Video Summarization and Scene Detection by Graph Modeling

Chong-Wah Ngo, *Member, IEEE*, Yu-Fei Ma, *Member, IEEE*, and Hong-Jiang Zhang, *Fellow, IEEE*

***Abstract*—In this paper, we propose a unified approach for video summarization based on the analysis of video structures and video highlights. Two major components in our approach are scene modeling and highlight detection. Scene modeling is achieved by normalized cut algorithm and temporal graph analysis, while highlight detection is accomplished by motion attention modeling. In our proposed approach, a video is represented as a complete undirected graph and the normalized cut algorithm is carried out to globally and optimally partition the graph into video clusters. The resulting clusters form a directed temporal graph and a shortest path algorithm is proposed to efficiently detect video scenes. The attention values are then computed and attached to the scenes, clusters, shots, and subshots in a temporal graph. As a result, the temporal graph can inherently describe the evolution and perceptual importance of a video. In our application, video summaries that emphasize both content balance and perceptual quality can be generated directly from a temporal graph that embeds both the structure and attention information.**

***Index Terms*—Attention model, normalized cut, scene modeling, video summarization.**

I. Introduction

RECENTLY, techniques for automatic video content summarization have attracted numerous attention due to its commercial potential especially for home video applications. A concise video summary, intuitively, should highlight the video content and contain little redundancy while preserving the balance coverage of the original video. A video summary, nevertheless, should be different from video trailers where certain contents are intentionally hidden so as to magnify the attraction of a video.

Techniques in automatic video summarization, in broad, can be categorized into two major approaches: static storyboard summary [1], [3], [5], [23] and dynamic video skimming [4], [9], [10], [19]. The former is a collection of static keyframes of video shots, while the latter is a shorter version of video composed of a series of selected video clips. Static storyboard allows nonlinear browsing of video content by sacrificing the temporal evolution of a video. Dynamic video skimming, in contrast, preserves the time-evolving nature of a video by linearly and continuously browsing certain portions of video content depending on a given time length. For both approaches, the appropriate selection of video segments plays a major role in maximizing the entropy information and perceptual quality of a video summary.

To date, compared with static storyboard summary, there are relatively few works that address dynamic video skimming. Nonetheless, due to the advance and popularity of audio-visual capturing tools, effective techniques for dynamic video skimming is highly in demand. Imagine that most people will be bored by an unedited and long-winded home video that lasts for hours. A tool that can automatically shorten the original video while preserving most events by highlighting only the important content would be greatly useful to most users.

Techniques for dynamic video skimming include applying expectation maximization (EM) [17], singular value decomposition (SVD) [4], motion model [12], [22], utility framework [21], attention model [13], and semantic analysis [9], [19]. Most techniques are based mainly on visual information except approaches like [9], [19] where audio and linguistic information are also incorporated in order to derive semantic meaning. In [9], audio and motion signals are used to detect emotional dialogues and violent scenes for summarization. However, this approach can only be applied to certain videos, and the resulting summaries may not be useful in revealing the content coverage. In [19], the InfoMedia system was developed to generate the short synopsis of a video. Language understanding techniques are applied with the aid of audio and visual features. Nevertheless, this text-driven approach could not generate satisfactory results when speech signals are noisy, which happens frequently in life video recording.

Recently, SVD emerges as an attractive computational model for video summarization [4]. However, this approach is computationally intensive since it operates directly on video frames. In [10], a hierarchical tree that consists of events, activities, actions, and shots is constructed to represent the video content. Then a summary is generated by randomly removing subtrees at different levels to meet the output video length. In [21], the rules of cinematic syntax are utilized to give the syntactical-based reduction schemes for summarization. Utility functions are derived to maximize the content and coherence of summaries based on the audio-visual information. Besides [4], [10], and [21], other sophisticated mathematical models include [17] and [22]. However, these models are only applied to a single video shot. It is unclear how to extend their works to summarize an entire video.

Most existing approaches emphasize either content coverage [4], [21], [10] or perceptual quality (highlight) [9], [12], [13]. In this paper, we propose a unified approach for dynamic video

Manuscript received April 3, 2003; revised July 3, 2003. This work was supported in part by a grant from the Research Grants Council of the Hong Kong Special Administrative Region under Project CityU 1072/02E and by a grant from the City University of Hong Kong under Project 7001546. This paper was recommended by Associate Editor R. Lancini.

C.-W. Ngo is with the Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong (e-mail: cwngo@cs.cityu.edu.hk).

Y.-F. Ma and H.-J. Zhang are with Microsoft Research Asia, Beijing 100080, R.O.C. (e-mail: yfma@microsoft.com; hjzhang@microsoft.com).

Digital Object Identifier 10.1109/TCSVT.2004.841694

skimming that emphasizes both content coverage and perceptual quality and is capable of reducing content redundancy. Our previous works in video skimming can be found in [12] and [13]. In [12], skimming is achieved by modeling and detecting the motion-attended regions in videos. Specifically, summaries are generated by gluing together those video segments that contain high confidence scores in the motion-attended regions. In [13], the attention model in [12] is further generalized by considering the static, face, and audio information. One limitation of [12] and [13] is that the structural information such as the intershot relationship is not exploited for video skimming. As a result, a video summary is solely a collection of video highlights that do not take into account the content coverage. Similar highlights may be repeatedly shown in a summary.

In this paper, we propose approaches to tackle the problems in [12] and [13]. The major contributions in this paper are as follows.

- A unified approach is proposed to capture both the *video structure* and *attention values* for video summarization.
- To maintain *content balance* and *reduce redundancy*, a video is structured according to scenes, clusters, shots, and subshots in a hierarchical tree. Two major techniques are: *normalized cut* algorithm for the decomposition of a video into clusters and *temporal graph analysis* for scene change detection.
- To measure *perceptual quality*, an *attention model* is employed to model human's attention when viewing a video.
- The selection of video clips for summarization is based jointly on the probability of subtrees and their attention values.

In the remaining two subsections, we will first describe the basic approaches (e.g., shot detection and keyframe construction) that we adopt for structuring video content. Then, we present an overview of our approach for video summarization and scene detection.

A. Video Structure

A video usually consists of scenes, and each scene includes one or more shots. A shot is an uninterrupted segment of video frame sequence with static or continuous camera motion, while a scene is a series of shots that are coherent from the narrative point of view. These shots are either shot in the same place or they share similar thematic content. Clusters can be viewed as intermediate components between shots and scenes. Basically, each cluster contains one or more shots with similar visual content.

