Content Analysis and Summarization for Video Documents

Lu, Shi

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Department of Computer Science & Engineering

December, 2004

Content Analysis and Summarization for Video Documents

submitted by

Lu, Shi

for the degree of Master of Philosophy

at the Chinese University of Hong Kong

Abstract

Video is increasingly becoming the favorite medium for many communication entities for its extraordinary expressive power. With the fast advancement of network bandwidth and high-capacity storage devices, large-scale digital video library systems are growing rapidly. However, such an expanding video repository gives rise to new challenges to both the end users and the content providers. Since it is time consuming to download and browse through the whole contents of the video, browsing and managing the video database can be quite tedious. To solve this problem, video summarization, which engages in providing concise and informative video summaries to help people to browse and manage video files efficiently, has received more and more attention in these 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.

This thesis presents our work on automatic video summarization. First, we present our early work on a greedy method-based video skim generation approach. Several important features are extracted from the video. Given

the distribution of the important features, a greedy selection algorithm is used to select video segments to form a video skimming. A refinement process is employed to increase the coherency of the video skimming.

Then, we propose a graph optimization-based video summarization framework. The workflow of this framework consists four major steps: (1) video structure analysis, (2) video scene analysis, (3) graph modeling and optimization. Both a dynamic video skimming and a static video summary are generated as the content preview for a video document. By doing optimization on the graph, can ensure both the temporal coverage and the visual content diversity of the video summary.

In order to adapt the high-level semantic information of the video, we have designed a semi-automatic video annotation system to help the user to efficiently create content descriptions for each detected video shot. Given the semantic content descriptions of the video shots, we employ the mutual reinforcement principle to calculate an importance measure for each video shot, then create video summaries based on the importance measure and shots arrangement patterns. Experiments show that the new approach achieves better performance than our previous approaches.

We have implemented the three proposed video summarization frameworks, provided the functionality to the users and conducted some subjective experiments. The user tests show that the video summaries generated by our framework are able to guarantee both the balanced content coverage and the visual coherence.

Content Analysis and Summarization for Video Documents

submitted by

Lu, Shi

for the degree of Master of Philosophy
at the Chinese University of Hong Kong

— ȣ ḟ
↱ ꟼ ɉ ȣ 4 ḫ ῼ ḟ v
ΰ Ṯ >
ḃ ῼ

>
>

>
>

>
>

>

Acknowledgment

I would like to take this opportunity to express my gratitude to my supervisors, Prof. Irwin King and Prof. Michael R. Lyu, for their patient guidance, ceaseless support, and encouragement during my M. Phil study. The inspiring advice and insightful criticism from Prof. King and Prof. Lyu are extremely essential and valuable to my research papers and my thesis. This dissertation could not be completed without their effort.

I am also very grateful for the valuable suggestions and comments that Prof. T. T. Wong and Prof. M. C. Lee have given to me in marking my term papers, term presentations and my thesis.

I would also like to show my gratitude to the Department of Computer Science and Engineering, The Chinese University of Hong Kong, for the provision of the best equipment and pleasant office environment for high quality research.

I want to give my thanks to my fellow colleagues in the department of computer science. They helped me in solving technical problems, enlightened me with new research ideas, helped me to finish the user experiments and gave me encouragement. It is they that have made my two years' research time joyful and wonderful.

My special thanks must go to my family who has given me the greatest support and encouragement, so that I can concentrate on my postgraduate study.

Last but not least, I would like to thank all the scientists and engineers for their endeavor and contributions upon which I can ground my research work. All my work done is based on the fact that I can stand on their shoulders.

Contents

Table 2.1: Comparison of major static video summary generation methods

Methods	Advantages	Disadvantages
Shot-based methods [15, 80, 53, 13, 29, 6]	Comprehensive content coverage	Often generate too many un-organized video key frames
Motion-based methods [67, 65, 66, 26]	Can adapt to the motion characteristics of the video	Often generate too many un-organized video key frames
Mosaic-based methods [28, 72, 38, 2]	Can well reflect the camera motion	Not robust under intense motion and complex background
V-ToC and STG [50, 74, 21]	Reveal the high-level structure of the video	The video shot grouping algorithm is not very robust

2.2 Dynamic Video Skimming

Compared with static video summary, dynamic video skimming preserves the dynamic properties of the original video, thus it is more attractive and helpful to the user. Moreover, the audio information is also preserved thus the video skimming is able to make more senses.

The moving video skimming can be classified into two types: Overview and Highlight. Facing a new movie, mostly the user is totally unaware about the content. They can only specify a target length and hope to see enough detail about the movie. The request may be like \Give me three minutes of preview showing what this movie is about", and we call this kind of video skimming \overview". But for 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. In this situation, the request may be like \Give me three minutes of video about goals and corner kicks". This kind of video skimming is called \highlight".

For overview generation, early attempts include the VAbstract [47] and MOCA project [47, 29]. In the VAbstract system, the most characteristic video segments are extracted to form a movie trailer. The frames with high-contrast are detected as the parts containing important contents, while the frames having largest frame differences are extracted as the high-action parts. In order to preserve the basic mood of the original movie, the scenes that have color composition similar to the average color composition of the entire movie are included in the skimming. Moreover, the recognition of dialog scenes is performed by detecting the spectrum of a spoken "a" since "a" occurs frequently in most languages. Finally, all selected scenes are organized in their original temporal order to form the movie trailer. There are some interesting ideas in this system, but some parts of the algorithm are too simple to be effective and will need a lot of improvement. The researchers also lack a thorough user study to support their conclusions. In the MoCA project, which is an improved version of VAbstract, special events such as close-up shots of leading actors, explosions, and gunfire, are detected to help determining the important scenes.

Another straightforward approach to save the video viewing time is proposed by speeding up the playback speed of the original video. As studied by Omoigui, et al. [46] at Microsoft Research, the entire video could be watched in a shorter amount of time by fast playback with almost no pitch distortion using the time compression technology. CueVideo system provides faster playback speed when playing long, static video scenes and slower speeds for short, dynamic video scenes [48]. Although the playback time has been reduced but the temporal property of the original video is distorted, which may mislead the users about the nature of the original video. Similar work

4.2 Video Shot Group Construction

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 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 four-level hierarchical structure: video, video scenes, video shot groups, and video shots [50]. Figure 4.5 shows the hierarchical structure of a video.

