EURASIP Journal on Applied Signal Processing 2003:2, 160–169

Summarizing Audiovisual Contents of a Video Program

Yihong Gong
NEC Laboratories America, Inc., 10080 North Wolfe Road, SW3-350 Cupertino, CA 95014, USA
Email: ygong@ccrl.sj.nec.com

Received 19 March 2002 and in revised form 22 October 2002

In this paper, we focus on video programs that are intended to disseminate information and knowledge such as news, documentaries, seminars, etc, and present an audiovisual summarization system that summarizes the audio and visual contents of the given video separately, and then integrating the two summaries with a partial alignment. The audio summary is created by selecting spoken sentences that best present the main content of the audio speech while the visual summary is created by eliminating duplicates/redundancies and preserving visually rich contents in the image stream. The alignment operation aims to synchronize each spoken sentence in the audio summary with its corresponding speaker's face and to preserve the rich content in the visual summary. A Bipartite Graph-based audiovisual alignment algorithm is developed to efficiently find the best alignment solution that satisfies these alignment requirements. With the proposed system, we strive to produce a video summary that: (1) provides a natural visual and audio content overview, and (2) maximizes the coverage for both audio and visual contents of the original video without having to sacrifice either of them.

Keywords and phrases: video summarization, audiovisual summarization, partial audiovisual alignment, bipartite graph, minimum spanning tree, maximum bipartite matching.

1. INTRODUCTION

Video programs are voluminous, redundant, and are a time-sequential medium whose overall contents cannot be captured at a glance. The voluminous and sequential nature of video programs not only creates congestions in computer systems and communication networks, but also causes bottlenecks in human information comprehension due to the limited human information processing speed. In the past decade, great efforts have been made to relieve the computer and communication congestion problems. However, in the whole video content creation, storage, processing, delivery, and utilization loop, the human bottleneck problem has long been neglected. Without technologies enabling fast and effective content overviews, browsing through video collections or finding desired video programs from a long list of search results will remain arduous and painful tasks.

Automatic video content summarization is one of the promising solutions to the human bottleneck problem. A concise and informative video summary will enable the user to quickly figure out the general content of a video and help him/her to decide whether the whole video program is worth watching. On the Internet, a compact video summary can be used as a video thumbnail of the original video, which requires much less efforts to download and comprehend. For most home users with very limited network bandwidths, this type of video thumbnails can well prevent them from spending minutes or tens of minutes downloading lengthy video programs, only to find them irrelevant. For video content retrieval, a video summary will certainly save the user's time and effort to browse through large volumes of video collections and to spot the desired videos from a long list of search results.

There are many possible ways for summarizing video contents. To date, the most common approach is to extract a set of keyframes from the original video and display them as thumbnails in a storyboard. However, keyframes extracted from a video sequence are a static image set that contains no temporal properties nor audio information of the video. While keyframes are effective in helping the user to identify the desired shots from a video, they are far from sufficient for the user to get a general idea of the video content, and to judge if the content is relevant or not.

In this paper, we begin with the discussion of different types of video programs and their appropriate summarization methods. Then, we proceed to propose an audiovisual summarization system which summarizes the audio and the visual contents of the given video separately, and then integrates the two summaries with a partial alignment. The audio content summarization is achieved by using the

latent semantic analysis technique to select representative spoken sentences from the audio track, while the visual content summarization is performed by eliminating duplicates/redundancies and preserving visually distinct contents from the image track of the given video program. A bipartite graph-based audiovisual alignment algorithm is developed to efficiently find the best alignment solution between the audio and the visual summaries that satisfies the predefined alignment requirements. With the proposed system, we strive to produce a motion video summary for the original video that (1) provides a natural and effective audio and visual content overview and (2) maximizes the coverage for both audio and visual contents without having to sacrifice either of them. Such audiovisual summaries dramatically increase the information intensity and depth, and lead to a more effective video content overview.

2. RELATED WORK

To date, video content overview is mainly achieved by using keyframes extracted from original video sequences. Many works focus on breaking video into shots and then finding a fixed number of keyframes for each detected shot. Tonomura et al. used the first frame from each shot as a keyframe [1]. Ueda et al. represented each shot using its first and last frames [2]. Ferman and Tekalp clustered the frames in each shot and selected the frame closest to the center of the largest cluster as the keyframe [3].

An obvious disadvantage of the above equal-number keyframe assignment is that long shots in which camera pan and zoom, as well as object motion, progressively unveil the entire event will not be adequately represented. To address this problem, DeMenthon et al. proposed to assign keyframes of a variable number according to the activity level of the corresponding scene shot [4]. Their method represents a video sequence as a trajectory curve in a high-dimensional feature space and uses the recursive binary curve splitting algorithm to find a set of perceptually significant points to approximate the video curve. This approximation is repeated until the approximation error comes below the user's specified value. Frames corresponding to these perceptually significant points are then used as keyframes to summarize the video contents. As the curve splitting algorithm assigns more points to a larger curvature, this method naturally assigns more keyframes to shots with more variations.

Keyframes extracted from a video sequence may contain duplications and redundancies. In a TV program with two talking persons, the video camera usually switches back and forth between the two persons, with the insertion of some global views of the scene. Applying the above keyframe selection methods to this kind of video sequences will yield many keyframes that are almost identical. To remove redundancies from keyframes, Yeung et al. selected one keyframe from each video shot, performed hierarchical clustering on these keyframes based on their visual similarity and temporal distance, and then retained only one keyframe for each cluster [5]. Girgensohn and Boreczky also applied the hierarchical clustering technique to group the keyframes into as many clusters as specified by the user. For each cluster, a keyframe is selected such that the constraints of an even distribution of keyframes over the length of the video and a minimum distance between keyframes are met [6].

