Automatic Video Summarization by Graph Modeling

Chong-Wah Ngo[*], *Yu-Fei Ma*[‡] *& Hong-Jiang Zhang*[‡]

[*]Department of Computer Science
City University of Hong Kong
cwngo@cs.cityu.edu.hk

[‡]Microsoft Research Asia
Haidian District, Beijing, PRC
{yfma,hjzhang}@microsoft.com

Abstract

We propose a unified approach for summarization based on the analysis of video structures and video highlights. Our approach emphasizes both the content balance and perceptual quality of a summary. Normalized cut algorithm is employed to globally and optimally partition a video into clusters. A motion attention model based on human perception is employed to compute the perceptual quality of shots and clusters. The clusters, together with the computed attention values, form a temporal graph similar to Markov chain that inherently describes the evolution and perceptual importance of video clusters. In our application, the flow of a temporal graph is utilized to group similar clusters into scenes, while the attention values are used as guidelines to select appropriate sub-shots in scenes for summarization.

1. Introduction

Techniques in automatic video summarization, in broad, can be categorized into two major approaches: static storyboard summary [1, 2, 3, 17] and dynamic video skimming [9, 7, 4, 15]. The former is a collection of static keyframes of video shots, while the latter is a shorter version of video that composed of a series of selected video clips. Static storyboard allows non-linear browsing of video content by sacrificing the temporal evolution of a video. Dynamic video skimming, in contrast, preserves time-evolving nature of video by linearly and continuously browsing certain portion of video content depending on a given time length. For both approaches, the appropriate selection of video segments plays a major rule in maximizing the entropy information and perceptual quality of a video summary.

To date, compared with static storyboard summary, there are relatively few works being addressed for dynamic video skimming. Techniques for dynamic video skimming include applying EM [8], SVD [9], motion model [10, 16] and semantic analysis [7, 15, 11]. Most techniques are based mainly on visual information except approaches like [7, 15] where audio and linguistic information are also incorporated in order to derive semantic meaning. In [7], 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 summary may not be useful in revealing the content coverage. In [15], the InfoMedia system was developed to generate short synopsis of 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.

Recently, singular value decomposition (SVD) emerges as an attractive computational model for video summarization [9]. However, this approach is computationally intensive since it operates directly on video frames. In [4], an hierarchical tree that consists of events, activities, actions and shots is constructed for each video. Then a summary is generated by randomly removing subtrees at different levels to meet the output video length. Other sophisticated mathematical models include [8, 16]. However, these models are only applied to single video shot. It is unclear how to extend their works to summarize an entire video.

Most existing approaches emphasize either content coverage [9, 4] or perceptual quality (highlight) [7, 10, 11]. In this paper, we propose a unified approach for dynamic video skimming that emphasizes both content coverage and perceptual quality, in addition, reduces content redundancy. To measure perceptual quality, a motion attention model is employed to model human's attention when viewing a video. To maintain content balance and reduce redundancy, a video is structured according to scenes, clusters, shots and subshots in a hierarchical tree. The selection of video clips for summarization is based on the probability of sub-trees and their attention values.

1.1. 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

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

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 algorithms of [12, 13] to temporally partition videos into shots and then into sub-shots. We also apply the adaptive keyframe selection and construction scheme proposed in [13] to select/construct one keyframe for each sub-shots. These keyframes are used for shot similarity measure by normalized cut algorithm to obtain clusters. The similarity measure is based on the video representation techniques given in [13].

1.2. Overview of Our Approach

Figure 1 illustrates the flow of our proposed approach. The whole process is carried out in MPEG compressed domain. Initially, a complete undirected weighted graph is constructed to model the similarity among all pairs of shots in a video. We employ a global criterion, normalized cut [14], to optimally decompose the graph into sub-graphs (clusters). Meanwhile, based on the MPEG motion vector flow field, a motion attention model is utilized to compute the perceptual attention of video shots. The computed attention values and the partitioned sub-graphs subsequently 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, through the shortest path algorithm, is utilized to analyze and detect scene transitions. Once scene changes are detected, video structure is constructed hierarchically in the form of scenes, clusters, shots, sub-shots and keyframes. A summary is then generated in a top-down manner. 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.