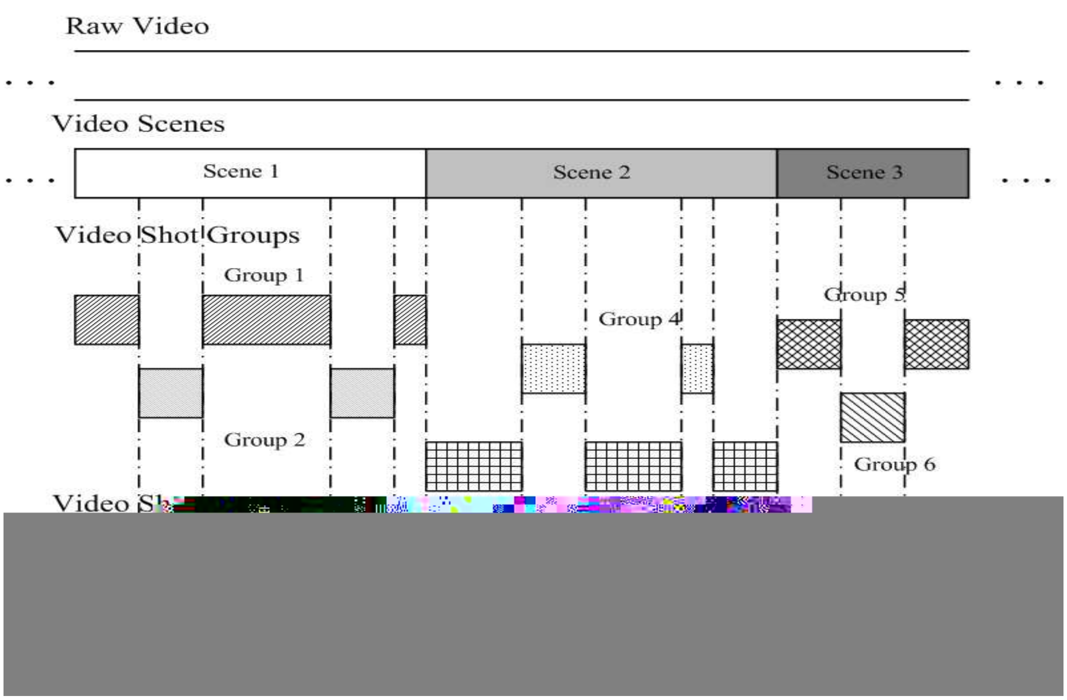

Figure 4.5: Hierarchical video structure

In the remaining part of this chapter, 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, representing the total number of images containing in them respectively.

The structure of a video is built in a bottom-up manner. After we have determined the video shot boundaries, we continue to build up the hierarchical

4.2.2 Video Shot Grouping by VToC

In [50], the authors proposed the VToC (video table of contents) algorithm to group the visually similar video shots. The VToC algorithm set up the video shot-group-scene structure with the following time-constrained grouping method. First, two measures are calculated for each video shot:

1. Color histogram of the entire shot key frame image.
2. Shot activity, as the average color difference of the frames contained in the shot.

In this chapter, we use the video shot similarity measure defined in Equation 4.13 in the VToC construction procedure.

To determine if a video shot sh_i should be put in group $g_j = \{sh_{j_1} \ldots sh_{j_m}\}$, a shot-to-group similarity should be calculated. The shot-to-group similarity is calculated as:

$$sim_{sh_i, g_j} = \max_t sim_{i, j_t}, \tag{4.14}$$

which is the maximum similarity or

$$sim_{sh_i, g_j} = \frac{\sum_t sim_{i, j_t}}{m}, \tag{4.15}$$

which is the average similarity.

The temporal distance $d_T(sh_i, sh_j)$ between video shots sh_i and sh_j is defined as the temporal distance between their center image frames. Consequently, the temporal distance between video shot sh_i and video shots group g_k is defined as:

$$d_T(sh_i, g_j) = \min_t d_T(sh_i, sh_{k_t}), \tag{4.16}$$

where $sh_{k_t} \in g_k$.

When the shot pairwise similarity is obtained, the VToC algorithm makes the video shot groups in the following way, as shown in Algorithm 4.1:

Algorithm 4.1 VToC Video Shot Grouping Algorithm

Input: The set of video shots $SH = \{sh_1, \ldots, sh_n\}$, the shot pair similarity $\{sim_{ij}\}$ for all $i, j \in \{1 \ldots n\}$;
Output: The set of video shot groups $G = \{g_1, \ldots, g_m\}$
1. Add the first group g_1 to G, assign sh_1 to g_1;
for $sh_i \in SH$ do
  for $g_j \in G$ do
    calculate the shot-to-group similarity $sim_{sh_i; g_j}$;
    if $sim_{sh_i; g_j} \geq s_{th}$ and $d_T(sh_i; g_j) \leq T_{th}$ then
      assign sh_i to g_j
    else
      add new shot group g_k to G, assign sh_i to g_k
    end if
  end for
end for

The performance of VToC algorithm is significantly influenced by the similarity threshold s_{th}. However, since different video shot groups may have different proper similarity thresholds, so that the time-constrained grouping with a hard threshold may not be able to correctly detect all video shot groups and is likely to generate a lot of over-groupings and miss-groups.

4.2.3 Spectral Graph Partitioning

Spectral graph clustering [54] is a recursive clustering method proposed to segment images into regions based on the color similarity and spatial distance between image pixels. In [54], a complete graph is constructed with the pixels in the image as vertices and on each edge there is a weight describing the color similarity and spatial distance between image pixels. The graph is then recursively partitioned into smaller clusters, on each partitioning, a global optimal normalized cut is calculated to ensure the performance. Finally the original image is segmented into coherent regions.

The graph partitioning can be migrated from image segmentation to other clustering scenarios, in which if the graph can be built. Our shot clustering problem is one of such cases that can utilize the spectral graph clustering. Given a series of video shots, we can also construct a graph $G(V;E)$, where V is the vertex set, in which each element corresponds to a video shot. E is the edge set, in which the edges connects each shot pair in V. On each edge e_{ij} there is an edge weight w_{ij}, which is a measure of the visual similarity between the two video shots.

Given the graph $G(V;E)$, we may cut the vertex set V into disjointed sets A and B, and compute the Normalized Cut Value to evaluate a cut:

$$NCut(A;B) = \frac{cut(A;B)}{assoc(A;V)} + \frac{cut(A;B)}{assoc(B;V)}; \tag{4.17}$$

where

$$cut(A;B) = \sum_{i \in A; j \in B} w_{ij} \tag{4.18}$$

is the cut value and

$$assoc(A;V) = \sum_{i \in A; j \in V} w_{ij} \tag{4.19}$$

is the association of A with the vertex set V, an example figure is shown in Figure 4.6. In the figure, the vertex set V is divided into vertex set A and B. The edges contribute to $cut(A;B)$ is red, the edges contribute to $assoc(A;V)$ is green, while the edges contribute to $assoc(B;V)$ is blue.

The meaning of the cut value is the total similarity summation between partition A and B. The meaning of the association is in fact the total similarity summation within the vertices in partition A and B. A good partition would create two self-similar and mutual-dissimilar partitions.