Apart from the above methods of keyframe selection, summarizing video contents using keyframes has its own limitations. A video program is a continuous recording of real-world events. A set of static keyframes by no means captures the dynamics and main content of the video program. In viewing a movie or a TV program, the user may well prefer a summarized motion video with a specified time length to a set of static keyframes.

There have been research efforts that strive to output motion video summaries to accommodate better content overviews. The CueVideo system from IBM provides the fast video playback function which plays long, static shots with a faster speed (a higher frame rate) and plays short, dynamic shots with a slower speed (a lower frame rate) [7]. However, this variable frame rate playback causes static shots to look more dynamic and dynamic shots to look more static, and therefore, it dramatically distorts the temporal characteristics of the video sequence. On the other hand, the Informedia system from CMU provides the video skim that strives to identify, and playback only the semantically important image segments along with the semantically important audio keywords/phrases in the video sequence [8]. The importance of each image segment is measured using a set of heuristic rules which are highly subjective and content specific. This rule-based approach has certainly put limitations for handling diversified video images. Yahiaoui, et al. also proposed a similar method that summarizes multi-episode videos based on statistics as well as heuristics [9].

3. THREE TYPES OF SUMMARIES

Video programs, such as movies, dramas, talk shows, and so forth, have a strong synchronization between their audio and visual contents. Usually what we hear from the audio track directly corresponds to what we see on screen, and vice versa. For this type of video programs, since synchronization between audio and image streams is critical, the summarization has to be either audiocentric or imagecentric. The audiocentric summarization can be accomplished by the following two steps. First, an audio summary is composed by selecting audio segments of the original video that contain either important audio sounds or semantically important speeches. Advanced audio/speech recognition and text analysis techniques can be applied here to accomplish the goal. To enforce the synchronization, the corresponding visual summary has to be generated by selecting the image segments corresponding to those audio segments forming the audio summary. Similarly, an imagecentric summary can be created by selecting representative image segments from the original video to form a visual summary, and then taking the corresponding audio segments to form the associated audio summary. For these types of summarizations, either audio or

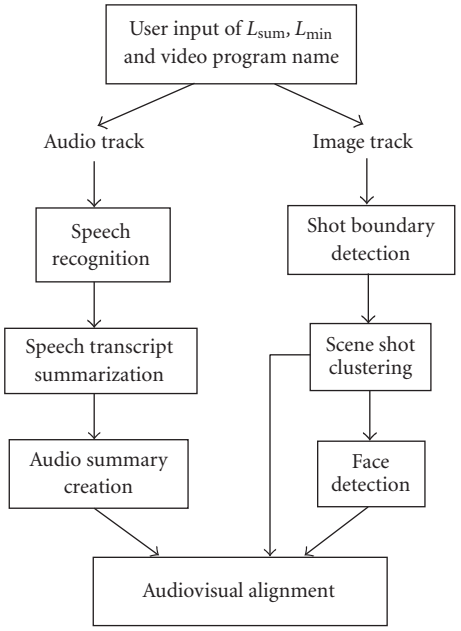


Figure 1: System-block diagram.

image contents of the original video will be sacrificed in the summaries.

Conversely, there are certain video programs that do not have a strong synchronization between their audio and visual contents. Consider a TV news program in which an audio segment presents information concerning the number of casualties caused by a recent earthquake. The corresponding image segment could be a close shot of a reporter in the field, of rescue teams working at the scene of a collapsed building, or of a regional map illustrating the epicenter of the quake. The audio content is related but does not directly refer to the corresponding image content. This kind of video production patterns are very common among such video programs as news, documentaries, seminars, and so forth. For this type of video programs, since there is no strong synchronization between the associated audio and visual contents, we propose to summarize the audio and the visual contents separately and then integrate the two summaries with a partial alignment. With this approach, we can maximize the coverage for both audio and image contents without having to sacrifice either of them. The following sections present our approach to the audiovisual summarization.

4. CREATING AUDIOVISUAL SUMMARIES

We strive to produce a motion video summary for the original video that (1) provides a natural and effective audio and visual content overview and (2) maximizes the coverage for both audio and visual contents of the original video without having to sacrifice either of them.

Figure 1 is the block diagram of the proposed audiovisual content summarization system. The summarization process starts by receiving the user's input of the video's filename and the two summarization parameters: the summary length L_{sum} and the minimum time length of each image segment L_{min} in the summary. Given L_{sum} and L_{min}, the maximum number of image segments a video summary can incorporate equals $N = L_{sum}/L_{min}$. Here, L_{min} provides the user with a control knob to choose between the breadth- and the depth-oriented visual summary. A small value for L_{min} will produce a breadth-oriented summary that consists of more image segments, each is shorter in length, while a large value for L_{min} will produce a depth-oriented summary that consists of less image segments, each is longer in length.

For the audio content summarization, speech recognition is first conducted on the audio track of the original video to obtain a speech transcript which includes the recognized sentences along with their time codes within the audio track. Next, the text summarization method described in [10] is applied to the speech transcript to create a text summary of the user specified length L_{sum}. This method creates a text summary by selecting sentences that best represent the main content of the speech transcript. An audio summary is then created by taking the audio segments corresponding to the sentences comprising the text summary and then concatenating them in their original time order.

As for the visual content summarization, an ideal summary should be the one that retains only visually important image segments of the original video. However, finding visually important image segments requires an overall understanding of the visual content, which is beyond our reach given the state of the art in current computer vision and image understanding techniques. On the other hand, it is relatively easy to identify duplicates and redundancies in a video sequence. For the purpose of visual content overviews, the video watching time will be largely shortened, and the original visual content will not be dramatically lost if we eliminate those duplicate shots and curtail those lengthy and static shots. Therefore, instead of relying on heuristically picking "important" segments for generating visual summaries, we choose to eliminate duplicates and redundancies while preserving visually distinct contents in the given video.

