VIDEO SUMMARIZATION BY VIDEO STRUCTURE ANALYSIS AND GRAPH OPTIMIZATION

Shi Lu, Irwin King and Michael R. Lyu

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong SAR
{slu, king, lyu}@cse.cuhk.edu.hk

ABSTRACT

In this paper, we propose a novel video summarization method that combines video structure analysis and graph optimization. First, we analyze the structure of the video, find the boundaries of video scenes, then we calculate each scene's skimming length based on its structure and content entropy. Second, we define a spatial-temporal dissimilarity function between video shots and model each video scene as a graph, then find each scene's optimal skimming in the graph with dynamic programming. Finally, the whole video's skimming is obtained by concatenating the skimmings of the scenes. Experimental results show that our approach preserves the scene level structure and ensures balanced coverage of the major contents of the original video.

1. INTRODUCTION

The amount of video data on today's network is growing rapidly, thus giving rise to serious problems for efficient video browsing and management. To solve these problems, *video summarization*, which engages in creating a short summary of the original video to present its major content, has received more and more attentions in these days. Basically there are two kinds of video summaries: *static video summary*, which is composed of a set of salient images extracted or synthesized from the original video, and *dynamic video skimming*, which is a shorter version of the original video made up of several short video clips.

Compared with the static video summary, the dynamic video skimming makes more sense and is more attractive to the user. Recently much work has been conducted on dynamic video skimming generation. The Informedia system [1] selects the video segments according to the occurrence of important keywords in the corresponding caption text. Later work employs perceptually important features to summarize video. In [2] the authors construct a user attention curve to simulate the user's attention toward different video contents. [3] proposes a utility function for each video shot, and video skimmings are generated by utility maximization. [4] assigns different weight scores on several important features of the video then selects the video skimming that maximizes the feature score summation. [5] analyzes video structure by graph modelling then the video skimming is generated according to this structure and the motion attention values for video shots.

Edited videos have their intrinsic structures. In [6], the video is decomposed into a hierarchical V-TOC (Video Table Of Contents) tree structure. In [7], a scene transition graph is constructed by video shot clustering. [5] uses a graph to model the video and obtains the video structure by normalized graph cut.

A video summary should be able to cover the major video contents with balance. Although many video skimming generation techniques have been proposed, few of them have stressed on preserving the structure of the video. In this paper, we describe a novel video summarization approach that combines video structure analysis and graph optimization. We analyze the structure of the original video, find the scene boundaries, and determine each video scene's target skim length. We then model each scene into a graph, create video skimming for each video scene with graph optimization, and concatenate each local video skimming to obtain the final video skimming. Our approach preserves balanced structural coverage for the major video contents.

The rest of the paper is organized as follows. In Section 2 we describe our video summarization method. In Section 3 we show experimental results and make some discussions. Finally, in Section 4 we draw a conclusion and discuss our future work.

2. VIDEO SUMMARIZATION PROCEDURE

2.1. Video structure analysis

A video narrates a story just like an article does. From a narrative point of view, a video is composed of several video scenes $\{Sc_1 ... Sc_n\}$, each of which depicts an event like a

0-7803-8603-5/04/$20.00 ©2004 IEEE.

paragraph does in the articles. A video scene is composed by several semantic-related video shots $\{sh_1...sh_n\}$, each of which is an unbroken image sequence captured continuously by a camera. A video shot's role is just like a sentence in articles. The visual content of a video shot can be represented by its key frames. A video shot group Sg_i is the intermediate entity between video scenes and video shots, which is composed of several visually similar and temporally adjacent video shots. Thus from top to down, a video has a 4-level hierarchical structure: Video, Video scenes, Video shot groups, and Video shots [6]. Fig. 1 shows the hierarchical structure of a video.

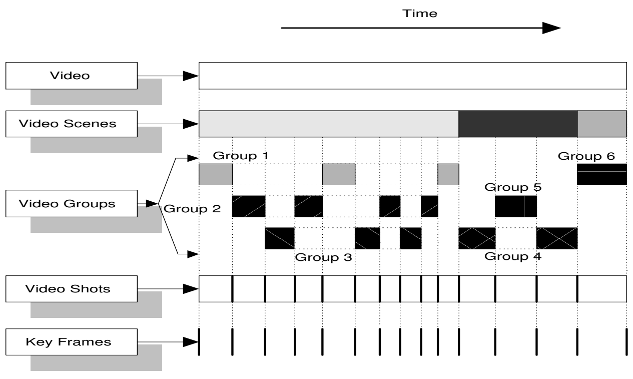

Fig. 1. Hierarchical video structure

In the remaining part of this paper, we use l_{sh_i}, l_{Sg_j} and l_{Sc_i} to represent the length of video shot sh_i, video shot group Sg_j, and video scene Sc_i, which is the total number of images containing in them respectively.