We hope that by cutting the graph into A and B, we can separate the dissimilar vertices into different parts. First, $Cut(A;B)$ should be small for we want the dissimilar vertices be separated. Second, the values of $assoc(A;V)$ and $assoc(B;V)$ should be large for cluster A and B should contain similar video shots. Thus the optimal graph cut problem can be formulated in Problem 4.1 as follows:

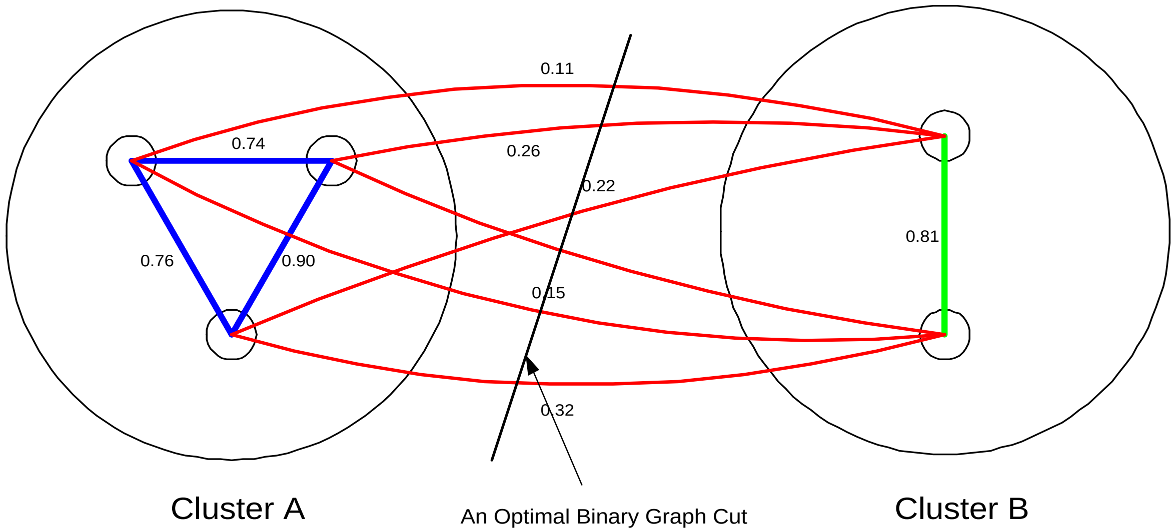

Figure 4.6: Spectral graph cut sample

Problem 4.1 Given the complete graph G, find the optimal partition $Cut(A;B)$ for G such that the Normalized Cut Value

$$NCut(A;B) = \frac{cut(A;B)}{assoc(A;V)} + \frac{cut(A;B)}{assoc(B;V)}$$

is minimized.

According to [54], the $NCut$ minimization problem can be transformed into solving a standard eigen system:

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}x = \lambda x. \tag{4.20}$$

Here D is a diagonal matrix, $d_{ii} = \sum_j w_{ij}$. And W is the shot similarity matrix. The eigenvector corresponding to the second smallest eigenvalue can be used to partition V into A and B. A detailed mathematical demonstration can be found in [54].

We can recursively construct sub-graphs A and B and solve the eigen system to get finer clusters until some ending conditions are met. In this way we can partition the vertex set into smaller sets. When the elements in a vertex set are "similar" enough we cease the partitioning process. Then we get several video shot groups and a series of "single" video shot groups, in which only one video shot is in the group. We put all the single un-grouped shots together to form a background video shot group. With this grouping information we can easily build up the video scene structure. The merit of this recursive algorithm is that every time it splits the vertex set, a global optimal cut is found thus the clusters can better be adjusted to the nature of the original data. The recursive algorithm is described in Algorithm 4.2.

Algorithm 4.2 Recursive Clustering Algorithm based on Spectral Graph Partitioning

---

Input: The set of video shots $SH = \{sh_1, \dots, sh_n\}$, the shot pair similarity $\{sim_{ij}\}$ for all i,j $\in \{1 \dots n\}$;
Output: The set of video shot groups $SG = \{sg_1, \dots, sg_m\}$
1. Add the first group sg_1 to SG, assign sh_1 to sg_1;
for $sh_i \in SH$ do
  for $sg_j \in SG$ do
    calculate the shot-to-group similarity sim_{sh_i, sg_j};
    if $sim_{sh_i, sg_j} \geq s_{th}$ and $d_T(sh_i, sg_j) \leq T_{th}$ then
      assign sh_i to sg_j
    else
      add new shot group sg_k to SG, assign sh_i to sg_k
    end if
  end for
end for

The graph partitioning algorithm outperforms the highly threshold-dependent window sweeping algorithm, for every time the algorithm generates a new cluster, it considers the global similarity information and the splitted clusters are global optimal [54]. It is also easy to create self-adaptive thresholds for clusters with different sizes, while the hard-threshold based grouping algorithm tend to over-divide groups or fail to classify different groups, according to the fixed threshold. Some shot groups generated by each method are shown in Figure 4.9 and Figure 4.10.

According to the definition of the video shot groups, we can pose an inte threshold, T_{th}, to the intervals between consecutive video shots in the detected video shot clusters, then we split the cluster into video shot groups if the interval is greater than T_{th}.

4.3 Video Scene Detection

After we have detected the video shot groups, we can continue to find the video scenes. According to our definition there are two kinds of video scenes, loop scenes, which is composed by more than one video shot groups, and progressive scenes, which is composed by a series of visually dissimilar video shots.

According to the definition, the contents of interlaced video shot groups should be relevant thus should be put in the same scene, which is a story unit in the video like a paragraph in an article. Given the detecte

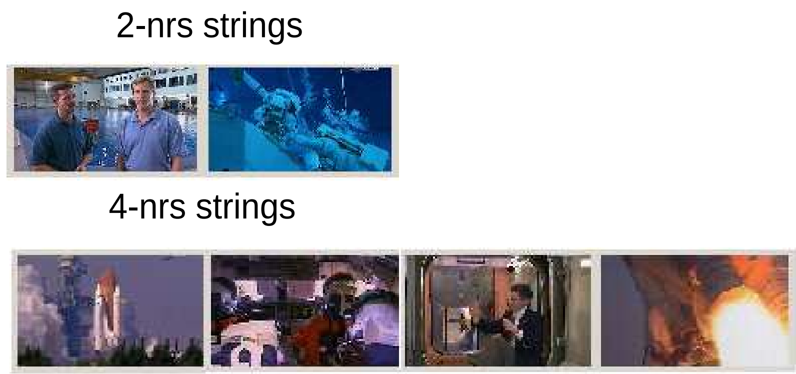

Figure 4.8: Several detected nrs strings

Given a video shot sh_i and a k-nrs string $str_j = \{sh_{j_1}...sh_{j_k}\}$, we can define the visual similarity between them as:

$$sim(sh_i, str_j) = \sum_{x=1}^{k} sim(sh_i, sh_{j_x}) \times \frac{Length(sh_{j_x})}{\sum_{y=1}^{k} Length(sh_{j_y})}. \tag{4.21}$$

After that, we can define the visual similarity function between two k-nrs strings str_i and str_j:

$$sim(str_i, str_j) = \sum_{x} sim(sh_{i_x}, str_j) \times \frac{Length(sh_{i_x})}{\sum_{y} Length(sh_{i_y})}, \tag{4.22}$$

where $sh_{i_x} \in str_i$. In the next chapter we will discuss how we model the video shot strings in a video scene with a graph based on their similarities and temporal orders.

4.5 Experiments

In this section we evaluate the algorithms proposed for video structure analysis. We carry out the experiments on a PC platform with win2000 OS, P4 2.0GHz CPU and 512MB RAM. The video processing module is developed with Visual

C++, based on Microsoft DirectsShow SDK. We employ several documentary videos and several movie clips as the test data set.