To structure videos, we adopt the approaches in [14]–[16] to temporally partition videos into shots and then into subshots. These approaches are based on the analysis of motion patterns extracted directly from three-dimensional (3-D) spatial-temporal image volumes. In addition, we apply the adaptive keyframe selection and construction scheme proposed in [15] to select/construct one keyframe for each subshots, as shown in Table I. These keyframes are used for shot similarity measure by the proposed normalized cut algorithm to decompose videos into clusters. The similarity measure is based on the video representation techniques given in [15].

TABLE I
VIDEO REPRESENTATION THROUGH KEYFRAME SELECTION AND CONSTRUCTION

Motion Type	Action
static	select one frame
pan or tilt	construct a panoramic image
zoom	select the first and last image
multiple motion	segment and construct foreground and background scenes
indeterministic	select one frame

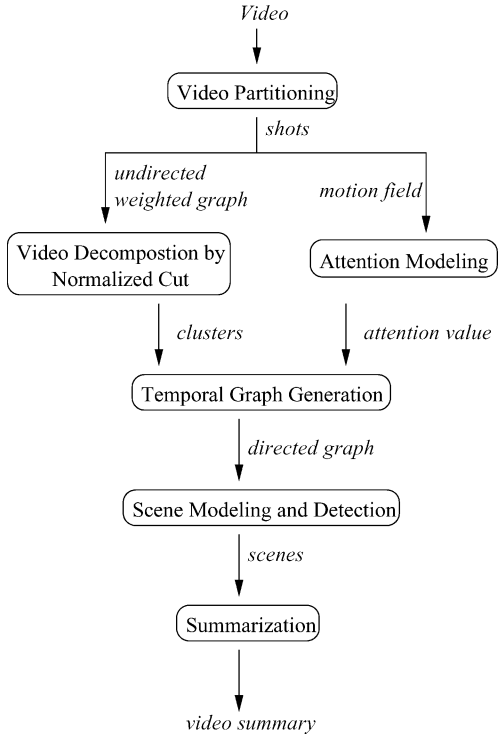


Fig. 1. Proposed approach.

B. Overview of our Approach

Fig. 1 illustrates the flow of our proposed approach. The whole process is carried out in MPEG compressed domain. Initially, a video is temporally partitioned into shots based on a spatio-temporal slice model in [14]. The model extracts three temporal slices horizontally, vertically, and diagonally from an image volume. These slices are basically two-dimensional (2-D) images with one dimension in space and the other in time. We employ jointly the color, texture, and statistical information to segment the slices into regions that are originally connected by cuts, wipes, or dissolves. Each region basically corresponds to one shot after video partitioning. Based on these shots, a complete undirected weighted graph, with shots as its nodes and with shot similarities as its edges, is constructed to model the similarity among all pairs of shots in a video. We employ a global criterion, normalized cut [18], to optimally decompose the graph into subgraphs (clusters). Normalized cut criterion takes into account the total dissimilarity among clusters and the total similarity within clusters for graph partitioning. Ideally,

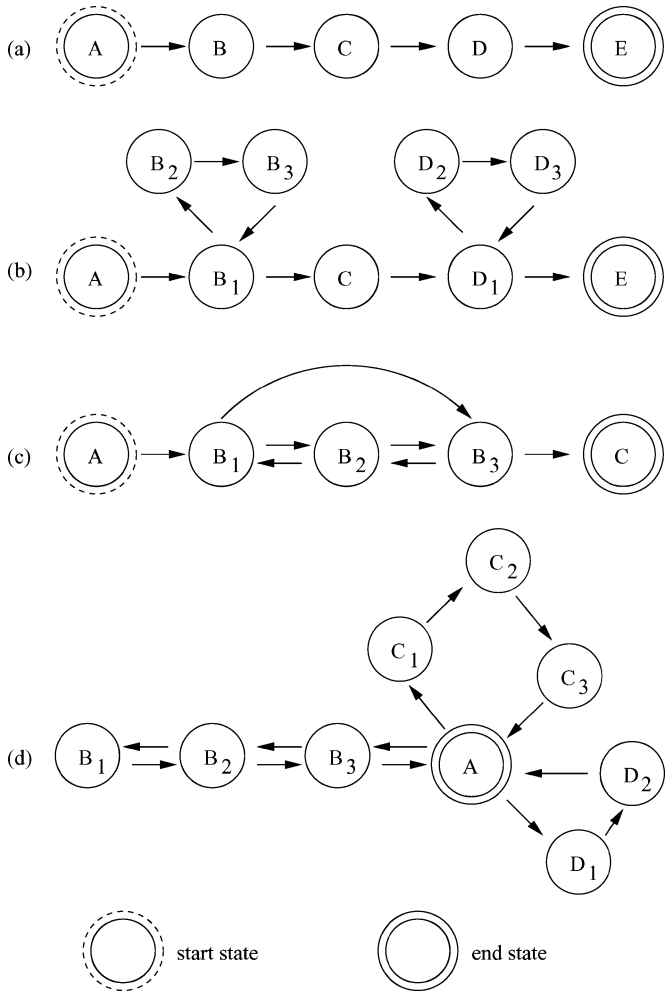


Fig. 2. Examples of temporal graphs.

shots in a cluster will share similar video content after the partitioning.

Meanwhile, a motion attention model is utilized to compute the perceptual attention of video shots based on the MPEG motion vector flow field. The computed attention values and the partitioned subgraphs form a directed temporal graph. This graph captures both the attention value and the occurrence probability of every cluster and, most importantly, describes the scene structure of a video. As a result, a simple approach based on the shortest path algorithm is proposed to analyze and detect scene transitions. Once scene changes are detected, video structure is constructed hierarchically in the form of scenes, clusters, shots, subshots, and keyframes. A summary is then generated in the hierarchical top-down manner. In our approach, the video structure provides useful hints for maintaining the content balance of a summary, while the attention values captured in a temporal graph facilitate the selection of useful video clips.

The concept of temporal graph is similar to scene transition graph in [23]. Fig. 2 shows several examples of temporal graph. Each circle represents a cluster of shots. Intuitively, a temporal graph is utilized because of its capability in describing the narrative flow of a video. For instance, in Fig. 2, (a) shows a progressive story line, (b) shows the detailed description of two events, (c) shows the interaction of different clusters (e.g., dialogue), while (d) shows the description of several events anchored by a center clusters (e.g., news). The inherent structure of a temporal graph provides important cues to model and describe the scene composition. For instance, $B_1 - B_3$ and $D_1 - D_3$ in Fig. 2(b) can be viewed as two different scenes. In our approach, scenes are detected by segmenting a temporal graph into subgraphs where each subgraph corresponds to one scene. For Fig. 2(a)–(c), these subgraphs can be easily obtained by removing the edges that along the shortest path from the start state to the end state. In Fig. 2(d), subgraphs can be obtained by removing all edges connecting the start state. Scene decomposition has been actively studied in [5], [15], [20], and [23]. Previous attempts are mostly based on the time-constraint clustering or grouping algorithms. Our approach is different in that, instead of depending on a time-constrain threshold, normalized cut is employed to optimally obtain clusters while the shortest path algorithm is utilized to efficiently detect scenes.

