Semantic Video Summarization Using Mutual Reinforcement Principle and Shot Arrangement Patterns

Shi Lu, Michael R. Lyu and Irwin King
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong SAR
{slu, lyu, king}@cse.cuhk.edu.hk

Abstract

We propose a novel semantic video summarization framework, which generates video skimmings that guarantee both the balanced content coverage and the visual coherence. First, we collect video semantic information with a semi-automatic video annotation tool. Secondly, we analyze the video structure and determine each video scene's target skim length. Then, mutual reinforcement principle *is used to compute the relative importance value and cluster the video shots according to their semantic descriptions. Finally, we analyze the arrangement pattern of the video shots, and the key shot arrangement patterns are extracted to form the final video skimming, where the video shot importance value is used as guidance. Experiments are conducted to evaluate the effectiveness of our proposed approach.*

1. Introduction

Video is increasingly becoming the favorite medium for many communication entities for its extraordinary expressive power. With the ever increasing computing power and storage device capacity, the large scale digital video library system is growing rapidly. This massive growing video data thus gives rise to a challenge for efficient video browsing and management since it is time consuming to download and browse through the whole contents of the video. To solve this problem, video summarization, which engages in providing concise and informative video summaries to help people to browse and manage video files more efficiently, has received more and more attention in recent years. Basically there are two kinds of video summaries: *static video story board*, 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.

In recent years much work has been conducted on video summarization. For static summary generation, [1] tends to adapt to the dynamic video content. Later work presents video contents according to the detected video structure. In [2], the authors analyze the video structure after video segmentation, and then get a tree-structured Video-Table-Of-Contents(V-TOC). In [3], a scene transition graph is constructed as the video content presentation. A curve simplification approach is proposed in [4].

Compared with static video summary, dynamic video skimming is more attractive for it maintains the dynamic property of the video thus it makes more sense to the user. Much effort is also devoted to dynamic video skimming generation. In the VAbstract system [5], key movie segments are selected to form a movie trailer. The Informedia system [6] selects the video segments according to the occurrence of important keywords in the corresponding caption text. Later work employs perceptional important features to summarize video. In [7] the authors construct a user attention curve to simulate the user's attention toward different video contents. [8] proposes a utility function for each video shot, and video skimmings are generated by utility maximization. [9] assigns different weight scores on several important features of the video, and then selects the video skimming that maximizes the feature score summation. In [10], a graph optimization approach is proposed to guarantee the content coverage of the generated video skim.

Most of the traditional video skimming generation approaches are based on low-level video features, and they may not be able to guarantee that the generated video skim contains the semantically important contents. Thus, the video skim may not make sense to the users. To attack this problem, semantic information is needed to produce a meaningful video skimming. Unfortunately, although quite a lot attempts have been done to automatically annotate generic video and image contents [11, 12] as well as event detection in specific video categories like sports video [13], recognition of high-level semantic information like key ac-

tors, action taken is still beyond the capacity of present techniques. To collect reliable video semantic information we still need to manually annotate the video contents. Video summarization based on semantic annotation can be found in [14, 15].

In this paper, we propose a framework for dynamic video skimming generation that emphasizes both *balanced content coverage* and *visual coherence*. Figure 1 shows the overview workflow of our approach. We first segment the video into video shots, then we create a semantic content description for each shot with a semi-automatic annotation tool. To guarantee the balanced content coverage, by video structure analysis we determine the scene boundaries, and then we obtain the target skimming length for each scene. For each video shot, an importance value is calculated according to the *mutual reinforcement principle* [16], and the video shots are clustered according to their semantic content descriptions. Finally, we analyze the arrangement pattern of the video shots and the important shot strings are selected as the video skimming. In comparison with the traditional approaches, our approach has the following contributions: First, we employ the *mutual reinforcement principle* to calculate a global importance rank value for each shot, based on which we can ensure that the semantically important contents can be covered by the skimming; Secondly, we analyze the shot arrangement patterns, which is neglected by most existing approaches, and we utilize this information to make a tradeoff between content coverage and visual coherence.