For video shot cut detection, we have implemented our proposed approach and appled it to several test videos from movie and documentary video. The experimental results are shown in Table 4.1. We compare the result with the approach in [42] and found that our method indeed outperforms the previous method in [42], in which the shot cut detection accuracy reported varies from 63.4% to 85.9%. The difference should lie in our non-linear neighborhood filtering. From the resulting data we can see that our method is quite robust and accurate, in most cases, our method achieves a correct detection rate around 95%.

Table 4.1: Video cut detection results. F.D. means False Detection and M.D. means Missed Detection

Video type	Length	Ground truth	Detected	F. D.	M. D.	Accuracy
Movie	477 sec.	166	157	0	9	94.6%
Movie	1440 sec.	237	235	6	8	96.6%
Document	1380 sec.	147	141	3	9	94.3%
News		40	39	1	1	95.0%

For fade detection, our result is shown in Table 4.2. The fade detection process is more sensitive to noise, thus its accuracy is not as good as the video shot cut detection. In the experiments the accuracy ranges from 75% to 85.7%.

Table 4.2: Video fade detection results

Video type	Length	Ground truth	Detected	F. D.	M. D.	Accuracy
Document	620	12	10	1	3	75.0%
Document	1380	21	23	5	3	85.7%
Movie	477	4	3	0	0	75.0%

For videos shot group formation, we have implemented the graph cut algorithm and the window-sweeping algorithm in [50]. Some sample video shot groups generated by the two algorithms are shown in Figure 4.9 and Figure 4.10.

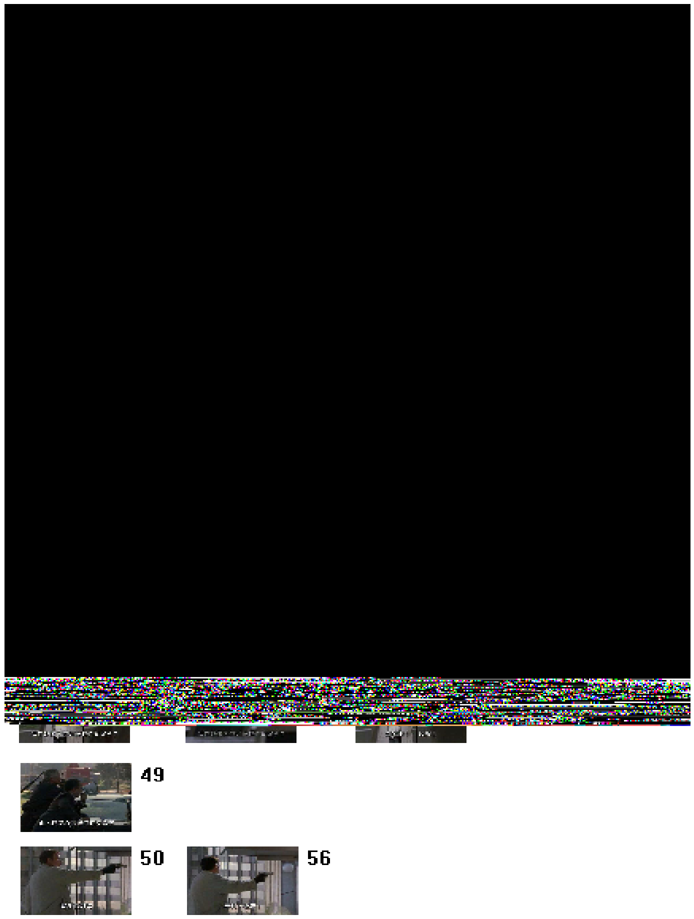

Figure 4.9: Several detected video shot groups by the ToC method

From the detected video shot groups we can construct the video scenes. However, as video scene is a high level concept, the ground truth of "video scene boundary" is difficult to be defined objectively. So in the following experiment we employ the "Human opinion" as the ground truth. Table 4.3 and Table 4.4 summarize some results for scene detection generated by spectral graph clustering method and the original ToC method (F. D. for False Detection and M. D. for Missed Detection).

From the experimental result data in Table 4.3 and Table 4.4, we can see that the spectral clustering method outperforms the ToC method in correctly

Figure 4.10: Several detected video shot groups by spectral graph partition

detecting video scenes. Our video skimming selection procedure is based on the video structure information detected with this method, which is shown in the next chapter.

4.6 Summary

A video has its intrinsic structure just like an article does. In this chapter, we describe our work done on analyzing the structure information of the video

however, since they are composed of several video shot groups, their content complexity should relate with both the number of video shot groups it contains and the lengths of the member video shot groups. To quantitatively evaluate the complexity of a loop video scene based on these two factors, we define the content entropy of a video scene and employ it as the measure of content complexity, as shown in Equation 5.1:

$$Entropy(Sc_i) = \sum_{Sg_j \, 2 \, Sc_i} -\frac{l_{Sg_j}}{l_{Sc_i}} \log_2(\frac{l_{Sg_j}}{l_{Sc_i}}). \tag{5.1}$$

5.1.2 Target Skim Length Assignment

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 of the video scenes in the video as follows:

Step 1. For each progressive scene Sc_x,

$$Sl_x = l_{Sc_x} \times r_s. \tag{5.2}$$

If Sl_x is less than the preset threshold t_1, we discard scene Sc_x as short skimmings do not make sense to people.

Step 2. Suppose that after we have assigned a part of the L_{vs} to some progressive scenes, the left skim length is L^0_{vs}. Then, for the loop scenes $\{Sc_1...Sc_n\}$,

$$Sl_i = L^0_{vs} \times \frac{Entropy(Sc_i) \times l_{Sc_i}}{\sum_{j=1}^{n} Entropy(Sc_j) \times l_{Sc_j}}. \tag{5.3}$$

In a similar manner, we discard Sc_i if Sl_i is less than a preset threshold t_2.

Step 3. Suppose that after Step 2, the remaining loop scenes are $\{Sc_1^0 ... Sc_m^0\}$, then for the remaining video scenes we set

$$Sl_i = L_{vs}^0 \times \frac{Entropy(Sc_i^0) \times l_{Sc_i'}}{\sum_{j=1}^{m} Entropy(Sc_j^0) \times l_{Sc_j'}}. \tag{5.4}$$

The above skim length assignment algorithm ensures that more important scenes are assigned with more skim length, thus the balanced content coverage can be achieved. Moreover, loop scenes are assigned with more skim length, since they are regarded as more important than the progressive scenes.

5.2 Graph Modelling of Video Scenes

Coherency is another important goal that the video skimming should achieve. If a lot of breaks are included in the video skimming, the user may feel that it is jumpy and less enjoyable. Since video shots are continuous image sequences and they are the building block of videos, in our work, we select complete video shots as the elements in the video skimming to guarantee the coherence of the video skimming. Moreover, we employ nrs strings as skim candidate to achieve better coherency.

With each scene's target 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 be able to cover both the visual diversity and the temporal distribution of the original video scene; meanwhile, the coherency of the video skim should be ensured. To achieve all these objectives simultaneously, we model each video scene with a graph based on the video nrs shot strings it contains, then we select the skimming video shots by performing optimization on that graph.