Our visual content summarization is composed of the following major steps (see Figure 1). First, shot boundary detection is conducted to segment the image track into individual scene shots. Next, shot clustering is performed to group scene shots into the required number $N = L_{sum}/L_{min}$ of clusters based on their visual similarities. The summary length L_{sum} is then divided into N time slots each of which lasts for L_{min} seconds, and each time slot is assigned to a suitable shot from an appropriate shot cluster. The decision of assigning a time slot to which shot from which cluster is made by the alignment process to fulfil the predefined alignment constraints (see Section 7 for detailed descriptions). Once a shot is assigned a time slot, its beginning portion (L_{min}-second long) is collected, and a visual summary is composed by concatenating these collected segments in their original time order. Moreover, face detection is conducted for each scene shot to detect the most salient frontal face that steadily appears in the shot. Such a face is considered as the speaker's face and will play an important role in the alignment operation.

For the alignment task, to achieve the summarization goals listed at the beginning of this section, we partially align the spoken sentences in the audio summary with the associated image segments in the original video. With video programs such as news and documentaries, a sentence spoken by an anchor person or a reporter lasts for ten to fifteen seconds in average. If a full alignment is made between each spoken sentence in the audio summary and its corresponding image segment, what we may get in the worst case is a video summary whose image part consists mostly of anchor persons and reporters. The summary created this way may look natural and smooth, but it is at the great sacrifice of the visual content. To create a content-rich audiovisual summary, we propose the following alignment operations. For each spoken sentence in the audio summary, if the corresponding image segment in the original video displays scenes rather than the speaker's face, perform no alignment operations and create the corresponding portion of the visual summary with shot segments from appropriate shot clusters; otherwise, align the spoken sentence with its corresponding image segment for the first $L_{\min}$ seconds, and then fill the remaining portion of the corresponding visual summary with shot segments from appropriate shot clusters. The decision of selecting which shot from which cluster is made by the alignment process to fulfill the predefined alignment constraints.

Detailed descriptions of the text summarization, scene shot clustering, and alignment operations are provided in the following sections.

5. TEXT SUMMARIZATION

In our audiovisual content summarization system, an audio summary is created by applying the text summarization method described in [10] to the speech transcript of the original audio track. This text summarization method uses the latent semantic analysis technique to select sentences which best represent the main content of the given text document. The process starts with the creation of a terms-by-sentences matrix $\mathbf{A} = [A_1 \ A_2 \ \cdots \ A_n]$, where each column vector $A_i = [a_{1i} \ a_{2i} \ \cdots \ a_{mi}]^T$ represents the term-frequency vector of sentence i in the document, and each element a_{ji} in A_i represents the weighted occurrence frequency of word j in sentence i. If there are a total of m terms and n sentences in the document, then we have an $m \times n$ matrix $\mathbf{A}$ for the document. Since every word does not normally appear in each sentence, the matrix $\mathbf{A}$ is usually sparse.

Given an $m \times n$ matrix $\mathbf{A}$, where without loss of generality $m \geq n$, the SVD of $\mathbf{A}$ is defined as in [11]

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{1}$$

where $\mathbf{U} = [u_{ij}]$ is an $m \times n$ column-orthonormal matrix whose columns are called left singular vectors, $\mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is an $n \times n$ diagonal matrix whose diagonal elements are nonnegative singular values sorted in descending order, and $\mathbf{V} = [v_{ij}]$ is an $n \times n$ orthonormal matrix whose columns are called right singular vectors. If $\text{rank}(\mathbf{A}) = r$, then $\mathbf{\Sigma}$ satisfies

$$\sigma_1 \geq \sigma_2 \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0. \tag{2}$$

The interpretation of applying the SVD to the terms by sentences matrix $\mathbf{A}$ can be made from two different viewpoints. From transformation point of view, the SVD derives a mapping between the m-dimensional space spanned by the term-frequency vectors and the r-dimensional singular vector space with all of its axes linearly independent. This mapping projects each column vector A_i in matrix $\mathbf{A}$, which represents the term-frequency vector of sentence i, to column vector $\psi_i = [v_{i1} \ v_{i2} \ \cdots \ v_{ir}]^T$ of matrix $\mathbf{V}^T$, and maps each row vector j in matrix $\mathbf{A}$, which tells the occurrence count of a word j in each of the documents, to row vector $\varphi_j = [u_{j1} \ u_{j2} \ \cdots \ u_{jr}]$ of matrix $\mathbf{U}$. Here, each element v_{ix} of ψ_i and u_{jy} of φ_j is called the index with the i'th and j'th singular vectors, respectively.

From the semantic point of view, the SVD derives the latent semantic structure from the document represented by matrix $\mathbf{A}$ [12]. This operation reflects a breakdown of the original document into r linearly independent base topics or concepts. A unique SVD feature which is lacking in conventional IR technologies is that the SVD is capable of capturing and modeling interrelationships among terms (word combination patterns). Each salient word combination pattern usually defines and describes a major topic/concept of the document. It can be demonstrated that, in the singular vector space [10, 12],

(1) each singular vector represents a salient topic;
(2) the singular vector with the largest corresponding singular value represents the topic that is the most salient in the document;
(3) the sentence having the largest index with singular vector i best describes the topic represented by this singular vector.