The paper is organized as follows. Section II describes video decomposition by normalized cut algorithm. Section III presents the construction and properties of temporal graphs. Section IV proposes an approach for modeling scene decomposition, while Section V presents a computational attention model based on motion information. Finally, Section VI combines both scene modeling and attention computation for video summarization. Section VII shows our experimental results, and Section VIII concludes this paper.

II. Video Decomposition

A video is initially represented as a weighted undirected graph that composes of shots. Let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ denote a graph, where the vertices $\mathbf{V}$ are the feature points of shots and edges $\mathbf{E}$ connect every pair of vertices. The weight on each edge $w(i, j)$ is a function that measures the similarity between shots i and j. In our approach, the normalized cut algorithm [18] is adopted to recursively bipartition $\mathbf{G}$ into clusters (disjoint sets) of shots. Normalized cut can optimally partition a graph $\mathbf{G}$ into two disjoint sets A and B ($A \cup B = \mathbf{V}$) by removing edges between A and B. Mathematically, we have

$$Ncut(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, \mathbf{V})} + \frac{\text{cut}(A, B)}{\text{assoc}(B, \mathbf{V})} \tag{1}$$

where $\text{cut}(A, B) = \sum_{i \in A, j \in B} w(i, j)$ is a cut value, and $\text{assoc}(A, \mathbf{V}) = \sum_{i \in A, j \in \mathbf{V}} w(i, j)$ is the total connection from the vertices of a set to all vertices in $\mathbf{G}$. The optimal bipartitioning of $\mathbf{G}$ is the one that minimizes Ncut. Equation (1) can be transformed into a standard eigen system

$$\mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2} z = \lambda z \tag{2}$$

where $\mathbf{D}$ is a diagonal matrix with $\sum_j w(i, j)$ on its diagonal and $\mathbf{W}$ is a symmetrical matrix with $w(i, j)$ as its elements. The eigen vector that corresponds to the second smallest eigen value can be utilized to find sets A and B.

The detailed algorithm for video decomposition consists of the following steps.

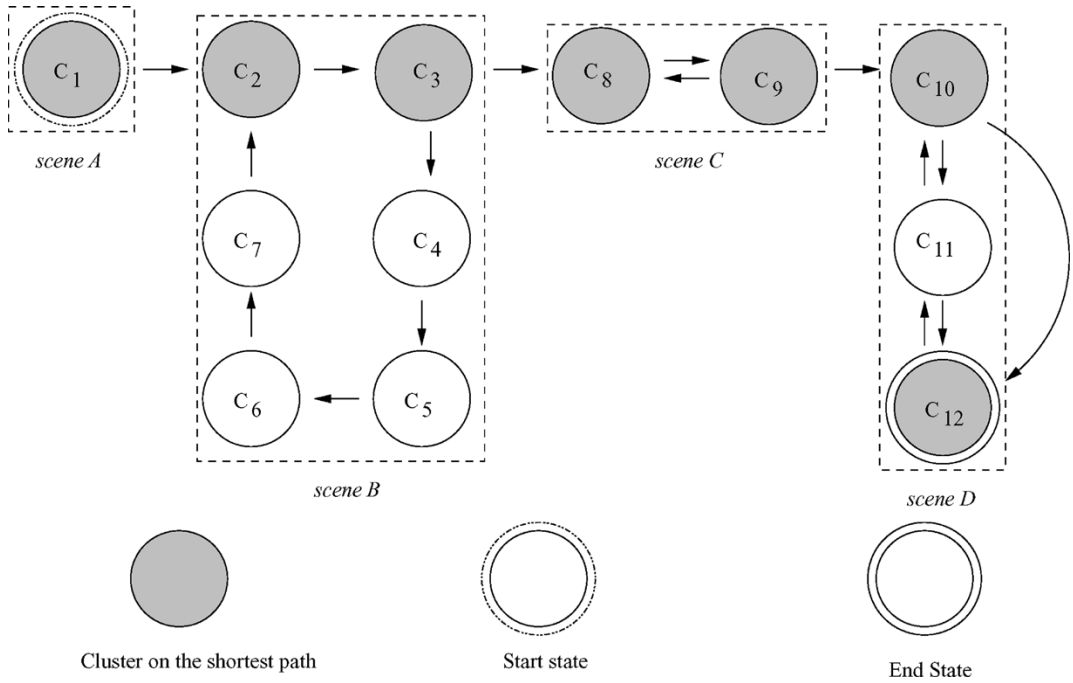


Fig. 3. Temporal graph and scene change detection.

- Partition a video temporally into shots, and set up a weighted graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. The weight $w(i, j)$ on the edge connecting shots i and j is

$$w(i,j) = \exp\left\{\frac{-k \times |f_j - f_i|}{\mathbf{T}} \times Sim(i,j)\right\} \tag{3}$$

  which takes into account the color similarity,[1] $Sim(i, j)$, and temporal frame distance, $|f_j - f_i|$, between two shots i and j. The parameter k is used to emphasize the importance of temporal distance. Intuitively, the similarity between two shots should be inversely proportional to their temporal distance. In our experiment, k is set to 8. The normalization constant $\mathbf{T}$ is the total number of frames in a video.
- Solve (2) and employ the eigen vector that corresponds to the second smallest eigen value to bipartition $\mathbf{G}$. The value 0 is used as the splitting point to divide the eigen vector into two parts. The algorithm is run recursively for the two partitioned subgraphs and terminated when the similarity between all pairs of shots in a subgraph is lower than an adaptive threshold $T_s = \mu + \sigma$, where μ and σ are, respectively, the average and standard deviations of shot similarity between all pairs of shots in a given video.

By recursively decomposing $\mathbf{G}$ into two subgraphs, in fact, we form a binary tree that could be utilized directly for hierarchical video browsing. In our case, only the leaves of the binary tree are used to form the clusters of a video.

[1]The similarity between two shots s_i and s_j is based on the color similarity among the keyframes in s_i and s_j. The similarity measure between a pair of keyframes is based on the histogram intersection in hue, saturation, and intensity (HSV) color space. The exact details can be found in [15].