5.2.1 Decompose the Video Scene into Candidate Video Strings

Since the non-repetitive string (nrs strings) is a longer coherent part of a video and it contains no visual redundancy, it is a very good candidate to be selected into the video skim. To model the scene as a graph, we first need to decompose the video scene into a set of non-overlapped nrs strings $Nrs_{l_{\mathrm{str}}}$. However, there are still some problems to meet. First, the longer the nrs strings are, the more coherent the final selected video skim will be. On the other hand, if the candidate nrs strings are too long, the content coverage might not be able to get guaranteed. Another problem is that there may be more than one method to decompose the video scene into a set of video shot strings.

So we propose the following bounded decomposition process for a video scene:

1. We specify an integer l^u_{str}, which is the upper bound of shot number in the nrs strings to be decomposed from the video scene.

2. We decompose the video scene shots into a set of nrs strings whose length is upper bounded by l^u_{str}. Given a video scene Sc_i, we note the nrs string set decomposed from the scene as Nrs_k if we set the parameter l^u_{str} with value k. Note that there are more than one possible methods to decompose the video scene into a set of nrs strings. For simplicity, every time we decompose the first longest nrs string from left to right. For example, the Nrs_3 set for a video scene $\{1245141316\}$ should be $\{124, 514, 13, 16\}$.

We can use the integer l^u_{str} to control the coherence of the extracted video skim. As a special case, the Nrs_1 set of a scene is just all the video shots it contains. The longer l^u_{str} is, the more coherency will the selected skim be, but the content coverage might decrease accordingly.

5.2.2 The Spatial-Temporal Relation Graph

Based on the shot nrs strings we detect from the video shot list, we define the spatial-temporal relation graph as follows:

The spatial-temporal relation graph $G_{st}(V, E)$ is a graph defined on a video shot string set $S_{sh} = \{str_1, ...str_n\}$ such that:

1. $G_{st}(V, E)$ is a complete graph.

2. Each vertex $v_i \in V$ is corresponding to a video shot string str_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 string str_i.

3. On each edge $e_{ij} \in E$, there is an edge weight $w_{e_{ij}}$ which is equal to the spatial-temporal dissimilarity function $Dis(str_i, str_j)$ between video shot strings str_i and str_j. The direction of edge e_{ij} is from the temporally earlier shot string to the temporally later video shot string. Thus G is acyclic.

A simple example of the spatial-temporal relation graph on a scene is shown in Figure. 5.2. The vertex weight (shot string length) and edge weight are shown as numbers in the brackets associated with the vertices and edges in the graph.

To determine the value on each edge, we define the spatial-temporal dissimilarity function between two video shot strings str_i, str_j as:

$$Dis(str_i, str_j) = 1 - sim(str_i, str_j) \times e^{k \; d_T(str_i, str_j)}, \tag{5.5}$$

and

$$w_{e_{ij}} = Dis(str_i, str_j). \tag{5.6}$$

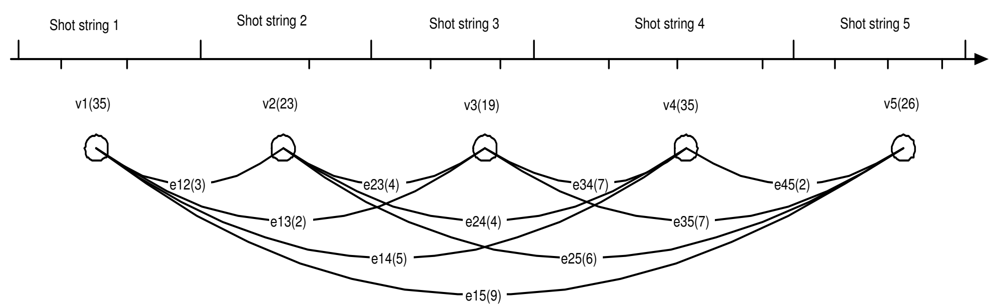

Figure 5.2: Spatial temporal dissimilarity graph on five shot strings

Here $sim(str_i, str_j)$ can be any visual similarity measure between video shot strings, and here we use the definition given in the previous chapter: $d_T(str_i, str_j)$ is the temporal distance between the temporal middle points of video strings str_i and str_j, in terms of frame number. k is the parameter to control the slope of the exponential function, also in terms of frame number. An example of dissimilarity function between video shots in a sample video is shown in Figure 5.3:

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. Such a dissimilarity definition will guarantee those visually dissimilar video shot strings are chosen, and it can guarantee that the chosen video shot strings will be well distributed on the time axis, so that a good temporal coverage of the original contents can be achieved.

5.2.3 The Optimal Skim Problem

Given the target skimming length L_{vs}, we can search a path in the spatial-temporal graph then use the video shots corresponding to the vertices 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 shot strings $\{str_{x_1}, ...str_{x_n}\}$,

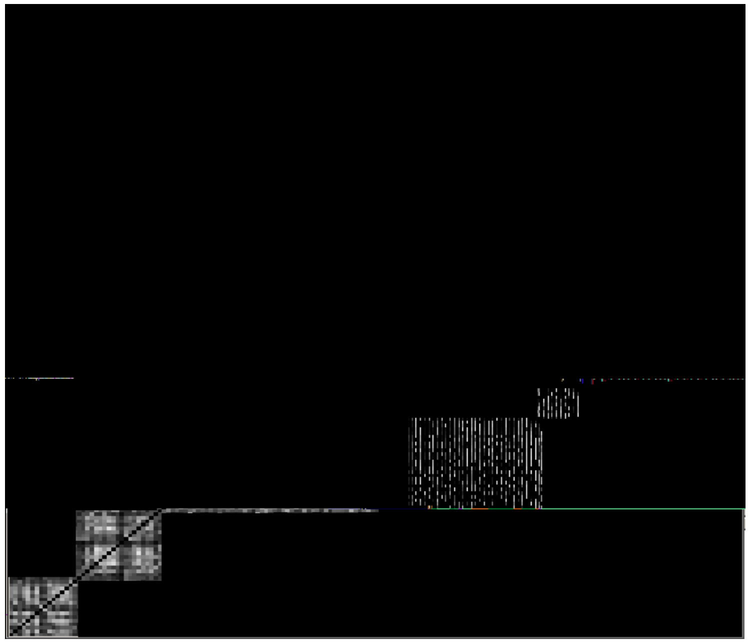

Figure 5.3: The spatial-temporal dissimilarity function for a sample video with seven scenes

which is a video skimming whose total length is the summation of the weights on the vertices $v_{x_1}, ...v_{x_n}$ in the path. We let $VWS(p_i)$ 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 optimal 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 shot strings. Second, we want $VWS(p_s)$ to be as close to L_{vs} as possible, but not to exceed it. So the problem is a constrained longest path problem. The result path varies with different target skim lengths, as shown in Figure 5.4.

We combine the above two goals in the objective function f^{st}_{obj}, which is described in the following video skimming generation problem.