The structure of video is built in a bottom-up manner. First, we decompose the original video into video shots. Our shot detection method is similar to the method in [8], while we improve the filtering step for more accuracy. Then we continue to build up the hierarchical structure with the windowed-sweeping method in [6].

The detected video scenes can be classified into two types: *loop scenes* and *progressive scenes*, as shown in Fig 2. A loop scene is composed of more than one video shot groups, while a progressive scene is composed of a series of dissimilar video shots. Loop scenes are often used to depict an event happening in a place that needs detailed description, e.g., a conversation, while the progressive scenes are often used to depict changes between two events. Normally the loop scenes contain more important contents that need repeated illustrations.

Obviously, longer and more complex video scenes should be more important. For progressive scenes, we simply use their length to measure their importance. For loop scenes, however, since they are composed of several video shot groups, we define the content entropy of a scene Sc_i as:

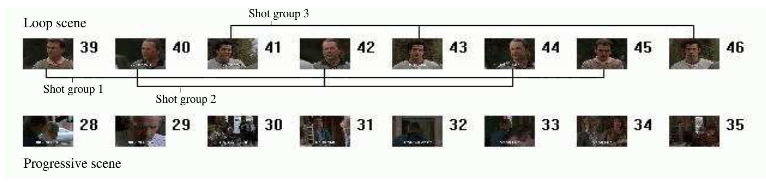

Fig. 2. Example of loop and progressive scenes

$$Entropy(Sc_i) = \sum_{Sg_j \in Sc_i} -\frac{l_{Sg_j}}{l_{Sc_i}} \log_2(\frac{l_{Sg_j}}{l_{Sc_i}}).$$

With the above definition, given the target video skimming length L_{vs} and the length of the video L_v, the skim ratio r_s is thus $\frac{L_{vs}}{L_v}$. We determine the skim length Sl of each scene and each group in the video as follows:

1. For each progressive scene Sc_x, $Sl_x = l_{Sc_x} \times r_s$. If S_x is less than the preset threshold t_1, we discard this progressive scene as too short skim does not make sense to people.

2. Suppose that after the first round, the left skim length is L'_{vs}, for the loop scenes $\{Sc_1...Sc_n\}$, $Sl_i = L'_{vs} \times \frac{Entropy(Sc_i)\times l_{Sc_i}}{\sum_{j=1}^{n} Entropy(Sc_j)\times l_{Sc_j}}$. In a similar manner, we discard Sc_i if Sl_i is less than the preset threshold t_2.

3. For the remaining loop scenes $\{Sc'_1...Sc'_m\}$, we set $Sl_i = L'_{vs} \times \frac{Entropy(Sc'_i)\times l_{Sc'_i}}{\sum_{j=1}^{m} Entropy(Sc'_j)\times l_{Sc'_j}}$.

The above skim length assignment algorithm ensures that more important scenes are assigned with more skim length.

2.2. Graph modelling and optimization

With each scene's skimming length determined, we need to select several video shots according to the skim length of each video scene and generate the final skimming. The selected video shots should cover both the visual diversity and the temporal distribution of the original video scene. To achieve this, we model each video scene into a graph based on the video shots contained in it, then we select the optimal skimming by performing optimization on that graph.

The spatial-temporal dissimilarity function between two video shots is defined as:

$$Dis(sh_i, sh_j) = 1 - VisualSim(sh_i, sh_j) \times e^{-k\times d_T(sh_i,sh_j)},$$

Here $VisualSim(sh_i, sh_j)$ can be any similarity measure between video shots, and here we use the color histogram correlation between video shot key frames. $d_T(sh_i, sh_j)$ is the temporal distance between the middle frames of video shot sh_i and sh_j, in terms of frame number. k is the parameter to control the slope of the exponential function, also in terms of frame number. To allow for a good coverage of both the visual and temporal contents of the video scene, we define the dissimilarity function such that it changes linearly with the visual similarity, but exponentially with the temporal distance.

With the shot pairwise spatial-temporal dissimilarity function, we define the spatial-temporal relation graph as follows:

The spatial-temporal relation graph $G(V, E)$ is a graph defined on a video shot set $S_{sh} = \{sh_1, ...sh_n\}$ such that:

1. $G(V, E)$ is a complete graph.
2. Each vertex v_i in the vertex set V is corresponding to a video shot sh_i in S_{sh} and vise versa. On each v_i there is a weight w_i which is equal to the length of video shot sh_i.
3. On each edge e_{ij}, there is an edge weight $w_{e_{ij}}$ which is equal to the spatial-temporal dissimilarity function $Dis(sh_i, sh_j)$ between video shots sh_i and sh_j. The direction of edge e_{ij} is from the earlier shot to the later video shot. Thus G is acyclic.

