), 147 (2000).

Chong-Wah Ngo received his Ph.D. from the Hong Kong University of Science & Technology (HKUST) in 2000, and his B.S. with honors in 1994 and M.S. in 1996, both in Computer Engineering, from Nanyang Technological University, Singapore. He is currently a Research Associate in HKUST. He was with Information Technology Institute, Singapore, in 1996, and was with Microsoft Research China as a summer intern in 1999. His current research interests include image and video indexing, computer vision and pattern recognition.

Hong-Jiang Zhang received his Ph.D. from the Technical University of Denmark and his B.S. from Zheng Zhou University, China, both in Electrical Engineering, in 1982 and 1991, respectively. From 1992 to 1995, he was with the Institute of Systems Science, National University of Singapore, where he led several projects in video and image content analysis and retrieval and computer vision. He also worked at MIT Media Lab in 1994 as a Visiting Researcher. From 1995 to 1999, he was a Research Manager at Hewlett-Packard Labs, where he was responsible for research and technology transfers in the areas of multimedia management; intelligent image processing and Internet media. In 1999, he joined Microsoft Research, China, where he is currently a Senior Researcher and the Assistant Managing Director mainly in charge of media computing and information processing research. Dr. Zhang is a Senior Member of IEEE and a Member of ACM. He has authored three books, over 120 refereed papers and book chapters, seven special issues of international journals in multimedia processing, content-based media retrieval, and Internet media, as well as numerous patents or pending applications. He currently serves on the editorial boards of five professional journals and a dozen committees of international conferences.

Ting-Chuen Pong received his Ph.D. in Computer Science from Virginia Polytechnic Institute and State University in 1984. In 1991, Dr. Pong joined the Hong Kong University of Science & Technology, where he is currently a Reader of Computer Science and Associate Dean of Engineering. Before joining HKUST, he was an Associate Professor in Computer Science at the University of Minnesota, Minneapolis. Dr. Pong is a recipient of the Annual Pattern Recognition Society Award in 1990 and Honorable Mention Award in 1986. He has served as Program Co-Chair of the Third International Computer Science Conference in 1995 and the Third Asian Conference on Computer Vision in 1998. He is currently on the Editorial Board of the Pattern Recognition Journal.