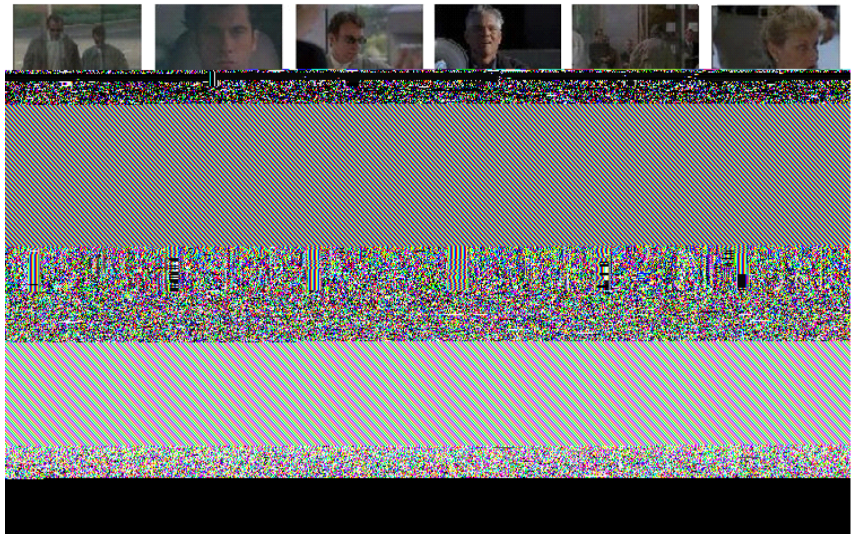

Figure 5.7: An example of static summary

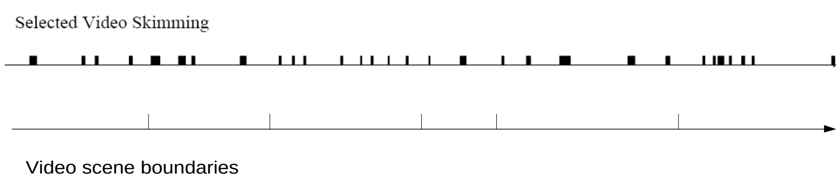

Figure 5.8: Temporal distribution of selected video shots

From the static video summary we observe that the visually dissimilarity video shots have been selected; also the video shots distribute quite evenly on the temporal axis, shown in Figure 5.8. Thus we make the conclusion that the graph optimization-based algorithm is able to simultaneously guarantee the visual content diversity and temporal coverage.

5.6 Summary

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 graph based 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 sub-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.

Chapter 6

Video Content Annotation and Semantic Video Summarization

In Chapter 4 and Chapter 5 we describe our video skimming generation approach based on graph modelling and optimization. This approach is able to guarantee both the temporal coverage and visual diversity of the video skimming. However, just like most of the traditional video skimming generation methods, the graph-optimization approach still relies on the low-level features of the original video. Therefore, it may not be able to guarantee that the generated video skimming contains the semantically important contents.

In this chapter, we describe a dynamic video skimming generation framework utilizing high-level video semantic information. Figure 6.1 shows the overview of our framework. We first segment the video into video shots, and then we annotate each of the shots with a semi-automatic annotation tool. To guarantee the balanced content coverage, we determine the scene boundaries by video structure analysis, and then assign appropriate target skimming length for each scene. For each video shot, an importance value is calculated according to the Mutual Reinforcement Principle [78], 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 obtain the semantic content description of video shots then employ the Mutual reinforcement principle to calculate a global importance rank value for each shot. Based on the importance values, we can ensure that the semantically important contents can be covered by the video skimming. Second, we analyze the shot arrangement patterns and select shot strings to form a more coherent video skimming. Our method guarantees both the content coverage and visual coherence of the video skimming, which are neglected by most other approaches.

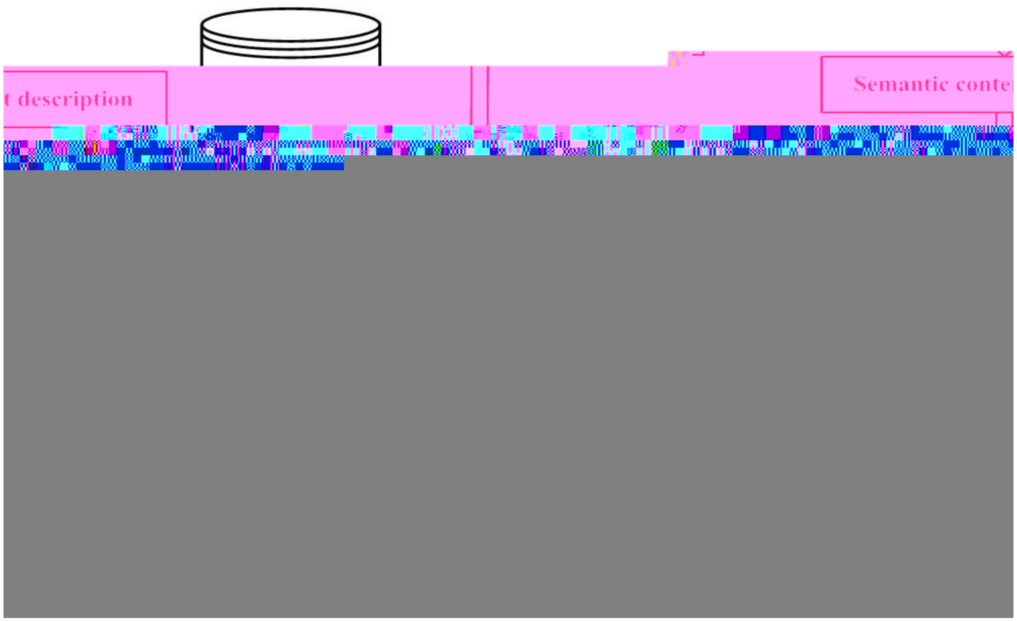

Figure 6.1: Overview of the semantic video summarization framework

The chapter is organized as follows. In Section 6.1 we describe our video annotation procedure. In Section 6.2 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 6.3 we present our video skim generation scheme based on shot arrangement analysis. Section 6.4 we show some experimental results. Finally, we make conclusion in Section 6.5.

6.1 Semantic Video Content Annotation

6.1.1 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. With the methods described in Chapter 3, 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 each detected video shot sh_i, we use its first frame $kf_{i_{\mathsf{begin}}}$ and the last frame $kf_{i_{\mathsf{end}}}$ as the key frames to represent its visual content.

6.1.2 Semi-Automatic Video Shot Annotation

Given the detected video shots, we can define the content description template, then create semantic annotation for each of 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 chapter, 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 events

measure sim_{ij}:

$$sim_{ij} = t \times sim^l_{ij} + (1 - t) \times sim^s_{ij}. \tag{6.3}$$

and based on measure sim_{ij} we can use the window sweeping algorithm in Chapter 3 to find the video shot group and video scene boundaries.

6.2.2 Video Structure and Video Edit Process

The video editing process, described in [20], is as follows. 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, the process is shown in Figure 6.3. For example, to depict a conversation, there should be some overview shots showing all the people involved at the beginning and the ending 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 (person) but since they might be shot from different angles so they might not be able to be grouped together by analyzing the low-level features. However, 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 depicting 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 between video group and the 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 arranges the semantic shot groups. The pattern of the shot arrangement will be analyzed in Section 6.4.