A simple example of the spatial-temporal relation graph on a scene is shown in Fig. 3.

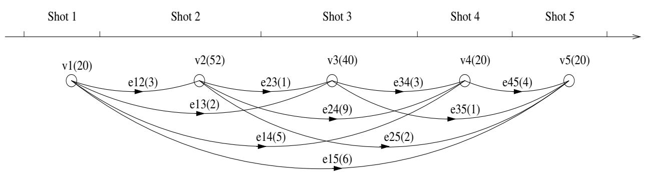

Fig. 3. Spatial temporal dissimilarity graph on 5 shots

Given the target skimming length L_{vs}, we can search a path in the spatial-temporal graph then use the video shots in that path as the video skimming for the video shot set. A path $p = \{v_{x_1}, ...v_{x_n}\}$ in the spatial-temporal graph consists of a set of video shots $\{sh_{x_1}, ...sh_{x_n}\}$, which is a video skimming whose total length is the summation of the weights on the vertexes $v_{x_1}, ...v_{x_n}$ in the path. We use $VWS(p_i)$ to represent the vertex weight summation of the path p_i. The length of the path is the summation of the spatial-temporal dissimilarity function between consecutive video shot pairs.

For this path p_s, we have two goals to meet: first, we want to maximize the length of the path L_{p_s}, which is the summation of dissimilarity function between consecutive video shots; second, we want $VWS(p_s)$ to be as close to L_{vs} as possible, but not to exceed it. We combine these two goals in the objective function f_{obj}, described in the following video skimming generation problem:

Problem 2.1 *Given a set of video shots $S_{sh} = \{sh_1...sh_n\}$, the spatial-temporal graph $G(V, E)$ built on S_{sh}, the target video skimming length L_{vs}, and a weight parameter w, search the path $p_s = \{v_{s_1}...v_{s_n}\}$ such that it maximizes the object function $f_{obj}(p_s, L_{vs}) = L_{p_s} + w \times (VWS(p_s) - L_{vs})$, and $VWS(p_s) \leq L_{vs}$.*

2.3. Solution and algorithm

Problem 2.1 is a constrained optimization problem. Brute force searching is feasible but inefficient; however, the problem has an *optimal substructure* [9] and can be solved with dynamic programming.

Suppose there are n video shots in the video shot set. We add a virtual vertex v_0 such that $w_0 = 0$ and $w_{e_{0j}} = 0$ for all $0 < j \leq n$. We use $p^i_{v_x,l_r} = \{v_x, ...\}$ to denote a path in the spatial-temporal relation graph such that it begins with vertex v_x, and its vertex weight summation is upper bounded by l_r. We then use $p^o_{v_x,l_r}$ to denote the optimal path among all such paths, which means $f_{obj}(p^o_{v_x,l_r}) = \max_i f_{obj}(p^i_{v_x,l_r})$. Thus $p^o_{v_0,L_{vs}}$ is the path we want to find.

Then we have the following optimal substructure:

1. $f_{obj}(p^o_{v_n,l_r}) = w \times (l_{sh_n} - L_{vs})$, for all $l_r \leq L_{vs}$;
2. $f_{obj}(p^o_{v_x,l_r}) = \max^n_{y=x+1}[Dis(sh_x, sh_y) + f_{obj}(p^o_{v_y,l_r - l_{sh_y}}) + w \times l_{sh_x}] \times \tau(l_r, y)$, for $x < n$.

   Here $\tau(l_r, y) = 1$ if $l_r - l_{sh_y} \geq 0$, otherwise $\tau(l_r, y) = 0$.

With the above optimal-substructure we can calculate the object function value of the optimal path $f_{opt}(p^o_{v_0,L_{vs}})$ and all optimal sub-solutions. Then we can easily trace back and find the global optimal path as well as the skimming shots of the scene. In the case there are multiple global optimal pathes, the trace back algorithm will also find all of them. We concatenate the skimmings of each video scene then get the whole video skimming. Note that the algorithm might generate a video skimming that is a little shorter than the target length L_{vs}. This will not affect much about the content coverage of our video skim, in any case, we randomly select some video shots to fill that length.

The time complexity of this dynamic programming algorithm is $O(n^2 \times L_{vs})$, while the spatial complexity is $O(n \times L_{vs})$. For most video scenes, n and L_{vs} would not be very large and the algorithm completes quite quickly.

3. EXPERIMENTS AND DISCUSSIONS

We implemented the video summarization algorithms then applied them to some video clips. We employed a PC with 2.0G hz P4 CPU on the Win2000 OS for the experiments. The exponent control parameter k in the spatial-temporal dissimilarity function is set to 250, and the weight factor w in the objective function is set to 0.01. The threshold parameters t_1, t_2 are set to 3 seconds and 4 seconds, respectively. The selected video clips are from movies and sitcom videos described in Table 1. An example for a scene's key frames (shown as video shot groups) and the selected skimming video shots' key frames are shown in Fig. 4.