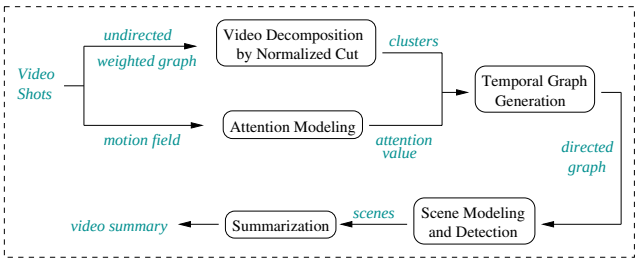

Figure 1: Proposed approach

2. Video Decomposition

A video is initially represented as a weighted undirected graph that composes of shots. Let $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ denotes 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, normalized cut algorithm [14] 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, B ($A \cup B = \mathbf{V}$), by removing edges between A and B. Mathematically we have

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

where $cut(A,B) = \sum_{i \in A, j \in B} w(i,j)$ is a cut value, and $assoc(A,\mathbf{V}) = \sum_{i \in A, j \in \mathbf{V}} w(i,j)$ 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 minimize $Ncut$. Eqn (1) can be transformed into a standard eigen system

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

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:

- 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

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

  which takes into account the similarity, $Sim(i,j)$, 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 out experiment, k is set to 8. The normalization constant $\mathbf{T}$ is the total frame numbers in a video.

- Solve Eqn (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 sub-graphs 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 is, respectively, the average and standard deviation of shot similarity between all pairs of shots in a given video.

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

3 Temporal Graph Generation

Once $\mathbf{G}$ is partitioned into sub-graphs, a set of clusters that consist of temporally adjacent or non-adjacent shots is obtained. The temporal relationship among these clusters can be constructed to form a temporal graph by adding the time order information of video shots. 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 Figure 2 for an illustration of temporal graph.

Formally, a cluster C_m transits to another cluster C_n if there exists a shot in C_m that is temporally adjacent to a shot in C_n. Each cluster is modeled by two parameters: its prior probability $\mathrm{P}_r(C_m)$ and attention value $\mathcal{A}(C_m)$, while every pair of clusters is modeled by a transition probability $\mathrm{P}_r(C_m|C_n)$. Mathematically, they are computed by

$$\mathrm{P}_r(C_m) = \frac{1}{\mathbf{N}} \sum_{s_i \in C_m} 1 \quad (4)$$

$$\mathrm{P}_r(C_m|C_n) = \frac{1}{|C_n|} \sum_{s_i \in C_m} \sum_{s_j \in C_n} \mathcal{T}(i-j) \quad (5)$$

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

4 Scene Modeling

A temporal graph can be partitioned into scenes by analyzing the inter-connectivity among clusters. Figure 2 illustrates the temporal graph of a video that can be segmented into four scenes. Two important observations are: i) two different scenes are connected by at most one edge; ii) 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 detect scene boundaries by

- 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 $< \hat{C}_1, \hat{C}_2, \ldots, \hat{C}_n >$.
- Disconnect the edge from $\hat{C}_i$ to $\hat{C}_j$ if $i = j+1$. If there does not exist any path that traverses from $\hat{C}_i$ to $\hat{C}_j$ or vice verse, $\hat{C}_i$ 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.

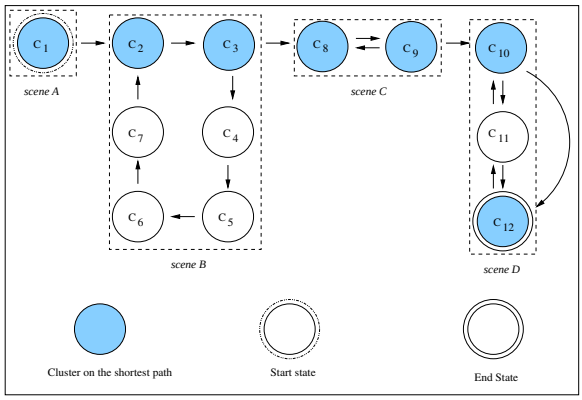

Figure 2: Temporal graph and scene change detection.