III. Temporal Graph Generation

Once $\mathbf{G}$ is partitioned into subgraphs, a set of clusters that consists of temporally adjacent or nonadjacent shots is obtained. The temporal relationship among these clusters can be constructed to form a temporal graph $\mathbf{TG} = (\mathbf{V}, \mathbf{E})$ by adding the time order information of video shots. $\mathbf{TG}$ is a directed graph, with clusters as its nodes $\mathbf{V}$, and the transition probabilities among clusters as its edges $\mathbf{E}$. If we order the shots as $\{\ldots, s_i, s_{i+1}, \ldots\}$ in time order, a directed edge is added from a cluster C_n to another cluster C_m if there is a shot s_i in C_n and another shot s_{i+1} in C_m. In other words, C_m transits to C_n if there exists a pair of shots that are temporally adjacent.

The temporal graph is basically a state transition diagram (or Markov chain) that models the evolution of a video from states to states. In this context, a state is equivalent to a cluster. See Fig. 3 for an illustration of a temporal graph. Each cluster is modeled by two parameters: its prior probability $P_r(C_m)$ and attention value $\mathcal{A}(C_m)$. Every pair of clusters can be further modeled by a transition probability $P_r(C_m|C_n)$. Mathematically, they are computed by

$$P_r(C_m) = \frac{1}{\mathbf{N}} \sum_{s_i \in C_m} 1 \tag{4}$$

$$P_r(C_m|C_n) = \frac{1}{|C_n|} \sum_{s_i \in C_m} \sum_{s_j \in C_n} \mathcal{T}(i-j) \tag{5}$$

where $\mathbf{N}$ is the total number of shots, $|C_n|$ is the number of shots in C_n, s_i is the ith shot ranked in time order, and $\mathcal{T}(x) = 1$ if $x = 1$, otherwise $\mathcal{T}(x) = 0$. The probability of a cluster $P_r(C_m)$ is directly proportional to the number of shots in C_m, while the probability of transitions $P_r(C_m|C_n)$ is directly proportional to the number of temporally adjacent pairs of shots from C_n to C_m.

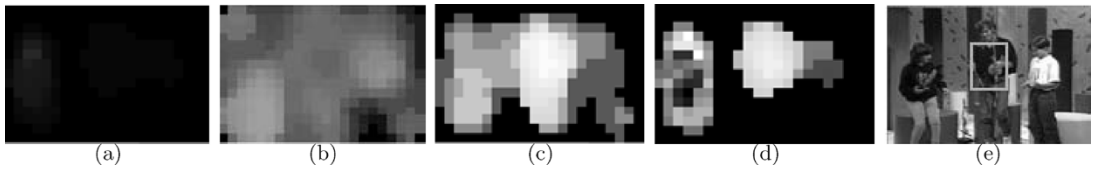


Fig. 4. Motion attention detection. (a) $\mathbf{I}$. (b) $\mathbf{C_s}$. (c) $\mathbf{C_t}$. (d) $\mathbf{MA}$. (e) Original video frame with the attention area marked by the bounded box.

IV. Scene Modeling

A temporal graph can be partitioned into scenes by analyzing the interconnectivity among clusters. Fig. 3 illustrates the temporal graph of a video that can be segmented into four scenes. Two important observations are:) two different scenes are connected by at most one edge and 2) each scene contains at least one cluster that locates along the shortest path from the starting scene to the ending scene.

Based on these observations, we can easily detect scene boundaries by the following steps:

- Compute the shortest path from the cluster that contains the first shot in a video to the cluster that contains the last shot. The weight of an edge is set to 1. Dijkstra's algorithm is employed to find the shortest path $\langle \hat{C}_1, \hat{C}_2, \ldots, \hat{C}_n \rangle$.
- Disconnect the edge from $\hat{C}_i$ to $\hat{C}_j$ if $i = j + 1$. If there does not exist any path that can traverse from $\hat{C}_i$ to $\hat{C}_j$ or vice versa, $\hat{C}_i$ and $\hat{C}_j$ belong to two different scenes.

The proposed approach is simple yet effective. It allows us to quickly discover and decompose the structure of a temporal graph. In fact, the clusters along the shortest path could be utilized directly for video skimming and summarization.

The idea of detecting scene changes in a temporal graph is similar to the idea of finding story units in a scene transition graph (STG) [23], except that the adopted algorithms are different. Since we adopt Dijkstra's algorithm [2], the time complexity of our approach is $O(n + e)$, where $n = |\mathbf{V}|$ is the number of nodes (or clusters), and $e = |\mathbf{E}|$ is the number of edges in a temporal graph $\mathbf{TG}$. In [23], the algorithm is not based on the search of the shortest path, but the analysis of a "label sequence" which is composed of a series of shots marked with their class labels. The time complexity is $O(2 \times \mathbf{N})$, where $\mathbf{N}$ is the total number of shots in a video. The relationship among n, e, and $\mathbf{N}$ is $n - 1 \leq e \leq \mathbf{N} - 1$. When the number of clusters is equal to the number of shots, we have $e+1 = n = \mathbf{N}$. Since most shots in a video are highly correlated (e.g., shots in a dialog scene), the numbers of clusters and edges in a temporal graph will usually much smaller than the number of shots, i.e., $n \ll \mathbf{N}$, $e \ll \mathbf{N}$. Thus, our algorithm is considered more efficient.

V. Motion Attention Model

Attention is a neurobiological term. It means the concentration of the mental powers upon an object after close or careful observation or listening. Computational attentional models have been studied in [7] and [8]. Motivated by these studies, we employ the motion attention model in [12] to compute the attention of humans when viewing videos. The model is based upon the manipulation of motion vector field (MVF) [11] that describes the spatial-temporal layout of motion vectors extracted directly from MPEG video streams. If we consider MVF as the retina of the eyes, the motion vectors can be the perceptual response of optic nerves. We assume that MVF is composed of three types of attention inductors: intensity inductor, spatial coherence inductor, and temporal coherence inductor. When the motion vectors go through these inductors, three maps (images) are generated to describe the responses of intensity, spatial, and temporal coherency. These maps are fused as a saliency map to model human attention.

A. Attention Inductors

Based on our assumption, there will be three inductors for each macro block in MPEG video frames. The intensity inductor $\mathbf{I}$ induces motion energy or activity. Denote (i, j) as the index of a macro block $\mathbf{I}$ is defined as

$$\mathbf{I}(i,j) = \frac{\sqrt{dx_{i,j}^2 + dy_{i,j}^2}}{\mathcal{Z}} \tag{6}$$

where $(dx_{i,j}, dy_{i,j})$ represents the motion vector in the macro block and $\mathcal{Z}$ is the maximum magnitude in MVF [11].