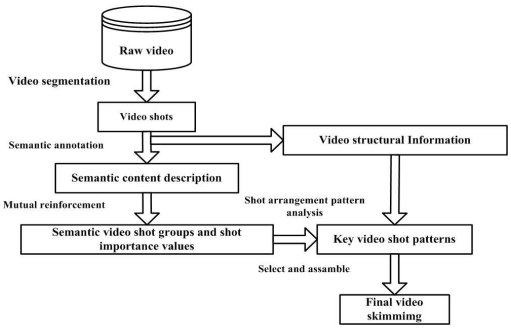

Figure 1. Overview of our framework

The paper is organized as follows. In Section 2 we describe the video annotation process. In Section 3 we analyze the structure of the video and describe how we calculate the semantic importance value for each video shot by mutual reinforcement principle. In Section 4 we present our video skim generation scheme. In Section 5 we show some experimental results. Finally, we make conclusion in Section 6.

2. Semantic video content annotation

2.1. Video shot segmentation

A video shot is an image sequence captured continuously by a single camera. It is the basic building block of edited videos like movies, broadcast news, TV shows, etc. By some video segmentation algorithms [17], we can efficiently detect the video shot boundaries. With the video shots detected, we can further make annotation for them, and explore the higher-level structure of the video.

For video shot sh_i, we use its first frame $kf_{i_{begin}}$ and the last frame $kf_{i_{end}}$ as the key frames that represent the visual content of the video shot.

2.2. Semi-automatic video shot annotation

Given the detected video shots, we define the semantic content description for a video shot, and we employ a semi-automatic process to annotate the video shots.

Normally when we see a video, the two questions we mostly want to ask is "Who?" (Who is the person this video is depicting?) and "What?" (What is the person doing?/What's happening?). Thus in this paper, for each video shot's content description, we currently use the following two *semantic-concept contexts* to describe the semantic concept of the video shots:

1. **Who**–This context describes the main person in this video shot.
2. **What**–This context describes the action taken by the actors, or the events happening.

Under each context there are several concept terms describing the contents. The video concept description, the corresponding contexts and the possible optional concept terms are organized in a tree-structured manner named *shot concept tree*, and the user can freely choose the right concept terms for the semantic context. Moreover, the user can easily extend, edit and reuse the shot concept tree. To accelerate the annotation process we employ a SVM-based relevance feedback image retrieval module proposed in [18]. When doing annotation, the annotator is provided with the video shot key frames for a preview. He can use the relevance feedback module to retrieve the visually similar video shots and copy the annotation to the similar shots thus the whole semantic annotation process can be highly accelerated. The similar video shots confirmed by the user are stored for further usage. The relevance-feedback-assisted video annotation interface is shown in Figure 2.

After the annotation, the video shot content description can be written in a two-unit tuple: $d = \{F_l, F_s\}$, which is the low level features and the high level semantic concept. Here $F_s = \{c_i\}$, where each context c_i contains sev-

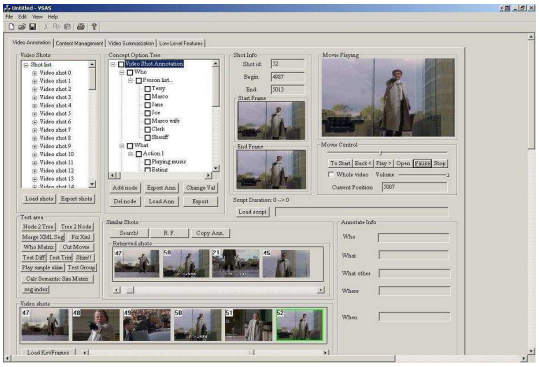

Figure 2. Video annotation interface

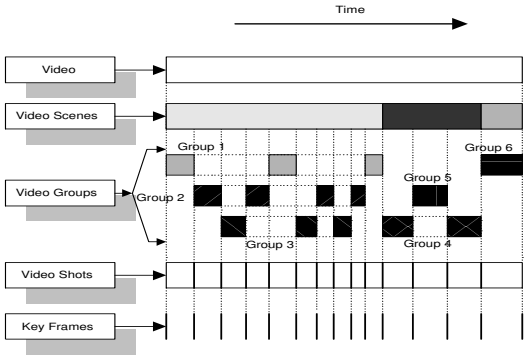

Figure 3. Hierarchical video structure

eral semantic concept terms $\{t_{ij}\}$. For a video shots sh_i, we can put the concept terms together into a keyword set $T_i = \{t_{ij}\}$.