5 Motion Attention Model

Attention is a neurobiological term. It means the concentration of the mental powers upon an object after a close or careful observation or listening. Computational attentional models have been studied in [5, 6]. Motivated by these studies, we employed the motion attention model in [10] to compute the attention of human when viewing videos. Mathematically, the motion attention model $\mathbf{MA}$ [10] of a frame is defined as

$$\mathbf{MA} = \mathbf{I} \times \mathbf{C_t} \times (1 - \mathbf{I} \times \mathbf{C_s}) \quad (6)$$

where $\mathbf{I}$ is an intensity inductor, $\mathbf{C_t}$ is a temporal coherency inductor and $\mathbf{C_s}$ is a spatial coherence inductor. $\mathbf{I}$ induces motion activity, while $\mathbf{C_s}$ and $\mathbf{C_t}$ induce respectively the spatial and temporal phase consistency of MPEG motion vectors. The intuition of this model is to highlight regions with object motion after the implicit compensation of camera motion through entropy information. The details of this model can be found in [10].

After the $\mathbf{MA}$ of a frame is computed, the regions of attention are detected subsequently by histogram balance, media filtering, binarization, region growing and region selection [10]. The number of detected regions in each frame is restricted to at most three since it is hard for human to focus for more than three objects simultaneously. The attention value $\mathcal{A}$ of a frame is defined as the average value of $\mathbf{MA}$ in the detected regions.

In our application, the attention value of a shot (sub-shot) is defined as the average $\mathcal{A}$ value of frames belong to that shot (sub-shot). Similarly, the attention value of a cluster

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

Table 1: Test Videos.

(scene) is defined as the average attention value of shots (clusters) in that cluster (scene).

6. Video Summarization

A shorter version of video could be generated directly from a temporal graph characterized by its prior probabilities and attention values. Temporal graph provides both structural and perceptual hints in selecting useful segments for reproducing a shorter yet enjoyable and informative video. In our approach, we adopt a top-down methodology to automatically summarize a video from scenes, clusters, shots to subshots based on the content of a temporal graph.

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 towards the entropy and perceptual importance of a final video. The algorithm is carried out as follows

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

$$\mathcal{Q}_i = \frac{1}{\mathbf{N}_i} \sum_{C_j \in S_i} \mathrm{P}_r(C_j) \times \mathcal{A}(C_j) \tag{7}$$

  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 $0.01 \times \mu \times (1-\mathcal{R})$, where μ is the average $\mathcal{Q}_i$ of all detected scenes. If the skim ratio is equal to $\mathcal{R}$, the algorithm will terminate.

- Sort the remaining scenes in ascending order according to their value $\mathcal{Q}_i$, and similarly, sort all clusters in a scene individually according to the value $\mathcal{QC}_j = \frac{\mathrm{P}_r(C_j) \times \mathcal{A}(C_j)}{Z}$, where $Z = \sum_{C_k \in S_i} \mathrm{P}_r(C_k) \times \mathcal{A}(C_k)$.

- Based on the sorted order, one scene S_i is picked up at a time. We discard some clusters C_j in S_i, in ascending order, whose accumulated value satisfies

$$\sum_{C_j \in S_i} \mathcal{QC}_j < (1+\mathcal{R}) \times \frac{\mathcal{Q}(S_i)}{\sum_k \mathcal{Q}(S_k)} \tag{8}$$

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

- Sort all the remaining clusters in ascending order according to their value $\mathcal{QC}_j$, and similarly, sort all shots in every cluster individually according to their attention values.

- Based on the sorted order, one cluster C_i is picked up at a time. We discard some shots S_j in C_i, in ascending order, whose accumulated value satisfies

$$\sum_{S_j \in C_i} \mathcal{A}(S_j) < (1+\mathcal{R}) \times \mathcal{QC}_i \tag{9}$$

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

- Sort all 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.

- 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.

7. Experiments

We conduct experiments on five videos as shown in Table 1. 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 decomposition (Section 4) 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. Basically, a scene border is identified if

there is a change of shotting 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 carried out a user study experiment to quantitatively evaluate the informativeness (content coverage) and the enjoyability (perceptual quality) of each machine generated summary.

7.1 Scene Change Detection

Table 3 shows the experimental results of scene change detection. As indicated in Table 3, our 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, shotting angles and shotting distances in scenes. For instance, when the shotting distance changes from a long take shot (normally this is a master shot) to a close up shot, the similarity between two shots is small even though they are shot in a 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, shotting angles and distances, false alarms are also due to the instability of camera motion which causes errors when keyframe construction is performed [13]. In addition to false alarms, the missed detection is mainly due to the similar color content of different outdoor scenes. This causes different scenes to be grouped together.