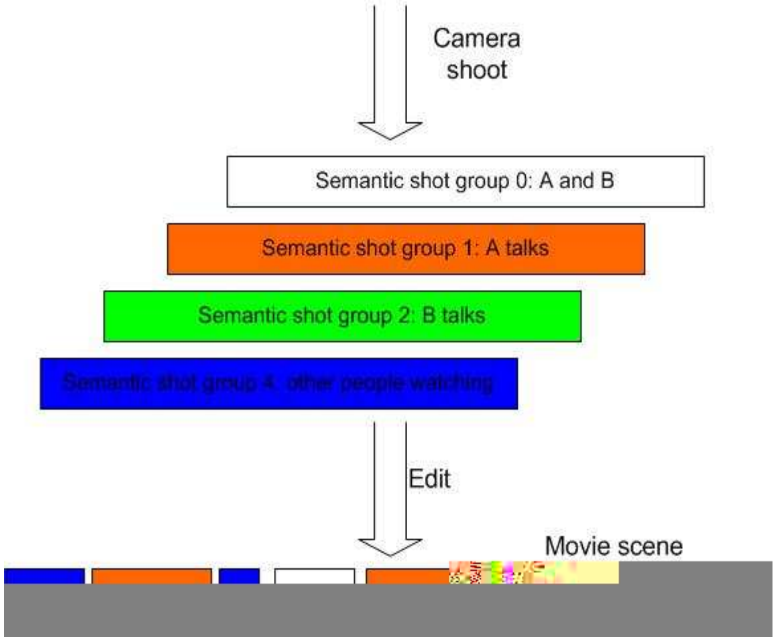

Figure 6.3: Movie edit process

6.2.3 Mutual Reinforcement and Semantic Video Shot Group Detection

Given a video scene composed by a set of annotated video shots and a set of video annotation concept terms and the corresponding contexts, we need to measure the relative importance of each video shots and each different concept term. We employ the following mutual reinforcement principle [78] to detect the semantic video shot groups and give a importance evaluation for each detected video shots. 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 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 a 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

becomes good candidates for video skimming since it is composed by video shots depicting non-repetitive contents, and is a coherent part of the original video. By scanning the video shot string we can easily get all k-nrs strings for all integer k.

We formulate the video skimming generation problem as Problem 6.1.

Problem 6.1 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. $\sum_j ip_{nrs_j}$ 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 Problem 6.1, we propose a greedy method algorithm, which is described in Algorithm 6.1:

Algorithm 6.1 continues selecting the most uncovered important nrs strings into the video skimming, and discards 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 coherency can be simultaneously achieved.

6.4 Experiments

To test the performance of our proposed approach, we have implemented the proposed video annotation and summarization framework then apply it to

Algorithm 6.1 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. $L_{nrs_{\text{opt}}} < L_{vs}$
  2. $\forall nrs_i \in N$ and $L_{nrs_{\text{i}}} < L_{vs}$, $ip_{nrs_{\text{opt}}} \geq ip_{nrs_{\text{i}}}$

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

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

some movie clips. We employed a PC with 2.0G hz P4 CPU and 512Mb RAM on the Win2000 OS as the test bed. The system is implemented with Microsoft VC++. The weight parameter t 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, for each test clip, two video skimmings generated at skim rate 0.15 and 0.30 are extracted. Details about the video clips are shown in Table 6.1.

To evaluate the quality of the generated video skimming, we employ two criterion: meaningfulnessand favorite. Since it is hard to objectively evaluate a video skimming, we use the following subjective test introduced in Chapter 5 to compare the performance of our new video skimming generation scheme and the method we proposed in [34]. 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, we ask the test users to answer several questions about the key events that the video depicts (who has done what?). The scores are scaled to $[0, 100]$. To evaluate favorite, we ask the users to select a "better" video skimming between the video skimmings generated by the two approaches, and the number of users who choose the skim as "better" is recorded as the favorite score. Figure 6.7 and Figure 6.8 show the average meaningfulness and favorite scores for the video skims generated by our proposed method and method in [34] respectively. The number of continuous video segments contained in the video skimming is employed as a measure of coherence. The experimental results are summarized in Table 6.1 (Mfn. means Meaningfulness, Fav. means favorite, N.S. means Number of Segments, SEM means the new semantic approach, GRA means our old graph based approach).

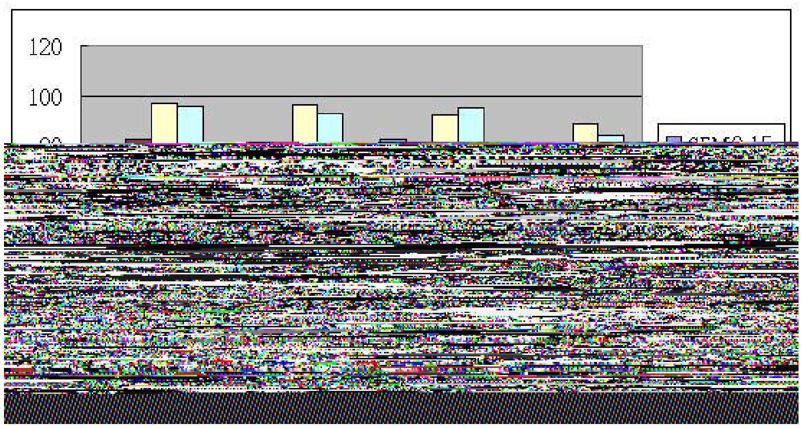

Figure 6.7: Meaningfulness Scores

Our experimental results are quite encouraging. In terms of the meaningfulness, at the skim rate 0.15, the proposed semantic video summarization method obtain a quite high mean score 77.95. At skim rate at 0.30, the score achieved is even higher. Moreover, in most cases, our new semantic approach has gained a higher score than our previous graph-optimization approach. In

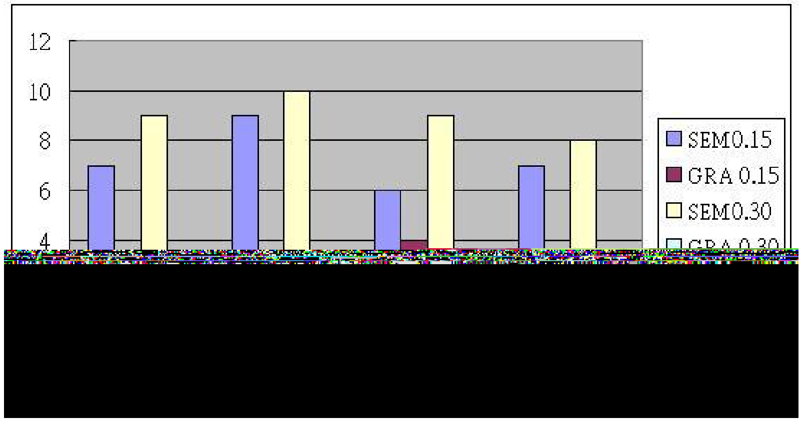

Figure 6.8: Favorite Scores

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 6.1: User test results. Scores for the new approach are **bold**

terms of favorite, we can see that although both video skimming is meaningful, most users would prefer the video skimming generated by the new method at both skim rates. The major reason should be that the new skim is more coherent. We also find that our new approach generates much less video segments than the previous approach, which greatly increases the video coherency. 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.