3. Video structures and semantics

3.1. 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 paragraph does in the articles. A video scene is composed by a series of 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 [2]. Figure 3 shows the hierarchical structure of a video.

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.

Given the low-level features and the high-level semantic description for each video shot, we can define two similarity measures sim^l_{ij} and sim^s_{ij} as the similarity between two video shots based on low-level features and high-level semantic features.

The low-level similarity between two video shots is defined as the maximal H-S histogram correlation between their key frames, that is

$$VisualSim(sh_i, sh_j) = \max_{x,y} HistCorr(kf_{i_x}, kf_{j_y}),$$

where $x, y \in \{begin, end\}$.

Furthermore, we use the keyword similarity to measure their semantic similarity, defined as follows:

$$sim^s_{ij} = \frac{|T_i \bigcap T_j|}{|T_i \bigcup T_j|}.$$

and we can linearly combine the two similarity measure into a net similarity measure sim_{ij}:

$$sim_{ij} = k \times sim^l_{ij} + (1-k) \times sim^s_{ij}.$$

Based on measure sim_{ij} we can use the window sweeping algorithm in [2] to find the video shot group and video scene boundaries.

3.2. Video structure and video edit process

We have just determined the scene boundaries based on the visual and semantic similarity. Now we continue to explore the semantic structure of a video scene.

The video editing process, described in [19], is like the following: To describe an event the director will first shoot the environment from several different angles, then mix the video shots from various angles to assemble the final edited video. For example, to depict a conversation, there should be some overview shots showing all the people involved at the beginning and the end of the scene, and there may be several sets of video shots depicting each involved actor from different angles. The video shot sets are depicting the same content (the actor), but since they might be shot from different angles, they might not be grouped together by analyzing the low level features. In this case, the shot semantic description can help us to find such structure.

To better model the intention of the director, we propose a new concept called *semantic video shot group*. It is made of a set of video shots that depict the same semantic content. However, a semantic video shot group might not be composed by visually similar video shots. The semantic video shot group can be viewed as an intermediate entity

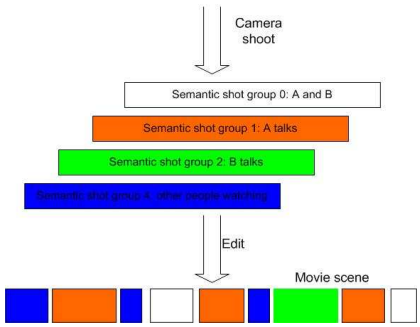

Figure 4. Movie edit process

between video shot group and video scene, and we can expect that video skimming generated upon this new structure can achieve better performance since it carries the semantic structure of the video. Another important sign of the director's intention is the way he or she arranges the semantic shot groups. The shot arrangement pattern analysis will be discussed in Section 5.

3.3. Mutual reinforcement and semantic video shot group detection

Given a video scene composed by a set of annotated video shots, a set of video annotation concept terms, and the corresponding contexts, we need to measure the relative importance of each video shot and each different concept term. We employ the following mutual reinforcement principle [16] to detect the semantic video shot groups and perform an importance evaluation for each detected video shot. Suppose that we have obtained a set of video shot descriptions $D = \{d_1...d_n\}$ based on a set of concept terms $T = \{t_1...t_m\}$ under the description context c, we hope to get a rank to measure the priority of the description items in the video shot description set. A weighted bipartite graph can be built from T to D in the following way: If description d_i contains term t_j, then we set up an edge between d_i and t_j, and we can compute a weight w_{ij} associated with the edge. w_{ij} can be any non-negative measure of the relationship between concept terms and descriptions. In this paper, we define the weight such that if description d_i contains concept term t_j then $w_{ij} = 1$, else $w_{ij} = 0$.

The idea of mutual reinforcement principle [16] is as follows: An important term should occur in many important descriptions and an important description should contain many important terms. The principle dictates that, the importance score of a concept term is determined by the importance scores of the descriptions it appears in; Moreover, the importance score of a semantic description is determined by the importance scores of the concept terms contained in it. Given the shot description set D, the term set T, and the weight matrix $W = [w_{ij}]$, we use the vector U and V to denote the importance scores for the term set T and the description set D. Mathematically, we have the following relationship:

$$U = \frac{1}{k_1} WV$$

and

$$V = \frac{1}{k_2} W^T U,$$

where k_1 and k_2 are some constants.