The spatial coherency inductor $\mathbf{C_s}$ and the temporal coherency inductor $\mathbf{C_t}$ are computed, respectively, by measuring the entropy of spatial and temporal phase distribution of motion vectors. For each macro block indexed by (i, j), a phase histogram $SH_{i,j}^W$ with a spatial window of size $W \times W$ is generated for $\mathbf{C_s}(i, j)$, while a phase histogram $TH_{i,j}^L$ with a temporal window of L frames is generated for $\mathbf{C_t}(i, j)$. Based on the phase histograms, $\mathbf{C_s}(i, j)$ and $\mathbf{C_t}(i, j)$ are computed as

$$\mathbf{C_s}(i,j) = -\sum_{k=1}^{n} P_s(k) \log(P_s(k)) \tag{7}$$

$$\mathbf{C_t}(i,j) = -\sum_{k=1}^{n} P_t(k) \log(P_t(k)) \tag{8}$$

where

$$P_s(k) = \frac{SH_{i,j}^W}{\sum_{l=1}^{n} SH_{i,j}^w(l)} \tag{9}$$

$$P_t(k) = \frac{TH_{i,j}^L}{\sum_{l=1}^{n} TH_{i,j}^w(l)} \tag{10}$$

and n is the number of histogram bins. The three inductors, $\mathbf{I}$, $\mathbf{C_s}$, and $\mathbf{C_t}$, together compose a motion perception system for attention modeling. Fig. 4(a)–(c) shows the outputs of three inductors on an image sequence.

B. Saliency Map

The outputs of the three inductors reciprocally characterize the spatio-temporal attributes of motion in a particular way. By observing the relationship between motion vectors and the attended motions, we have drawn some conclusions as follows. First, generally speaking, the motion with high intensity always attracts human attention. However, this may not be true for certain camera motions since they can also induce high intensity for $\mathbf{I}$ inductor. For instance, when a camera pans rapidly to track an object, the human probably pays attention to the tracked object only, even if this object sometime appears still when both the camera and the object move. Although $\mathbf{I}$ is not sensitive to the motion with lower energy, we can take advantage of the other inductors to compensate for the negative effects. Second, the spatial phases consistency provides us two cues. One is that the phases of motion vectors in a moving object tend to be consistent. The other is that, if the phases of motion vectors are disordered and the magnitudes of them are evidently large, it implies that the motion information is not reliable. Usually, the $\mathbf{C_s}$ inductor is sensitive to motion with lower intensity. Finally, since camera motion is generally more stable than object motion over a longer period of time, $\mathbf{C_t}$ can effectively exploit this property to discriminate object motion from camera motion. Based on these observations, we integrate and fuse the three inductors to form the motion attention model $\mathbf{MA}$ as

$$\mathbf{MA} = \mathbf{I} \times \mathbf{C_t} \times (1 - \mathbf{I} \times \mathbf{C_s}). \tag{11}$$

$\mathbf{MA}$ is represented as a saliency map, as shown in Fig. 4(d). Basically, a macro block with a high value of $\mathbf{MA}$ indicates that it locates at a high activity region induced by object motion. In principle, this model can highlight regions with object motion after the implicit compensation of camera motion through entropy information. In Fig. 4(d), the computed motion-attended regions are due to object motions.

After computing $\mathbf{MA}$, the regions of attention in a saliency map are located consecutively by histogram balance, media filtering, binarization, region growing, and region selection [12]. The number of located regions in each frame is restricted to at most three since it is hard for humans to focus on more than three objects simultaneously. Fig. 4(e) shows the located motion-attended region.

In our application, the attention value $\mathcal{A}$ of a frame is defined as the average value of $\mathbf{MA}$ in the located regions. The attention value of a shot (subshot) is defined as the average $\mathcal{A}$ of frames belong to that shot (subshot). Similarly, the attention value of a cluster (scene) is defined as the average $\mathcal{A}$ of shots (clusters) in that cluster (scene).

VI. Video Summarization

Summarization can be viewed as a process of selecting video segments based on the given criteria (e.g., entropy and perceptivity) and constraint (e.g., skim ratio). In our case, a video summary is generated directly from a temporal graph by exploiting its structural, entropy, and perceptual hints. The structural information provides a hierarchical way of selection, the entropy information inferred from prior probabilities provides cues on the selection of scenes and clusters, while the perceptual hints inferred from attention values facilitate the selection of the desired scenes, clusters, shots and subshots. In our approach, we adopt a top-down methodology to hierarchically summarize videos from the scene level, cluster level, shot level up to subshots level. Let $\mathcal{R}$ as the skim ratio of an original video. Our strategy is to discard approximately $1 - \mathcal{R}$ percentage of video frames by looking into their contribution, at each hierarchy, toward the entropy and perceptual importance of a final output video.

In our formulation, the entropy and perceptual information will jointly define the qualities of scenes and clusters, while the perceptivity will define the qualities of shot and subshot. The summarization is actually carried out by selecting the desired segments at each level, based on the constraint $\mathcal{R}$, in a recursive manner. The summarization terminates whenever the desired skim ratio $\mathcal{R}$ is attained. The detailed algorithms at different levels of hierarchies are carried out as follows.

A. At Scene-Level

- Let $\mathcal{Q}_i$ denote the quality of a scene S_i, where $\mathcal{Q}_i$ is computed as

$$\mathcal{Q}_i = \mathcal{Q}(S_i) = \frac{1}{\mathbf{N}_i} \sum_{C_j \in S_i} \mathrm{P}_r(C_j) \times \mathcal{A}(C_j) \tag{12}$$

  where $\mathrm{P}_r(C_j)$ and $\mathcal{A}(C_j)$ is, respectively, the prior probability and attention value of a cluster C_j, and $\mathbf{N}_i$ is the number of clusters in S_i. We discard those scenes whose $\mathcal{Q}_i$ is smaller than $\alpha \times \mu \times (1 - \mathcal{R})$, where μ is the average $\mathcal{Q}_i$ of all scenes, and α is a parameter to control the number of selected scenes. In the experiments, we set $\alpha = 0.01$ in order to retain most scenes, except those scenes with very low $\mathcal{Q}_i$. If the skim ratio is equal to $\mathcal{R}$, the algorithm will terminate.
- Sort the selected scenes for summarization in ascending order according to their value $\mathcal{Q}_i$.