Leveraging the above observations, the text summarization using the latent semantic analysis technique is devised as follows:

(1) decompose the document D into individual sentences, and use these sentences to form the candidate sentence set S, and set $k = 1$;
(2) construct the terms by sentences matrix $\mathbf{A}$ for the document D;
(3) perform the SVD on $\mathbf{A}$ to obtain the singularvalue matrix $\mathbf{\Sigma}$ and the right singular-vector matrix $\mathbf{V}^T$. In the singularvector space, each sentence i is represented by the column vector $\psi_i = [v_{i1} \ v_{i2} \ \cdots \ v_{ir}]^T$ of $\mathbf{V}^T$;
(4) select the k'th right singular-vector from matrix $\mathbf{V}^T$;
(5) select the sentence which has the largest index value with the k'th right singular vector, and include it in the summary;
(6) if k reaches the predefined number, terminate the operation; otherwise, increment k by one and go to step (4).

In step (5) of the above operation, finding the sentence that has the largest index value with the k'th right singular vector

is equivalent to finding the column vector ψ_i whose k'th element v_{ik} is the largest. By the hypothesis, this operation is equivalent to finding the sentence describing the salient concept/topic represented by the k'th singular vector. Since the singular vectors are sorted in descending order of their corresponding singular values, the k'th singular vector represents the k'th important concept/topic. Because all the singular vectors are independent of one another, the sentences selected by this method contain the minimum redundancy.

6. CLUSTERING SCENE SHOTS

To find redundancies and duplicates in the image track of the original video, we segment the image track into individual scene shots, and then group them into $N = L_{\text{sum}}/L_{\text{min}}$ clusters based on their visual similarities. The generated shot clusters meet the following two conditions:

(1) all the shots within the same cluster are visually similar;
(2) any pair of shots from two different clusters is remarkably different in terms of its visual contents.

After the shot clustering process, we pick one shot from each cluster and take a segment of this shot to compose the visual summary. The decision of selecting which shot from which cluster is made by the alignment process to fulfill the predefined alignment constraints (see Section 7 for detailed descriptions). Because the shot clustering process groups visually similar shots into the same cluster and only a segment of one shot is selected from each cluster, this visual summarization method ensures that duplicates and redundancies are diminished and visually distinct contents are preserved within the visual summary.

In the literature, hierarchical clustering is a commonly used technique for dynamically grouping scene shots into a specified number of clusters. However, this technique does not meet the above conditions for the following reasons:

(1) the computation time is $O(n^3)$ for building up the entire hierarchy, where n is the number of scene shots;
(2) when the clustering hierarchy reaches certain heights, some major, visually different scene shots will be merged together;
(3) at the lower layers of the clustering hierarchy, many visually similar shots have yet to be merged properly, and there are still some shot clusters that are visually similar to one another.

If the summarization process has to take clusters from layers described in either (2) or (3), this summary will either drop some major shots or will contain many duplicates.

To solve the above problems, we propose a novel method that performs scene shot clustering using the minimum spanning tree algorithm, together with the upper-bound and the lower-bound thresholds T_{high} and T_{low}.

Minimum spanning tree (MST) is a special kind of graph that connects all its vertices using the shortest path [13]. Let $G = (V, E)$ be a connected, undirected graph, where V is the set of vertices and E is the set of possible interconnections between pairs of vertices. For each edge $(u, v) \in E$, we define a weight $w(u, v)$ specifying the cost to connect u and v. MST of the graph $G = (V, E)$ is defined as $T = (V, P)$, where P is an acyclic subset $P \subseteq E$ that connects all the vertices and minimizes the total weight

$$w(P) = \sum_{(u,v)\in P} w(u, v). \tag{3}$$

Construction of MST T for a given graph G is not unique and could have multiple solutions.

For our scene shot clustering problem, MST T is constructed for such a graph $G = (V, E)$, where each vertex $v \in V$ represents a scene shot of the original video, and each edge $(u, v) \in E$ represents the distance between shots u and v in the image feature space. By definition, $T = (V, P)$ connects the shot set V with the shortest edge set P. Givena graph $G = (V, E)$, its MST T can be built in time $O(|E|lg|V|)$. The computation time is a dramatic improvement comparing to the time $O(|V^3|)$ required for the hierarchical clustering.

Once an MST $T = (V, P)$ is constructed for the shot set V, we sort all the edges in P in descending order of their lengths. As T is a tree structure, which means that any two vertices in T are connected by a unique simple path, a cut at any edge will break the tree into two subtrees. Therefore, if N shot clusters are required to compose the video summary, we can cut T at its top $(N - 1)$ longest edges to obtain N subtrees and to use each subtree to form a shot cluster. Let L_{N-1} denote the length of the $(N - 1)'$th longest edge of T. All of the N subtrees obtained above have a property that the distance between an arbitrary vertex and its nearest neighbor is less than, or equal to, L_{N-1}. Because the $(N - 1)'$th longest edge of T defines the upper-bound edge of the subsequent N subtrees, to highlight its special role in the shot clustering process we call it the threshold edge for cutting the MST.

To achieve a shot clustering result meeting the two conditions listed at the beginning of this section using MST, we must put certain restrictions on cutting the tree. Our experiments have shown that, using image features with a reasonable discrimination power, we can easily find an upper-bound threshold T_{high} and a lower-bound threshold T_{low} that divide pairs of scene shots into three categories:

(1) the two shots are completely different when their distance in the feature space exceeds T_{high};
(2) the two shots are similar when their distance in the feature space is smaller than T_{low};
(3) when the distance is between T_{high} and T_{low}, the two shots could be judged as either similar or different depending on how strict the similarity criterion is.