We can easily get

$$U = \frac{1}{k_1 k_2} WW^T U$$

and

$$V = \frac{1}{k_1 k_2} W^T WV.$$

Thus we can see that U and V should be the eigenvectors of the matrix WW^T and matrix $W^T W$, respectively. Since the elements in W are all non-negative, the largest eigenvalue of $W^T W$ and WW^T must be also non-negative. In that case, we may choose the eigenvectors corresponding to the largest eigenvalue of WW^T and $W^T W$ as the importance scores for the concept terms and the shot descriptions.

By this mutual reinforcement process we can find the importance score vector for all the video shot descriptions. We can see that video shots with similar content will have similar importance values. Thus by this importance value vector we can group the semantically similar video shots into semantic video shot groups. For a set of video shots, we compute importance value vectors based on context "who" and "what", thus resulting in two vectors V_{who} and V_{what}. Consequently, the final importance vector is obtained by

$$V = V_{what} + V_{who}.$$

We combine these two value vectors then use them to classify video shots. Since the importance value is a measure for a shot's relative semantic rank, according to the importance value vectors, we can select several semantic video shot groups with high importance values as *key video shot groups*, and the rest content is treated as a *background shot group*. Figure 5 shows the importance vector we obtained from a video scene. Figure 6 shows some classified shots based on the importance vector. On the top row the shots contain two key actors thus they have the highest importance value. The second and third row are those video shots depicting one main actors. The above three video shot groups form the key video shot groups. Other video shots which do not contain key actors form the background video shots group, as shown in the bottom row.

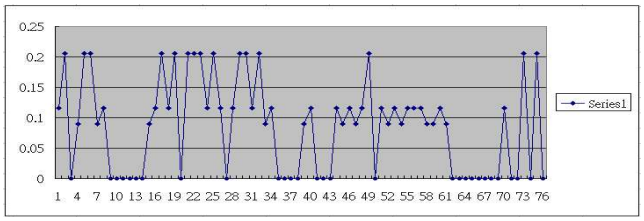

Figure 5. Importance vector of video shots

Figure 6. Some classified video shots

After we have found the semantic video shot groups, our video summarization process will be just like an inversion of the video edit process followed by another edit process. We first group the video shots depicting the same content, which is just the inversion of movie editing, and we select some shots from the same group according to some rules then reassemble them into the final video skimming.

4. Semantic video summarization

4.1. Summarization requests and goals

Basically there are two kinds of video skimmings: *overview* and *highlight*. For a specific domain like sports video and news, the user already knows some domain-specific knowledge and he may just request those video shots that he is interested in like "give me 3 minutes of video about goals and corner kicks". This kind of video skimming is called "highlight". But for movies, mostly the user is totally unaware of the content thus can only specify a target length and hope to see enough details about the video. The request may be like "give me 3 minutes of preview showing what this video is about". We call this kind of video skimming an "overview". In this paper we concentrate on the video overview generation.

To obtain meaningful video skimming, we specify several goals that we would achieve as follows:

1. **Conciseness**–As required, the length of the generated video skimming should be within the user specified length L_{vs}.
2. **Balanced content coverage**–As the video is a structured document, the video skimming should be able to represent the original content with balance. At the same time, the visual and semantic diversity of the original content should be reflected by the video skimming.
3. **Visual coherence**–One problem for traditional video skimming generation is that the user often feel that the video skimming is quite choppy. Thus we hope to increase the coherence of the video skimming while preserving the content coverage.

Now that we have established the semantic video shot groups and video scene boundaries, the final video skimming can then be made by first making sub-skimmings for each scene and then concatenating them. Thus our video skimming generation scheme includes three steps: First, determine the target length for each scene; second, extract the sub-skimming according to each length; and finally, assemble the sub-skims to form the final skimming.

4.2. Determine the sub-skimming length for each scene

Suppose that the video is composed by a set of detected scenes $\{Sc_i\}$, given total video skimming length L_{vs}, we need to distribute the skimming length to each scene. It's natural that longer and more complex video scenes should share a longer part in the final video skimming. To describe the complexity of a video scene, we define the content entropy for a video scene Sc_i as follows:

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

Here l_{Sg_j} and l_{Sc_i} are the length of the video shot group and video scene respectively, in terms of the image frame number.