B. At Cluster-Level

- Based on the sorted order of scenes, one scene S_i is picked up at a time.
- Let $\mathcal{QC}_j$ be the quality of a cluster C_j. For each S_i, sort its clusters in descending order according to

$$\mathcal{QC}_j = \frac{P_r(C_j) \times \mathcal{A}(C_j)}{Z} \tag{13}$$

  where $Z = \sum_{C_k \in S_i} P_r(C_k) \times \mathcal{A}(C_k)$. Based on the sorted order, a subset of clusters in S_i whose accumulated value satisfies

$$\begin{aligned} \sum_{C_j \in S_i} \mathcal{QC}_j &\geq \frac{\mathcal{Q}(S_i)}{\sum_k \mathcal{Q}(S_k)} + \mathcal{R} \times \frac{\mathcal{Q}(S_i)}{\sum_k \mathcal{Q}(S_k)} \\ &= (1 + \mathcal{R}) \times \frac{\mathcal{Q}(S_i)}{\sum_k \mathcal{Q}(S_k)} \end{aligned} \tag{14}$$

  will be selected while the remaining clusters will be discarded. The number of clusters selected in a scene is indeed directly proportional to its normalized scene quality,

TABLE II
DETAILS OF TEST VIDEOS

No.	Video	Genre	Sound Track	Scene	Shot	Time
1.	docon.mpg	Carton	Yes	14	209	11:41
2.	cm1002.mpg	Commercial	Yes (incl. music)	14	165	8:59
3.	hv1.mpg	Home video	No	29	98	20:14
4.	hv2.mpg	Home video	No	56	220	17:05
5.	hv3.mpg	Home video	No	44	127	10:40
Total	-	-	-	157	819	68:39

i.e., $\mathcal{Q}(S_i)/(\sum_k \mathcal{Q}(S_k))$, where $\sum_k \mathcal{Q}(S_k)$ is the summation of quality over all scenes. Here, to ensure robustness, we add an offset $\mathcal{R} \times (\mathcal{Q}(S_i)/(\sum_k \mathcal{Q}(S_k)))$ in (14) so that a few more clusters can be selected.

- If the skim ratio is equal to $\mathcal{R}$, the algorithm terminates. Otherwise, we pick up the next scene for investigation until all scenes are visited.

C. At Shot-Level

- Based on the sorted order of scenes and clusters, one cluster C_i is picked up at a time. For each cluster C_i, sort its shots s_j in descending order according to their attention values $\mathcal{A}(s_j)$. Based on the sorted order, a subset of shots whose accumulated value satisfies

$$\sum_{s_j \in C_i} \mathcal{A}(s_j) \geq (1 + \mathcal{R}) \times \mathcal{QC}_i \tag{15}$$

will be selected while the remaining shots will be discarded. Similar to (14), the number of shots selected in a cluster is directly proportional to its cluster quality $\mathcal{QC}_i$ plus an offset $\mathcal{R} \times \mathcal{QC}_i$. If the skim ratio is equal to $\mathcal{R}$, the algorithm terminates. Otherwise, we pick up the next cluster for investigation until all clusters are visited.

- Sort all of the selected shots in ascending order according to their attention values. Pick one shot at a time and only keep the subshot that has the largest attention value. If the skim ratio is equal to $\mathcal{R}$, the algorithm terminates. Otherwise, we pick up next shot until all shots are visited.

D. At Subshot-Level

- Based on the sorted order of shots, we discard one subshot at a time until the desired skim ratio is reached.

The aim of this algorithm is to maintain the content balance of scenes according to their probability of occurrence and attention values, while on the other hand, to hierarchically trim off those segments, from scenes down to subshots, that are comparatively less attended in order to achieve the desired skim ratio.

VII. EXPERIMENTS

We conduct experiments on five videos as shown in Table II. The first two videos that consist of sound tracks are from MPEG-7 video collection while the last three are home videos. We evaluate the performance of our proposed approach based on the results of scene detection and video summarization. Since the results of scene modeling can affect summarization, the first experiment assesses the recall and precision of the detected scene boundaries. The correct scene borders are manually identified by human subjects. Basically, a scene border is identified if there is a change of shooting site or story flow. The second experiment is based on subjective evaluation. Since the quality of a video summary is subject to human perception, we carry out a user study experiment to quantitatively evaluate the informativeness (content coverage) and the enjoyability (perceptual quality) of each machine-generated summary.

A. Scene Change Detection

We employ recall precision as the measure for performance evaluation. Let N_c be the number of correctly detected scenes, N_m the number of detected scenes by our approach, and N_h the number of scenes annotated by human subjects. The recall and precision is defined as

$$\text{recall} = \frac{N_c}{N_h}$$
$$\text{precision} = \frac{N_c}{N_m}.$$

The values of recall and precision is in the range of [0, 1]. A high recall indicates the capability of locating correct scenes, while a high precision indicates the capability of avoiding false matches.

Table III shows the experimental results of scene change detection. We compare the proposed approach with the method in [15]. Both approaches adopt the same algorithms for video partitioning and keyframe construction. For scene detection, the approach in [15] employs a time-constraint grouping algorithm to group similar shots. Basically, shots in one scene are progressively grouped until there is no similar shot found within a temporal distance. As shown in Table III, the proposed approach, on average, outperforms [15] in terms of recall and precision. One major deficiency of [15] is that the temporal distance parameter are set to a fixed value and used throughout a video. We found that this parameter, to be effective, should be adaptive from time to time depending on the content of a video. The proposed approach in this paper does not suffer from this problem since the temporal distance is embedded as part of the similarity measure as indicated in (3). The results of scene detection are dependent mainly on the accuracy of normalized cut and scene modeling.

Overall, the normalized cut algorithm and scene modeling based on the temporal graph generation perform satisfactorily.

TABLE III
RESULTS OF SCENE CHANGE DETECTION. C: CORRECT DETECTION, M: MISSED DETECTION, F: FALSE ALARM

	Proposed Approach					Approach in [15]				
No.	C	M	F	Recall	Precision	C	M	F	Recall	Precision
1.	14	0	2	1.00	0.88	10	4	3	0.71	0.77
2.	14	0	5	1.00	0.73	11	3	3	0.79	0.79
3.	25	4	2	0.87	0.93	27	2	10	0.93	0.73
4.	45	11	4	0.80	0.92	34	22	18	0.61	0.65
5.	36	8	4	0.82	0.90	29	15	9	0.66	0.76
Ave	-	-	-	0.90	0.87	-	-	-	0.74	0.74

By manually browsing all shots in the clusters that are constructed by the normalized cut, we found that most shots inside the same clusters are visually similar. Few exceptions occur when shots share similar color content but different semantic objects. The oversegmentation of clusters happens when there are changes of camera viewing angles and distances. Nevertheless, by analyzing the interconnectivity among clusters, our approach, in most cases, can correctly identify and group these clusters under same scene.