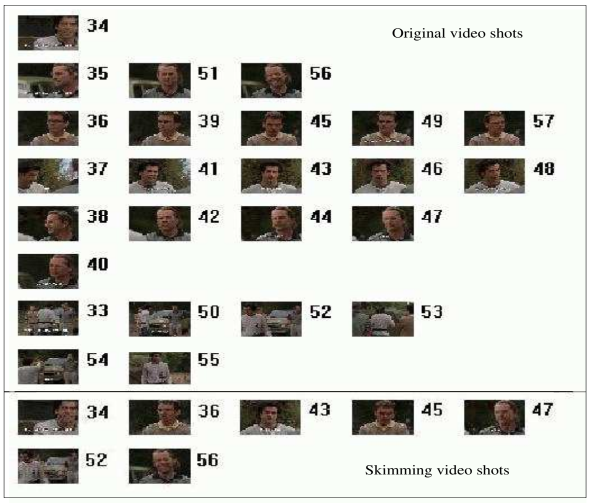

Fig. 4. Summarized scene key frames

In our experiments, ten people were invited as test users to watch the video skimming generated with two compress rates 0.15 and 0.30 then answer several questions about the video contents. Suppose there are N key event scenes in the video, we use the question "What?" to ask the test users to tell how many key events they can perceive by watching the video skimming. Then the score for the question "What?" is calculated as the average event number that the users can find divided by N. Similarly, question "Who?" deals with the key actors in the video. All scores are scaled to 10.

Table 1 shows the numerical results for the user test. From the table we conclude that the video skimmings' content coverage is still quit good at a skim rate of 0.15. Moreover, when the skim rate is 0.30, the skimming content coverage is even better.

Video Clip	Duration	Actors	Events	Skim Rate	Who?	What?
Movie1	1403 sec.	9	7	0.15	8.34	8.23
				0.30	9.42	9.56
Movie2	1230 sec.	7	8	0.15	8.22	8.44
				0.30	9.34	9.29
Movie3	477 sec.	6	4	0.15	7.78	8.05
				0.30	9.07	9.50
Sitcom1	1183 sec.	8	9	0.15	8.23	7.11
				0.30	8.89	8.43
Sitcom2	1202 sec.	7	10	0.15	8.43	6.84
				0.30	8.72	8.17

Table 1. User test results

4. CONCLUSION AND FUTURE WORK

Video summarization is an important technique for efficient video browsing and management. In this paper, we formulate the video skimming generation problem as a two-stage optimization problem. We obtain the video scene boundaries, determine each video scene's skim length, then we model each scene into a spatial-temporal relation graph, and employ dynamic programming to find each scene's optimal skimming. The whole video skimming is concatenated by each scene's skimming. We implemented the proposed algorithm and obtained encouraging experimental results.

In the future, we will further incorporate audio channel analysis to help our skimming generation. Moreover, intra-shot compression will be studied to shorten the video shots' length in order to further magnify the content coverage.

5. ACKNOWLEDGEMENT

The work described in this paper was fully supported by two grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK 4351/02E and Project No. CUHK4182/03E).

6. REFERENCES

[1] M. A. Smith and T. Kanade, "Video skimming and charaterization through the combination of image and language understanding techniques," in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 1997, pp. 775–781.

[2] Y. F. Ma, L. Lu, H. J. Zhang, and M. J. Li, "A user attention model for video summarization," in *Proceedings of ACM Multimedia*, 2002, pp. 533–542.

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

[4] S. Lu, I. King, and M. R. Lyu, "Video summarization using greedy method in a constraint satisfaction framework," in *Proceedings of 9th International Conference on Distributed Multimedia Systems*, 2003, pp. 456–461.

[5] C. W. Ngo, Y. F. Ma, and H. J. Zhang, "Automatic video summarization by graph modeling," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, 2003, pp. 104–109.

[6] Y. Rui, T.S. Huang, and S. Mehrotra, "Constructing table-of-content for videos," *ACM Multimedia Systems Journal, Special Issue Multimedia Systems on Video Libraries*, vol. 7, no. 5, pp. 359–368, Sept 1999.

[7] M. Yeung, B. L. Yeo, and B. Liu, "Extracting story units from long programs for video browsing and navigation," in *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems*, 1996, pp. 296–305.

[8] A. M. Ferman and A. M. Tekalp, "Efficient filtering and clustering methods for temporal video segmentation and visual summarization," *Journal of Visual Communication and Image Representation*, vol. 9, no. 4, pp. 336–51L, 1998.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C.Stein, "Introduction to algorithms," *The MIT Press*, 2001.