After we have calculated the content entropy for each video scene Sc_i, given the total video skimming length L_{vs}, we determine the target skimming length Sl_i for each video scene in the following way:

1. For the video scenes $\{Sc_1...Sc_n\}$, we first calculate $Sl_i = L_{vs} \times \frac{Entropy(Sc_i) \times l_{Sc_i}}{\sum_{j=1}^{n} Entropy(Sc_j) \times l_{Sc_j}}$. If Sl_i is less than the preset threshold t_1, then the corresponding scene is considered as non-important thus will be discarded.
2. For the remaining scenes $\{Sc'_1...Sc'_m\}$, we set $Sl_i = L_{vs} \times \frac{Entropy(Sc'_i) \times l_{Sc'_i}}{\sum_{j=1}^{m} Entropy(Sc'_j) \times l_{Sc'_j}}$.

4.3. Extracting video shots by string analysis

Now that we have determined all the semantic video scenes' target lengths, we can continue to extract some video shots from each video scene to form the sub-video skimming. In [10] we proposed a graph optimization algorithm to select video skimming shots. Each detected video scene is modeled into a graph, and the video skimming is generated by searching a constrained longest path in that graph, such that a balanced content coverage can be achieved. However, this method select separate video shots thus the video skimming seems choppy. In this paper, we use a new method based on string sequence analysis to select the shots, which is able to generate a more coherent video skimming while still guarantee the content coverage. We will compare the performance of the two methods in the experiment.

After the mutual reinforcement process, we have the importance vector V for the video shots, where each component v_i is the importance value of video shot sh_i. Based on the importance value we can classify the video shots into a set of semantic shot groups $G = \{g_k\}$, including several key video shot groups and one background video shot group. Each semantic group g_k has a group label lb_k, shared by the video shots contained in it. Let the set of group labels be LB. Given a video scene $Sc_x = \{sh_1....sh_n\}$, we can have a group label string $lb_1....lb_n$, where $lb_i \in LB$.

Here we provide some definitions for video shot string analysis.

1. A *video shot string* str is defined as a series of consecutive video shots $\{sh_1....sh_x\}$, with the group label string $\{lb_1...lb_x\}$. The importance value of a video shot string I_{str} is defined as $I_{str} = \sum_{j=1}^{x} v_j$, where v_j is the importance value of videos shot sh_j.
2. A *non repetitive shot string* (nrs string) is defined as a video shot string $\{sh_1....sh_x\}$, $\forall i, j \in \{1...x\}$, $lb_i \neq lb_j$.
3. A *k-non repetitive shot string* (k-nrs string) is defined as a non repetitive shot string with length k. We use $\{k\text{-}nrs_j\}$ to denote a set of nrs strings with length k.
4. If str_i is the sub-string of str_j, we say that str_j *covers* str_i. For example, the 4-nrs string 3124 covers two 2-nrs strings $\{312, 124\}$, three 2-nrs strings $\{31, 12, 24\}$ and four 1-nrs strings $\{3, 1, 2, 4\}$.

nrs strings carries important information about how the video editor arrange the video shots. We can easily find all $k-nrs$ strings by scanning the video label string. Then we use them as skimming candidates. Some sample nrs strings are shown in Figure 7.

To ensure the balanced content coverage, we hope that the skimming shots should cover as many semantically important shots as possible. On the other hand, to guarantee the coherence of the video skimming, we should pick more longer substrings from the video shot list in the skimming. Thus the k-nrs strings become good candidates for video skimming since they are composed by video shots depicting non repetitive contents, and they are a coherent part of the original video. By scanning the video shot string we can easily get all k-nrs strings for all k.

Figure 7. Several detected nrs string in a movie scene

We then formulate the video skimming generation problem as follows:

Problem 4.1 *For a video scene, given the target skimming length L_{vs}, a set of video shots $\{sh_1...sh_n\}$ contained in the scene, the corresponding video shot length set $\{l_i\}$, and the corresponding video shot group label set $\{lb_1...lb_n\}$, find a continuous nrs string set $SKIM = \{nrs_j\}$, such that:*

1. *$\sum_j I_{nrs_j}$ is maximized (semantic importance summation is maximized);*
2. *$|SKIM|$ is minimized;*
3. *Minimize the duplicated items in $SKIM$;*
4. *$\sum_j (l_{nrs_j}) = L_{vs}$.*

To solve the above problem, we propose a greedy method algorithm, as described in Algorithm 1.