As indicated in Table III, our proposed scene detection approach achieves 100% recall for both videos *docon.mpg* and *cm1002.mpg*. In these two videos, the false alarms are mainly due to the changes of lighting conditions, shooting angles, and shooting distances in scenes. For instance, when the shooting distance changes from a long take shot (normally this is a master shot) to a close-up shot, the similarity between the two shots is small even though they are shooting the same site. These circumstances happen frequently especially for the commercial video *cm1002.mpg*; as a result, only 73% of precision is attained. For the last three home videos, besides the changes of lighting, shooting angles, and distances, false alarms are also due to the instability of camera motion which causes errors when keyframe construction is performed [15]. In addition to false alarms, the missed detections in home videos are mainly due to the similar color content of different outdoor scenes. As a result, different scenes are grouped together under one scene. Fig. 5 shows the 16 detected scenes of the video *docon.mpg*. Each image shown in the figure represents a scene. These images are selected from the shots with the highest attention values.

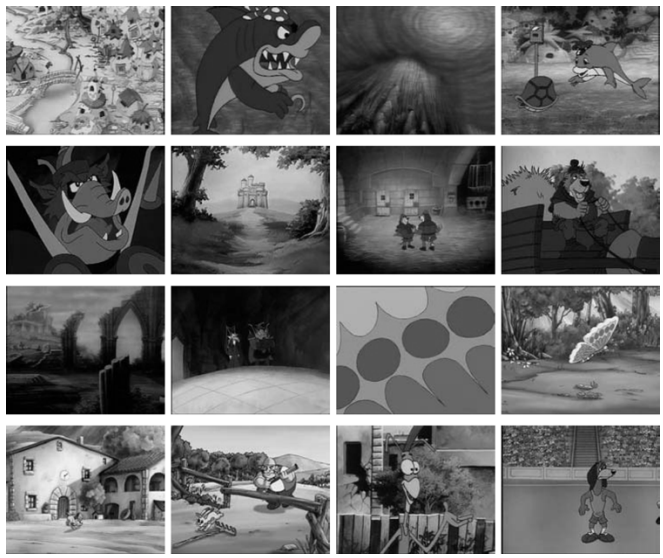


Fig. 5. Sixteen detected scenes in docon.mpg. Each image represents a scene. These images are selected from the shots with the highest attention values.

B. Video Summarization

To quantitatively investigate the performance of video summarization, two criterions, informativeness and enjoyability, are used for evaluation. Informativeness accesses the capability of maintaining content coverage while reducing redundancy. Enjoyability accesses the performance of the motion attention model in selecting perceptually enjoyable video segments for summaries. In this experiment, we generate ten summaries. Each tested video has two associated summaries, one with 10% of the original video length and the other with 25% of the original length. We invited 20 students to access the quality of these video summaries. The students watched the videos from high to low skim ratio, i.e., 10%, 25%, and then the original video (100%), in turn controlled by our evaluation tool. No fast forward or backward function is provided by this tool. After watching a video, a student is requested by the tool to assign two scores ranging from 0 to 100, in terms of informativeness and enjoyability, to the video before he or she can watch another video. To be fair, the students are also requested to give scores to the original videos in case they think that these videos are not informative or enjoyable. After watching an original video, the students are also given chance to modify the original scores assigned to the two associated summaries.

TABLE IV
PERFORMANCE EVALUATION OF VIDEO SUMMARIZATION FROM 20 STUDENTS

	Enjoyability			Informativeness		
No.	10%	25%	100%	10%	25%	100%
1.	68.35	77.85	93.10	64.55	77.35	92.85
	73.42	83.62	100	69.52	83.31	100
2.	66.75	76.80	94.30	68.10	80.95	94.90
	70.78	81.44	100	71.76	85.30	100
3.	63.10	71.15	91.10	61.35	72.75	92.15
	69.26	78.10	100	66.58	78.95	100
4.	64.80	74.75	90.00	67.70	76.85	91.80
	72.00	83.05	100	73.75	83.71	100
5.	56.10	65.95	84.08	63.10	73.10	90.00
	66.72	78.44	100	70.11	81.22	100
Average (%d)	70.44	80.93	-	70.34	82.50	-
Drop (%d)	29.56	19.07	-	29.66	17.50	-

Table IV shows the experimental results. Each nonshaded score is the average scores of 20 students, while each shaded score is the average of scores that are normalized by the scores assigned to the original video. The overall average scores shown at the bottom of the table are based on the mean of the normalized scores. As indicated in Table IV, the average scores for enjoyability are 70.44% and 80.93%, respectively, for video summaries of 10% and 25% skimming ratio. The average scores

TABLE V
STANDARD DEVIATION OF SCORES IN TABLE IV FROM 20 STUDENTS

	Enjoyability			Informativeness		
No.	10%	25%	100%	10%	25%	100%
1.	4.82	4.59	5.09	4.40	4.64	5.60
2.	5.02	4.67	4.85	4.59	4.72	4.97
3.	4.37	4.86	4.74	4.23	4.77	4.96
4.	4.77	4.38	4.67	4.48	4.25	4.59
5.	4.21	3.60	5.10	4.14	4.21	4.68

for informativeness are 70.34% and 82.50%, respectively. Compared to the scores given to the original videos, the enjoyability scores drop 29.56% and 19.07%, while the informative scores drop by 29.66% and 17.5%, respectively. Table V further shows the standard deviation of these scores for each tested video.

The experimental results are indeed encouraging. By reducing 90% of the original video content, the overall enjoyability and informativeness of a summaries drop only around 30%. By reducing 75% of the video content, the enjoyability and informativeness drop only around 20%. In overall, the scores of videos with sound track are higher than that of videos without sound track. This is not surprised since audio provides extra information, and most users feel enjoyable when the sound effect can appropriately simulate the visual rhythm effect. In this experiment, the scores of informativeness and enjoyability are fairly close. This result is interesting since it can be an indication that both criterions are closely correlated.

C. Speed Efficiency

Because all of the video analysis and processing are carried out directly in MPEG compressed domain, the proposed approach is reasonably fast. On a Pentium III platform, currently our motion attention model can run in real time. Excluding the time to detect shot boundaries and construct keyframes, our proposed approach took approximately 23 min to generate ten summaries for the five tested videos of approximately 69 min. In fact, most of the computational time is spent computing the adaptive threshold T_s mentioned in Section II by measuring the similarity among all pairs of shots. The normalized cut algorithm, which is traditionally slow when applied for image segmentation, however, is computationally efficient for our application. This is mainly because the number of shots in a 1-h video is typically less than 1000, far less than the number of pixels in an image.