The upper-bound and the lower-bound thresholds actually have created an ambiguous zone of judgment. This ambiguous zone provides us with a threshold range for cutting the MST. To ensure that the clustering process does not separate visually similar shots into different clusters nor merge completely different shots into the same clusters, the length of the threshold edge for cutting the MST must be between T_{high} and T_{low}.

Under the extreme circumstances where the required number of clusters N is either very high or very low, and cannot be generated without breaking the two conditions, we use either T_{high} or T_{low} to generate clusters of number N' that best approaches N. If $N' > N$, we let the alignment process to determine which cluster to keep and which cluster to discard. If $N' < N$, we use N' instead of N and evenly assign the total summary length L_{sum} among the N' shot clusters.

7. ALIGNMENT OPERATIONS

Let $A(t_i, \tau_i)$ and $I(t_i, \tau_i)$ denote the audio and image segments that start at time instant t_i and last for τ_i seconds, respectively. The alignment operation consists of the following two main steps:

(1) for a spoken sentence $A(t_i, \tau_i)$ in the audio summary, check the content of its corresponding image segment $I(t_i, \tau_i)$ in the original video. If $I(t_i, \tau_i)$ shows a closeup face, and this face has not been aligned with any other component in the audio summary, align $A(t_i, \tau_i)$ with $I(t_i, \tau_i)$ for L_{min} seconds. Otherwise, do not perform the alignment operation for $A(t_i, \tau_i)$. This L_{min}-second alignment between $A(t_i, \tau_i)$ and $I(t_i, \tau_i)$ is called an alignment point;

(2) once all the alignment points are identified, evenly assign the remaining time period of the summary among the shot clusters which have not received any playback time slot. This assignment must ensure the following two constraints.

Single-assignment constraint. Each shot cluster can receive only one time slot assignment.

Time-order constraint. All the image segments forming the visual summary must be in original time order.

The following subsections explain our approach to realizing the above alignment requirements.

7.1. Alignment based on bipartite graph

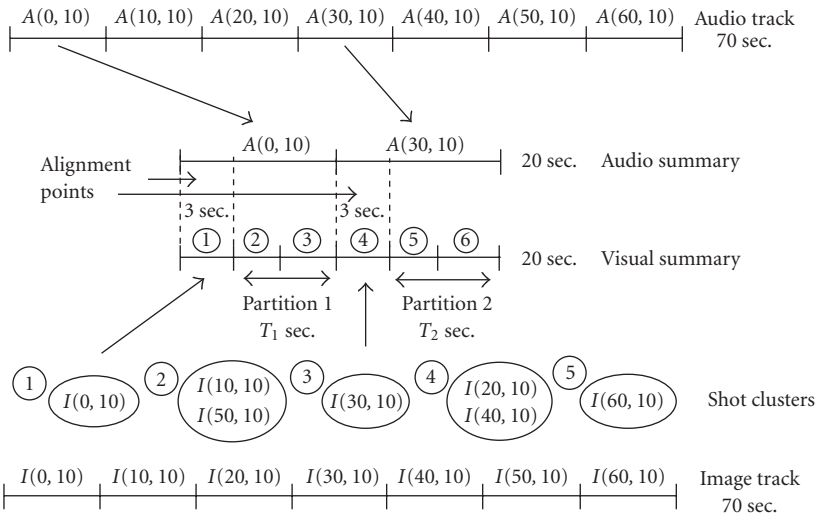
Assume that the whole time span L_{sum} of the video summary is divided by the alignment points into P partitions, and the time length of partition i is T_i (see Figure 2). Because each image segment forming the visual summary must be at least L_{min}-second long (a time duration of L_{min}-second long is called a time slot), partition i will be able to provide $S_i = \lceil T_i/L_{\text{min}} \rceil$ time slots, and hence the total number of available time slots becomes $S_{\text{total}} = \sum_{i=1}^{P} S_i$. Here, the problem becomes as follows: given a total of N shot clusters and S_{total} time slots, determine the best matching between the shot clusters and the time slots which satisfies the above two constraints. By some reformulation, this problem can be converted into the following maximum-bipartite-matching (MBM) problem [13]. Let $G = (V, E)$ represent an undirected graph, where V is a finite set of vertices and E is an edge set on V. A bipartite graph is an undirected graph $G = (V, E)$, in which V can be partitioned into two sets L and R such that $(u, v) \in E$ implies either $u \in L$ and $v \in R$ or $u \in R$ and $v \in L$. That is, all edges go between the two sets L and R. A matching is a subset of edges $M \subseteq E$ such that for any vertex pair (u, v), where $u \in L$ and $v \in R$, at most one edge of M connects between u and v. A maximum matching is a matching M such that for any matching M' we have $|M| \geq |M'|$.

To apply the MBM algorithm to our alignment problem, we use each vertex $u \in L$ to represent a shot cluster and each vertex $v \in R$ to represent a time slot. An edge (u, v) exists if a shot cluster u is able to take time slot v without violating the time-order constraint. If a shot cluster consists of multiple-scene shots, this cluster may have multiple edges that leave from it and enter different vertices in R. A maximum-bipartite-matching solution is a best assignment between all the shot clusters and the time slots. Note that a best assignment is not necessarily unique.