Algorithm 1 continues selecting the most important uncovered nrs string into the video skimming, and discard the already-covered short nrs strings so that the semantic important contents are selected while the redundancy of the video skimming is minimized. By this algorithm we obtain a set of coherent video segments as the video skimming, such that the content coverage and coherence can be simultaneously achieved.

5. Experiments

To test the performance of our proposed approach, we have implemented the proposed video annotation and summarization framework and applied it to some movie clips. We employed a PC with 2.0G hz P4 CPU and 512Mb RAM

Algorithm 1 Video skimming selecting algorithm

Input: The set of all nrs strings NRS; The target skimming length L_{vs};
Output: The selected nrs set $SKIM$ that form the video skimming
BEGIN $SKIM = \emptyset$
STEP 1: Sort the nrs strings in NRS according to their importance value;
while $L_{vs} > 0$ **do**
Select the best nrs string nrs_{opt}, such that:

1. $L_{nrs_{opt}} < L_{vs}$
2. $\forall nrs_i \in N$ and $L_{nrs_i} < L_{vs}$, $I_{nrs_{opt}} \geq I_{nrs_i}$

if Found **then**

1. $SKIM = S \cup \{nrs_{opt}\}$
2. $L_{vs} = L_{vs} - L_{nrs_{opt}}$
3. $NRS = NRS - \{nrs_t | nrs_{opt}$ covers $nrs_t\}$

else if Not found **then**
GOTO END
end if
end while
END

on the Win2000 OS as the test bed. The weight parameter k is set to 0.6, and the time threshold t_1 is set to 4 seconds. Three movie clips and one sitcom clip are processed, and two video skimmings at skim rate 0.15 and 0.30 are extracted for each test video clip. Details about the video clips are shown in Table 1.

To evaluate the quality of the generated video skimming, we employ two criterion: *meaningfulness* and *favorite*. Since it is hard to objectively evaluate a video skimming, we use the following subjective test to compare the performance of our proposed scheme and the method we proposed in [10]. We have invited 10 test users to watch the video skimming generated from the video by the two methods at skim rate 0.15 and 0.30. To evaluate meaningfulness, the test users are asked to answer several questions about the key events that the video depicts (Who has done what?). The scores are scaled to $[0, 100]$. To compare the favorite, we ask the user to select a "better" video skimming between the video skims generated by the two approaches, and the number of users who choose the skim as "better" is recorded as the favorite score. Figure 8 and Figure 9 show the average meaningfulness and favorite scores respectively for the video skims generated by our proposed method and the method in [10]. The number of continuous video segments according to the original video that the video skims contain is employed as a measure of coherence. The experimental results are summarized in Table 1 (Where Mfn. means Meaningfulness, Fav. means favorite, N.S. means Number of Segments, SEM means the new semantic approach, and GRA means our old graph based approach).

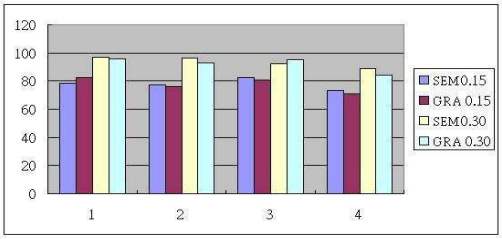

Figure 8. Meaningfulness Scores

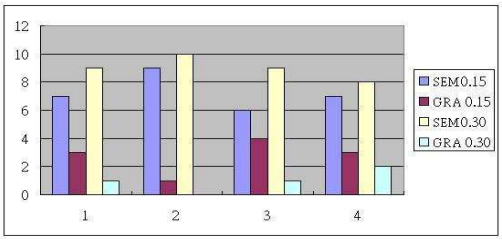

Figure 9. Favorite Scores

Our experimental results are quite encouraging. In terms of meaningfulness, at skim rate 0.15, the proposed method obtain a quite high mean score 77.95. At skim rate 0.30, the score is even higher. Moreover, in most cases, our new approach has gained a higher score than our previous approach. In terms of favorite, we can see that although both video skimmings are meaningful, most users would prefer the video skimming generated by the new method. We also find that our new approach generates much less video segments than the previous approach, which greatly increases the coherence. From the experimental results we can make the conclusion that our proposed method is able to generate a better video skimming in comparison with our previous work.