VIII. CONCLUSION

We have presented a novel approach for video summarization. On the one hand, the structure of videos is exploited in order to maintain the content coverage of summaries. On the other hand, a motion attention model is adopted to compute the perceptual quality of video segments for content highlight selection. Information for both video structure and highlight are then effectively encapsulated in a temporal graph. By modeling the evolution of a video through a temporal graph, the proposed approach can automatically detect scene changes and generate summaries.

In the future, we will focus our research on the video attention model. Besides motion, multimedia information such as audio, music, and video captions will be taken into consideration for a more effective selection of video segments. Automatic video editing techniques will also be developed for the composition of the selected segments for video summarization.

REFERENCES

[1] H. S. Chang, S. S. Sull, and S. U. Lee, "Efficient video indexing scheme for content-based retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 8, pp. 1269–1279, Dec. 1999.

[2] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*. New York/Cambridge, MA: McGraw-Hill/MIT Press, 1990.

[3] D. DeMenthon, V. Kobla, and D. Doermann, "Video summarization by curve simplification," in *Proc. 6th ACM Int. Conf. Multimedia*, 1998, pp. 211–218.

[4] Y. H. Gong and X. Liu, "Video summarization using singular value decomposition," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2000, pp. 174–180.

[5] A. Hanjalic and H. J. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 8, pp. 1280–1289, Dec. 1999.

[6] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video retrieval systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 5, pp. 580–88, Jun. 1999.

[7] L. Itti and C. Koch, "Computational modeling of visual attention," *Nature Rev. Neurosci.*, vol. 2, no. 3, pp. 194–203, Mar. 2001.

[8] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[9] J. Nam and A. T. Tewfik, "Dynamic video summarization and visualization," in *Proc. 7th ACM Int. Conf. Multimedia*, 1999, pp. 53–56.

[10] R. Lienhart, "Dynamic video summarization of home video," *SPIE*, vol. 3972, pp. 378–389, Jan. 2000.

[11] Y. F. Ma and H. J. Zhang, "A new perceived motion based on shot content representation," in *Proc. Int. Conf. Image Process.*, vol. 3, 2001, pp. 426–429.

[12] ——, "A model of motion attention for video skimming," in *Proc. Int. Conf. Image Process.*, vol. 1, 2002, pp. 129–132.

[13] Y. F. Ma, L. Lu, H. J. Zhang, and M. Li, "A user attention model for video summarization," in *Proc. 10th ACM Int. Conf. Multimedia*, 2002, pp. 533–542.

[14] C. W. Ngo, T. C. Pong, and R. T. Chin, "Video partitioning by temporal slices coherency," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 11, no. 8, pp. 941–953, Aug. 2001.

[15] C. W. Ngo, T. C. Pong, and H. J. Zhang, "Motion-based video representation for scene change detection," *Int. J. Comput. Vis.*, vol. 50, no. 2, pp. 127–143, 2002.

[16] ——, "Motion analysis and segmentation through spatio-temporal slices processing," *IEEE Trans. Image Process.*, vol. 12, no. 3, pp. 341–355, Mar. 2003.

[17] X. Orriols and X. Binefa, "An EM algorithm for video summarization, generative model approach," in *Proc. Int. Conf. Comput. Vis.*, vol. 2, 2001, pp. 335–342.

[18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

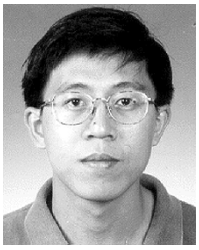
[19] M. A. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding techniques," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 1997, pp. 775–781.

[20] H. Sundaram and S. F. Chang, "Determining computable scenes in films and their structure using audio-visual memory models," in *Proc. 8th ACM Int. Conf. Multimedia*, 2000, pp. 95–104.

[21] H. Sundaram, L. Xie, and S.-F. Chang, "A utility framework for the automatic generation of audio-visual skims," in *Proc. 10th ACM Int. Conf. Multimedia*, 2002, pp. 189–198.

[22] N. Vasconcelos and A. Lippman, "A spatio-temporal motion model for video summarization," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, 1998, pp. 361–366.

[23] M. M. Yeung and B. L. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 5, pp. 771–785, Oct. 1997.

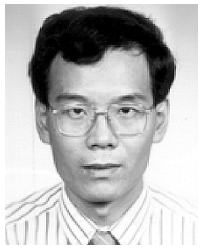


Chong-Wah Ngo (M'02) received the B.Sc. and M.Sc. degrees in computer engineering from Nanyang Technological University, Singapore, in 1996 and 1994, respectively, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2000.

Since 2002, he has been an Assistant Professor with the City University of Hong Kong (CityU). Before joining CityU, he was a Postdoctoral Visitor with the Beckman Institute, University of Illinois at Urbana-Champaign, and was a Research Associate with HKUST. He was with Microsoft Research China as a Visiting Researcher during the summer of 2002 and a summer intern in 1999. He was also with the Information Technology Institute, Singapore, in 1996. His current research interests include image and video indexing, computer vision, and pattern recognition.

Yu-Fei Ma (M'00) received the B.S. degree from Harbin Engineering University, Harbin, China, in 1994 and the M.S. degree in computer science from Tsinghua University, Tsinghua, China, in 2000.

He joined Microsoft Research Asia, Beijing, China, in 2000 and is currently an Associate Researcher with the Multimedia Computing Group. His current research interests are in video content analysis, image processing, and pattern recognition. He has published a number of papers in these fields. From 1994 to 1997, he was engaged in computer network system analysis as a System Engineer.



Hong-Jiang Zhang (M'91–SM'97–F'03) received the B.S. degree from Zhengzhou University, Henan, China, in 1982 and the Ph.D. degree from the Technical University of Denmark, Lyngby, in 1991, both in electrical engineering.

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. From 1995 to 1999, he was a Research Manager with Hewlett-Packard Laboratories, Palo Alto, CA, where he was responsible for research and technology transfers in the areas of multimedia management an intelligent image processing. In 1999, he joined Microsoft Research Asia, Beijing, China, where he is currently the Managing Director of the Advanced Technology Center. He has coauthored/coedited four books, over 300 referred papers, eight special issues of international journals on image and video processing, content-based media retrieval, and computer vision, as well as over 50 patents or pending applications. He currently serves on the editorial boards of five IEEE/ACM journals and a dozen committees of international conferences. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON MULTIMEDIA.