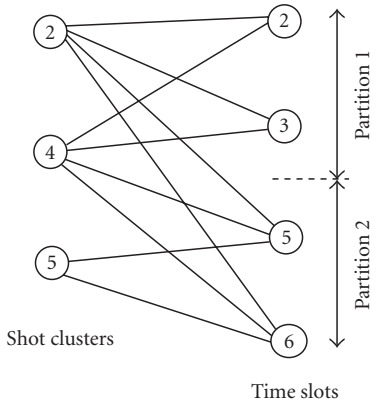
7.2. Alignment process illustration

Figure 2a illustrates the alignment process using a simple example. In this figure, the original video program is 70-second long and consists of 7 scene shots and 7 spoken sentences each of which lasts for 10 seconds. The user has set $L_{\text{sum}} = 20$ seconds and $L_{\text{min}} = 3$ seconds. Assume that the audio summarization has selected two spoken sentences $A(0, 10)$ and $A(30, 10)$, and that the shot clustering process has generated five shot clusters as shown in Figure 2a. As the audio summary is formed by $A(0, 10)$ and $A(30, 10)$, we must first examine the contents of the corresponding image segments $I(0, 10)$ and $I(30, 10)$ to determine whether the alignment operations are required. Suppose that $I(0, 10)$ and $I(30, 10)$ display the faces of the spoken sentences $A(0, 10)$ and $A(30, 10)$, respectively, and that $I(0, 10)$ and $I(30, 10)$ have not been aligned with other audio segments yet. Then, according to the alignment rules, $I(0, 10)$ will be aligned with $A(0, 10)$, and $I(30, 10)$ with $A(30, 10)$ for $L_{\text{min}} = 3$ seconds. Because $I(0, 10)$ and $I(30, 10)$ have been used once, they will not be used in other parts of the visual summary. By these two alignment points, the remaining time period of the visual summary is divided into two partitions, with each lasting for 7 seconds that can provide at most 2 time slots. Because there are three shot clusters and four time slots left for the alignment, we have a bipartite graph for the alignment task shown in Figure 2b. Since shot cluster 2 consists of two shots, $I(10, 10)$ and $I(50, 10)$, it could take a time slot in either partition 1 or partition 2. If $I(10, 10)$ is selected from cluster 2, it can take either time slot 2 or 3 in partition 1. On the other hand, If $I(50, 10)$ is selected, it can take either time slot 5 or 6 in partition 2. Therefore, we have four edges leaving from cluster 2, each entering time slots 2, 3, 5, and 6, respectively. Similarly, there are four edges leaving from cluster 4 and two edges leaving from shot cluster 5, respectively.

There are several possible maximum matching solutions for the bipartite graph in Figure 2b. Figure 3a shows one solution where the coarse lines represent the assignment of the shots to the time slots. Note that in this solution, time slot 3 remains unassigned. This example illustrates a fact that although the MBM algorithm will find a best matching between the available shot clusters and time slots, it may leave some time slots unassigned, especially when the number of



(a) Audiovisual alignment.



(b) Bipartite graph.

Figure 2: An example of the audiovisual alignment and the corresponding bipartite graph.

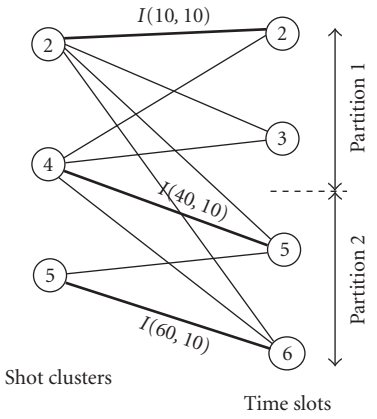
available shot clusters is less than that of available time slots. To fill these unassigned time slots, we loosen the single-assignment constraint, examine those clusters with multiple scene shots, and select an appropriate shot that has not been used yet, and that satisfies the time-order constraint. In the above example, the blank time slot 3 is filled using the shot $I(20, 10)$ in cluster 4 (coarse-dashed line in Figure 3b).

In case when the number of available shot clusters is more than that of available time shots, some shot clusters will not be assigned to time slots within the visual summary. The MBM algorithm determines which cluster to discard and which cluster to take during its process of finding the best matching solution.

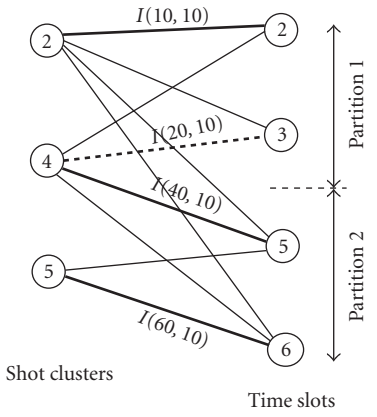
It is noticed that the MBM algorithm may generate some false solutions, and Figure 3c shows such an example. Here, because shot $I(60, 10)$ has been placed before shot $I(50, 10)$, it has violated the time-order constraint. However, this kind of false solutions can be easily detected and can be corrected by sorting the image segments assigned to each partition into their original time order. In the above example, the time order violation can be corrected by exchanging the two image segments assigned to time slots 5 and 6 in partition 2.

In a summary, step (2) of the alignment operation (Section 7) can be described as follows:

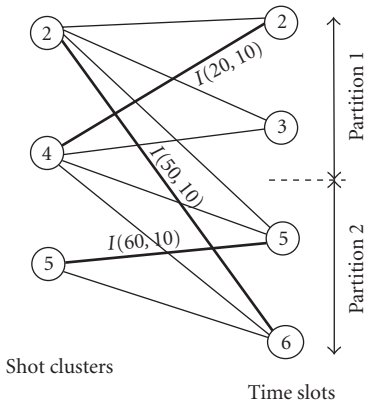
(1) after the alignment points have been identified, determine the number of shot clusters and time slots that are left for the assignment, and construct a bipartite graph accordingly;
(2) apply the MBM algorithm to find a solution;
(3) examine the solution with the time-order constraint and, if necessary, sort the image segments assigned to each partition into their original time order;
(4) if there exist unassigned time slots, examine those shot clusters with multiple scene shots, and select an appropriate shot that has not been used yet, and that satisfies the time-order constraint.