6.5 Summary

In this chapter, we illustrate a novel framework for semantic video summarization. We provide the users a semi-automatic system to help them annotating the video semantic contents efficiently. Then we combine the semantic information and structure information of the video, compute the semantic importance for each video shot, analyze the arrangement patterns of the video shots. Finally, we obtain a dynamic skimming by selecting the key video shot strings. The experimental results show that our approach ensures both the balanced content coverage and visual coherence. Experimental results show that the framework is effective in generating helpful video skimmings.

Chapter 7

Concluding Remarks

7.1 Summary

Video is getting more and more popular now than ever before, due to the rapid growth of the Internet bandwidth and the growing use of video in education, entertainment, and information sharing. Many organizations produce huge volume of video data everyday. Facing the massive data volume, end users find that it is inefficient to browse a favorite video from the Internet, and the content providers have to face the tedious work of managing the ever growing video database. The urgent problem brings a lot of attention to video summarization, which is a new technology intends to help the users to quickly grasp the major contents of a video by providing them concise and informative video content summaries.

The thesis presents our work done on automatic video summarization. We first specify three goals that a video summary with good quality should achieve:

1. **Conciseness**–For conciseness, the length of the generated video skimming should be within the user-specified length L_{vs}; the static summary should not contain too many images.

2. **Balanced content coverage** –As the video is a structured document, the video skimming should be able to represent the original contents

with balance. At the same time, both the visual diversity and the temporal coverage of the original contents should be reflected by the video skimming.

3. Visual coherence –One problem for traditional video skimming generation is that the user often feels that the video skimming is quite choppy. A good video skimming should increase the coherence of the video skimming while preserving the content coverage.

In conclusion, we have proposed an automatic video content analysis and summarization framework. First, we analyze the structure of the video. Based on the structure information, we propose a graph optimization-based approach to create video summaries that is able to achieve the above three goals. Our research work also comprises the initial work on video content annotation and semantic video summarization. To obtain and utilize the semantic information of the video, a semi-automatic video annotation system is built and video skimmings are generated based on the semantic content descriptions.

Our research work has the following contributions:

1. We have proposed several targets that a video summary with good quality should have, and we have achieved them in our framework.

2. We analyze the intrinsic video shot-shot group-scene structure of the video and employ the structure information to help us in video skimming generation. The balanced content coverage is thus guaranteed.

3. We model the video scenes into a graph based on the video shots contained in it, and select video shots by searching a constrained longest path in that graph. Our method is able to cover both the visual content coverage and the temporal content distribution simultaneously.

4. We propose a semantic video annotation and summarization framework to help the user to annotate the video shots semi-automatically. An

importance measure is derived from the semantic content descriptions based on mutual reinforcement principle. Video summaries are then generated with the semantic descriptions. The performance of our new approach is compared with our previous work, and we conclude that utilizing semantic information does help to enhance the quality of the produced video skimmings.

7.2 Future Work

In this thesis we describe the work we have done on video content analysis and summarization. A graph-optimization based video summarization framework and a semantic video summarization framework have been proposed. The frameworks themselves are quite flexible; many other features and constraints can be added into this framework as its extension. In the future, we may enhance the system by incorporating better feature analysis technique into our framework.

Currently our system provides only limited interactivity for the user. Our framework can be further extended by incorporating the user defined preference. The user may specify what he is specially interested in, or what he is not interested in to make a personalized video summary that is specially useful to the user. Different ways for different users to describe and express their preference during interaction with the video summarization system will be investigated.

For video semantic recovery, one possible way to obtain video semantic description is using the video production documents. During video document production, a lot of planning and paper work is produced, like movie screenplays, scripts, plans, etc. The information contained in them contains time, location, order, actor, dialog script, background music, etc., which is detailed enough for further processing. Moreover, recovering such detailed information from produced video automatically is far beyond up-to-date intelligent systems' abilities. One good practical way to resolve the semantic gap and obtain accurate semantic video descriptions is to find a method to better incorporate Mpeg-7 during video production and we can directly obtain video semantic description with very low cost, which is a alternative to obtain reliable video semantic descriptions.

Appendix A

Notations

Vd	A video.
$\{V_i\}$	A set of extracted video skimming segments.
$f_{obj}(\{V_i\})$	The objective function of the video skimming $\{V_i\}$.
$\{E_i\}$	The set of un-selected video segments.
L_{vs}	The target video skimming length.
$p(\{V_i\}, \{E_i\})$	The penalty function for the selected video segment set $\{V_i\}$ and the corresponding un-selected video segments $\{E_i\}$.
$N(\{V_i\}, \{E_i\})$	The new objective function for $\{V_i\}$ with consideration of the penalty function.
sh_i	The i_{th} video shot.
str_i	The i_{th} video shot string.
$sim(sh_i, sh_j)$	The visual similarity between video shots sh_i and sh_j.
$sim(str_i, str_j)$	The visual similarity between video shot strings str_i and

	str_j.
$d_T(sh_i, sh_j)$	The temporal distance between video shots sh_i and sh_j.
$d_T(str_i, str_j)$	The temporal distance between video shot strings str_i and str_j.
$Dis(sh_i, sh_j)$	The spatial-temporal dissimilarity between video shots sh_i and sh_j.
$Dis(str_i, str_j)$	The spatial- temporal dissimilarity between video shot strings str_i and str_j.
nrs	The non-repetitive video shot string.
l^u_{str}	The upper bound for the shot number of nrs shot strings.
Sg_i	The i_{th} video shot group.
Sc_i	The i_{th} video scene.
l_{sh_i}	The length of i_{th} video shot.
l_{Sg_i}	The length of i_{th} video shot group.
l_{Sc_i}	The length of i_{th} video scene.
$Entropy(Sc_i)$	The content entropy value of video scene Sc_i.
$G_{st}(V, E)$	The spatial-temporal relation graph based on a set of video shot strings.
L_{p_i}	The length of the path p_i.

$VWS(p_i)$	The vertex weight summation of a path p_i.
$f^{st}_{obj}(p_i)$	The objective function for a path p_i in the spatial-temporal relation graph.
$p^o_{v_x,l_r}$	The optimal path begin with vertex v_x with the vertex weight summation upper-bounded by l_r.
p_s	The global optimal path corresponds to the video skimming shots.
$G^0_{st}(V,E)$	The static spatial-temporal relation graph based on a set of video shots.
$p^{0o}_{i,n}$	The optimal path begins with vertex v_i and the vertex number is n.
$L^{0o}_{i,n}$	The length of the path $p^{0o}_{i,n}$.
p^0_s	The global optimal path corresponds to the static video summary images.
$T = \{t_1...t_n\}$	The semantic concept term set.
$D = \{d_1...d_n\}$	The semantic content description set.
IP	The semantic importance vector. The i_{th} component ip_i is the importance value of video shot sh_i.
ip_{str_i}	The semantic importance value of shot string str_i.

Bibliography

[1] A. Aner and J. R. Kender. A unified memory-based approach to cut, dissolve, key frame and scens analysis. In *Proceedings of the IEEE International Conference on Image Processing*, pages 370–373, 2001.

[2] A. Aner and J. R. Kender. Video summaries through mosaic-based shot and scene clustering. In *Proceedings of European conference on Computer Vision*, number 2353 in LNCS, pages 388–402, 2002.

[3] N. Bagaguchi. Generation of personalized abstract of sports video. In