6. Conclusion

In this paper, we illustrate a novel framework for semantic video summarization. We obtain the semantic information and structure information of the video, compute the semantic importance of each video shot, analyze the shot arrangement patterns, and finally, we obtain a dynamic video skimming by selecting the key video shot strings. The experimental results show that our approach ensures both balanced content coverage and visual coherence.

Video Clip	Duration	Actors	Events	Skim Rate	Mfn.	Fav.	N.S.
Movie1	1403 sec.	9	7	0.15	**78.5**/82.3	**7**/3	**15**/59
				0.30	**97.1**/95.6	**9**/1	**22**/89
Movie2	1230 sec.	7	8	0.15	**77.5**/ 76.4	**9**/1	**16**/44
				0.30	**96.2**/ 92.9	**10**/0	**22**/65
Movie3	477 sec.	6	4	0.15	**82.5**/ 80.5	**6**/4	**12**/30
				0.30	**92.5**/95.0	**9**/1	**19**/46
Sitcom1	1183 sec.	8	9	0.15	**73.3**/71.1	**7**/3	**24**/54
				0.30	**88.8**/84.3	**8**/2	**46**/87
Average	—	—	—	0.15	**77.95**/77.57	**7.25**/2.75	—
				0.30	**93.65**/91.65	**9**/1	

Table 1: User test results. Scores for the new approach are in **bold**

In the future, we will investigate interactive personalized video summary upon our framework, and develop more efficient automatic or semi-automatic video content annotation techniques.

7. Acknowledgements

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).

References

[1] H. J. Zhang, D. Zhong, and S. W. Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, 30(4):643–658, 1997.

[2] 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*, 7(5):359–368, Sept 1999.

[3] 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*, pages 296–305, 1996.

[4] D. DeMenthon, V. Kobla, and D. Doermann. Video summarization by curve simplification. In *Proceedings of ACM Multimedia 98*, pages 13–16, 1998.

[5] R. Leinhart, S. Pfeiffer, and W. Effelsberg. Video abstracting. *Communication of the ACM*, pages 55–62, December 1997.

[6] 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*, pages 775–781, 1997.

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

[8] 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*, pages 189–198, 2002.

[9] 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*, pages 456–461, 2003.

[10] S. Lu, I. King, and M. R. Lyu. Video summarization by video structure analysis and graph optimization. In *Proceedings of The 2004 IEEE International conference on multimedia and expo*, 2004.

[11] M. R. Naphade, I. V. Kozintsev, and T. S. Huang. A factor graph framework for semantic video indexing. *IEEE Transaction on Circuits and Systems for Video Technology*, 12(1):40–52, January 2002.

[12] R. Lienhart and A. Hartmann. Classifying images on the web automatically. *Journal of Electronic Imaging*, 11(4):40–52, October 2002.

[13] N. Bagaguchi. Generation of personalized abstract of sports video. In *Proceeding of IEEE ICME*, pages 800–803, 2001.

[14] X. Q. Zhu, J. P. Fan, A. K. Elmagarmid, and X. D. Wu. Hierarchical video content description and summarization using unified semantic and visual similarity. *ACM/Springer Multimedia Systems Journal*, 9(1):31–53, 2003.

[15] B. L. Tseng, C. Y. Lin, and J. R. Smith. Video summarization and personalization for pervasive mobile devices. In *SPIE Electronic Imaging 2002 - Storage and Retrieval for Media Databases*, pages 383–392, 2002.

[16] H. Y. Zha. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of SIGIR 2002*, January 2002.

[17] A. M. Ferman and A. M. Tekalp. Efficient filtering and clustering methods for temporal video segmentation and visual summarization. *Jounal of Visual Communication and Image Representation*, 9(4):336–51L, 1998.

[18] C. H. Hoi, C. H. Chan, K. Z. Huang, M. R. Lyu, and I. King. Biased support vector machine for relevance feedback in image retrieval. In *Proceedings of International Joint Conference on Neural Networks 2004*, July 2004.

[19] Merritt Greg. Film production : the complete uncensored guide to independent filmmaking. *Los Angeles, CA : Lone Eagle Pub*, 1998.