7.2 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 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, while the other one with 25% of the original length. We invited twenty 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 $[0, 100]$, in term of informativeness and enjoyability, to the video before he or she can continue to 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 completing watching an original video, the students are also given chance to modify the original scores assigned to the two associated summaries.

No.	C	M	F	Recall	Precision
1.	14	0	2	1.00	0.88
2.	14	0	5	1.00	0.73
3.	25	4	2	0.87	0.93
4.	45	11	4	0.80	0.92
5.	36	8	4	0.82	0.90
Ave	-	-	-	0.90	0.87

Table 3: Results of scene change detection. C: Correct detection, M: missed detection, F: false alarm.

Table 2 shows the experimental results. Each non-shaded score is the average scores of twenty 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 normalized scores. As indicated in Table 2, the average scores for enjoyability are 70.44% and 80.93%, respectively, for video summaries of 10% and 25% skimming ratio. The average scores 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 4 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 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 simulate the visual rhythm effect. 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.

8. Conclusion

We have presented a novel approach for video summarization. On 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 temporal graph, the proposed approach can automatically detect scene changes and generate summaries.

No.	Enjoyability			Informativeness		
	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 (%)	70.44	80.93	-	70.34	82.50	-
Drop (%)	29.56	19.07	-	29.66	17.50	-

Table 2: Performance evaluation of video summarization from twenty students.

No.	Enjoyability			Informativeness		
	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

Table 4: Standard deviation of scores in Table 2.

Acknowledgments

The work described in this paper is fully supported by a RGC Grant CityU1072/02E (Project No. 9040693).

References

[1] H. S. Chang, S. S. Sull & S. U. Lee, "Efficient Video Indexing Scheme for Content-based Retrieval", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, No. 8, Dec 1999.

[2] D. DeMenthon, V. Kobla & D. Doermann, "Video Summarization by Curve Simplification," *ACM Multimedia*, 1998.

[3] A. Hanjalic & H. J. Zhang, "An Integrated Scheme for Automated Video Abstraction based on Unsupervised Cluster-validity Analysis," *IEEE Trans on Circuits and Systems for Video Technology*, Vol. 9, No. 8, pp. 1280-1289, Dec 1999.

[4] R. Lienhart, "Dynamic Video Summarization of Home Video," *SPIE: Storage and Retrieval for Media Database*, Vol. 3972, Jan 2000.

[5] L. Itti & C. Koch, "Computational Modeling of Visual Attention," *Nature Reviews Neuroscience*, Vol. 2, No. 3, pp. 194-203, Mar, 2001.

[6] L. Itti, C. Koch & E. Niebur, "A Model of Saliency-based Visual Attention for Rapid Scene Analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1998.

[7] J. Nam, A. T. Tewfik, "Dynamic Video Summarization and Visualization" *ACM Multimedia*, 1999.

[8] X. Orriols & X. Binefa, "An EM Algorithm for Video Summarization, Generative Model Approach," *Int. Conf. on Computer Vision*, 2001.

[9] Y. H. Gong & X. Liu, "Video Summarization Using Singular Value Decomposition," *Int. Conf. CVPR*, 2000.

[10] Y. F. Ma & H. J. Zhang, "A Model of Motion Attention for Video Skimming," *Int. Conf. on Image Processing*, 2002.

[11] Y. F. Ma, L. Lu, H. J. Zhang & M. Li, "A User Attention Model for Video Summarization," *ACM Multimedia*, 2002.

[12] C. W. Ngo, T. C. Pong & H. J. Zhang, "Video Partitioning through Temporal Slices Analysis", *IEEE Trans. on Circuit and Sys. for Video Technol.*, vol. 11, no. 8, pp.941-953, 2001.

[13] C. W. Ngo, T. C. Pong & H. J. Zhang, "Motion-Based Video Representation for Scene Change Detection," *Int. Journal of Computer Vsion*, 2002.

[14] J. Shi & J. Malik, "Normalized Cuts and Image Segmentation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 888-905, Aug, 2000.

[15] M. A. Smith & T. Kanade, "Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques," *Int. Conf. on Computer Vision and Pattern Recognition*, 1997.

[16] N. Vasconcelos & A. Lippman, "A Spatio-temporal Motion Model for Video Summarization", *Int. Conf. on CVPR*, 1998.

[17] M. M. Yeung & B. L. Yeo, "Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 7, No. 5, Oct 1997.