(a) Initial solution.



(b) Complete solution.



(c) False solution.

Figure 3: Alignment solutions: the coarse lines represent the assignment of the shot clusters to the time shots; the notation $I(\cdot, \cdot)$ on each coarse line tells which shot from the cluster has been selected, and assigned to the time slot.

8. SUMMARIZATION PERFORMANCES

Conducting an objective and meaningful evaluation for a video summarization method is difficult and challenging and is an open issue deserving more research. The challenges are mainly from the fact that research for video summarization is still at its early stage, and there are no agreed-upon metrics for performance evaluations. These challenges are further compounded by the fact that different people carry different opinions and requirements towards video summarization, making the creation of any agreed-upon performance metrics even more difficult.

Our audiovisual content summarization system has the following characteristics:

(1) the audio content summarization is achieved by using the latent semantic analysis (LSA) technique to select representative spoken sentences from the audio track;
(2) the visual content summarization is performed by eliminating duplicates/redundancies and preserving visually distinct contents from the image track;
(3) the alignment operation ensures that the generated audiovisual summary maximizes the coverage for both audio and visual contents of the original video without having to sacrifice either of them.

In [10], systematic performance evaluations have been conducted on the LSA-based text summarization method. The evaluations were carried out by comparing the machine-generated summaries with the manual summaries created by three independent human evaluators, and the F-value was used to measure the overlap degrees between the two types of summaries. It has been shown that the LSA-based text summarization method achieved the F-value in a range of 0.57 and 0.61 for multiple test runs, the performance compatible with the top-ranking state-of-the-art text summarization techniques [14, 15].

In the audiovisual content summarization process, as a visual summary is composed by first grouping visually similar shots into the same clusters, and then selecting at most one shot segment from each cluster, this visual summarization method ensures that duplicates and redundancies are diminished and visually distinct contents are preserved within the visual summary. Our evaluations have shown that 85% of the duplicates or visually similar shots have been properly grouped together using the clustering method based on the MST together with the upper-bound and the lower-bound thresholds.

On the other hand, the alignment operation partially aligns each spoken sentence in the audio summary to the image segment displaying the speaker's face, and fills the remaining period of the visual summary with other image segments. In fact, this alignment method is a mimic of a news video production technique commonly used by major TV stations. A common pattern for news programs is that an anchor person appears on the screen and reports the news for several seconds. After that, the anchor person continues his/her reports, but the image part of the news video switches to either a field or some related interesting scenes. By doing so, visual contents of news broadcast are remarkably enriched, and viewers will not get bored. On the other hand, by mimicking this news video production technique in our summarization process, we get an audiovisual content

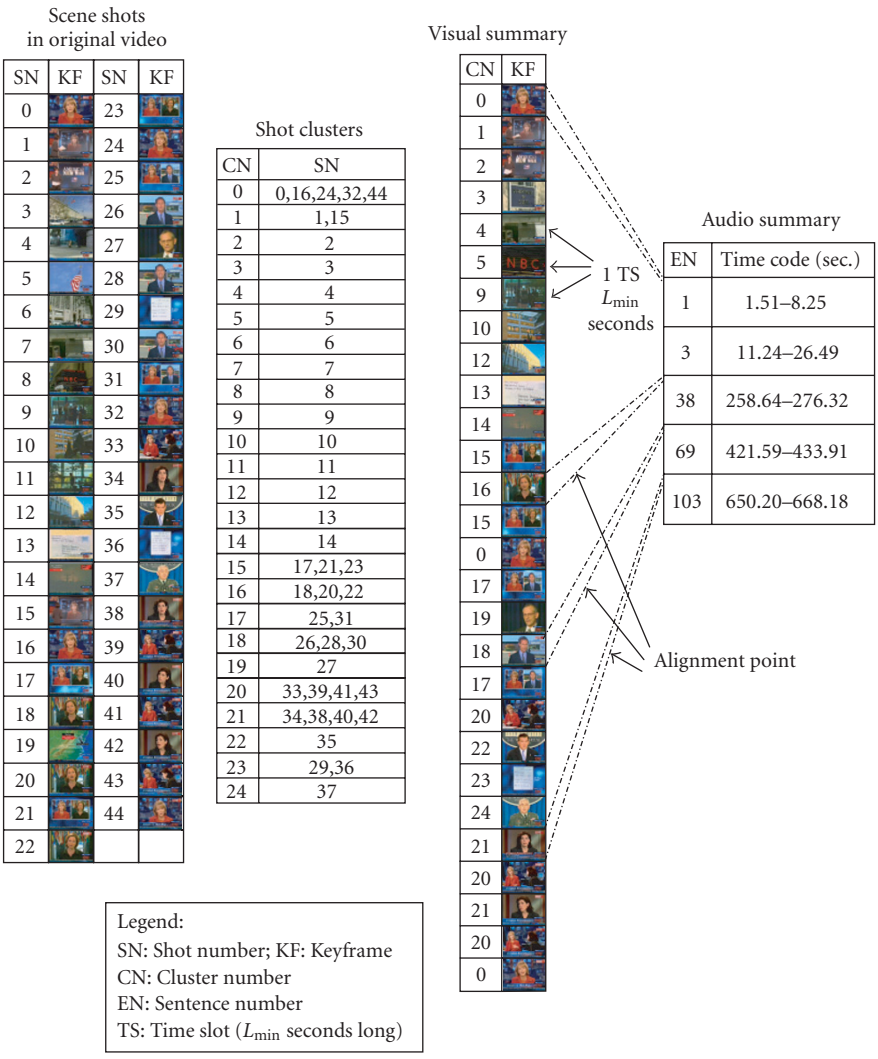


Figure 4: Audiovisual summarization of a 12-minute news program.

summary which provides a richer visual content and a more natural audiovisual content overview. Such audiovisual summaries dramatically increase the information intensity and depth, and lead to a more effective video content overview.

Figure 4 illustrates the process of summarizing a 12-minute CNN news program reporting the Anthrax threat after the September 11 terrorist attack. The news program consists of 45 scene shots in its image track, and 117 spoken sentences in its audio track. Keyframes of all the shots are displayed at the left-hand side of the figure in their original time order. Obviously, this news program contains many duplicated shots which arise from video cameras switching forth and back among anchor persons and field reporters.

The user has set $L_{sum} = 60$ seconds and $L_{min} = 2.5$ seconds. The shot clustering process has generated 25 distinct shot clusters, and the audio summarization process has selected sentences 1, 3, 38, 69, and 103 for composing the audio summary. The clustering result is shown by the table in the middle of Figure 4. It is clear from the clustering result that all the duplicated shots have been properly grouped into the appropriate clusters, and there is no apparent misplacement among the resultant clusters. Because the total time length of these five sentences equals 70 seconds, the actual length of the produced audiovisual summary exceeds the user specified summary length by 10 seconds.

Among the five sentences comprising the audio summary, four sentences have their corresponding image segments containing the speakers' faces. Each of these four audio sentences has been aligned with its corresponding image segment for $L_{min} = 2.5$ seconds. The dashed lines between the audio and the visual summaries denote these alignments.

With the actual $L_{sum} = 70$ seconds and $L_{min} = 2.5$ seconds, the audiovisual summary can accommodate 28 time slots. As there are a total of only 25 distinct shot clusters, some shot clusters were assigned more than one time slot (e.g., clusters 0, 15, 17, 20, 21). To find an alignment solution that fulfills the two alignment constraints listed in Section 7, several shot clusters were not assigned any time

slots and were consequently discarded by the alignment algorithm (e.g., clusters 6, 7, 8, 11). The configuration of the audio and the visual summaries are displayed at the right-hand side of Figure 4.

Video summarization examples can be viewed at http://www.ccrl.com/~ygong/VSUM/VSUM.html.

REFERENCES

[1] Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakata, "Videomap and videospaceicon: tools for anatomizing video content," in *Proc. ACM INTERCHI '93*, pp. 131–136, Amsterdam, The Netherlands, April 1993.

[2] H. Ueda, T. Miyatake, and S. Yoshizawa, "Impact: An interactive natural-motion-picture dedicated multimedia authoring system," in *Proc. ACM SIGCHI '91*, pp. 343–350, New Orleans, La, USA, April 1991.

[3] A. Ferman and A. Tekalp, "Multiscale content extraction and representation for video indexing," in *Multimedia Storage and Archiving Systems II*, vol. 3229 of *Proceedings of SPIE*, pp. 23–31, Dallas, Tex, USA, November 1997.

[4] D. DeMenthon, V. Kobla, and D. Doermann, "Video summarization by curve simplification," Tech. Rep. LAMP-TR-018, Language and Media Processing laboratory, University of Maryland, College Park, Md, USA, July 1998.

[5] M. Yeung, B. Yeo, W. Wolf, and B. Liu, "Video browsing using clustering and scene transitions on compressed sequences," in *Multimedia Computing and Networking*, vol. 2417 of *Proceedings of SPIE*, pp. 399–413, San Jose, Calif, USA, 1995.

[6] A. Girgensohn and J. Boreczky, "Time-constrained keyframe selection technique," in *IEEE International Conference on Multimedia Computing and Systems*, vol. 1, pp. 756–761, Florence, Italy, June 1999.

[7] D. Ponceleon, A. Amir, S. Srinivasan, T. Syeda-Mahmood, and D. Petkovic, "CueVideo: automated multimedia indexing and retrieval," in *Proc. ACM Multimedia*, vol. 2, Orlando, Fla, USA, October 1999.

[8] M. A. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding techniques," in *Proc. CVPR '97*, pp. 775–781, San Juan, Puerto Rico, USA, June 1997.

[9] I. Yahiaoui, B. Merialdo, and B. Huet, "Generating summaries of multi-episodes video," in *Proc. IEEE International Conference on Multimedia and Expo*, Tokyo, Japan, August 2001.

[10] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in *Proc. ACM SIGIR*, New Orleans, La, USA, September 2001.

[11] W. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

[12] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[13] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, Mass, USA, 1990.

[14] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell, "Summarizing text documents: Sentence selection and evaluation metrics," in *Proc. ACM SIGIR '99*, Berkeley, Calif, USA, August 1999.

[15] T. Firmin and B. Sundheim, "TIPSTER/SUMMAC summarization analysis participant results," in *Proc. TIPSTER Text Phase III Workshop*, Washington, D.C., USA, 1998.

Yihong Gong received his B.S., M.S., and Ph.D. degrees in electronic engineering from the University of Tokyo in 1987, 1989, and 1992, respectively. He then joined the Nanyang Technological University of Singapore, where he worked as an Assistant Professor at the School of Electrical and Electronic Engineering for four years. From 1996 to 1998, he worked for the Robotics Institute, Carnegie Mellon University as a Project Scientist, and was a Principal Investigator for both the Informedia Digital Video Library project and the Experience-On-Demand project funded in multimillion dollars by NSF, DARPA, NASA, and other government agencies. Now, he works as a Senior Research Staff for NEC Laboratories America, leading the Rich Media Processing Group. His research interests include image and video analysis, multimedia database systems, and machine learning.