Video Summarization Using Singular Value Decomposition

Yihong Gong, and Xin Liu
C&C Research Laboratories, NEC USA, Inc.
110 Rio Robles, San Jose, CA 95134, U.S.A.
{ygong,xliu}@ccrl.sj.nec.com

Abstract

In this paper, we propose a novel technique for video summarization based on the Singular Value Decomposition (SVD). For the input video sequence, we create a feature-frame matrix **A**, *and perform the SVD on it. From this SVD, we are able to not only derive the refined feature space to better cluster visually similar frames, but also define a metric to measure the amount of visual content contained in each frame cluster using its degree of visual changes. Then, in the refined feature space, we find the most static frame cluster, define it as the content unit, and use the content value computed from it as the threshold to cluster the rest of the frames. Based on this clustering result, either the optimal set of keyframes, or a summarized motion video with the user specified time length can be generated to support different user requirements for video browsing and content overview. Our approach ensures that the summarized video representation contains little redundancy, and gives equal attention to the same amount of contents.*

1 Introduction

Video images are voluminous, redundant, and unstructured data streams that span along the time sequence. Although video images convey real-world scenes most vividly, it is always a painful task to find either the appropriate video sequence, or the desired portion of the video from a large video data collection. The situation becomes even worse on the Internet. To-date, more and more web sites provide video images for news broadcasting, entertainment, or product promotions. However, with very limited network bandwidth for most home users, people spend minutes or tens of minutes to download voluminous video images, only to find them irrelevant. To turn unstructured, voluminous video images into exciting, valuable information resources, browsing and summarization tools that would allow the user to quickly get an idea of the overall content of video footage become indispensable.

Currently, most video browsing tools use a set of keyframes to provide content summary of a video sequence. Many systems use a constant number of keyframes for each detected scene shot, while others assign more keyframes to scene shots with more changes. There are also systems that remove redundancies among keyframes by clustering the keyframes based on their visual similarity. An important missing component in the existing video browsing and summarization tools is the mechanisms to estimate how many keyframes would be sufficient to provide a good, nonredundant representation of the video image. The simple methods that assign fixed number of keyframes to each scene shot suffer from poor video content representations, while the more sophisticated approaches that adaptively assign keyframes according to the activity levels often rely on the users to provide either the number of keyframes to be generated, or some thresholds (e.g., similarity distance or time interval between keyframes) used to generate keyframes. The user must go through several rounds of interactions with the system to obtain an appropriate set of keyframes. This kind of approach is acceptable when the user browses a small set of video images on a local workstation, while it becomes prohibitive when video images are accessed through the Internet with very limited bandwidth, or when a video summary must be created for each video image in a large-scale video database.

In this paper, we propose a novel technique that strives to automatically create an optimal and nonredundant video summarization based on the Singular Value Decomposition (SVD). Our approach generates video summaries based on the proposed visual content metric derived from the SVD properties rather than relying on shot boundaries of the video sequences. This ensures that the summarized representation of the original video contains little redundancy, and gives equal attention to the same amount of contents.

2 Related Work

To date, video browsing and content overview are mainly achieved by using keyframes extracted from original video sequences. Many works concentrate 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 variant 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 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, 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].

To create a concise summary of video contents, it is very important to ensure that the summarized representation of the original video (1) contains little redundancy, and (2) gives equal attention to the same amount of contents. While the sophisticated keyframe selection methods in the literature address these two issues to variant extents, they often rely on the users to provide either the number of keyframes to be generated, or some thresholds (e.g., similarity distance between keyframes, approximation errors) used to generate keyframes. The optimal set of keyframes will not be available without several rounds of trials. This could become prohibitive when video images are accessed through the Internet with very limited bandwidth, or when a keyframe-set must be created for each video image in a large-scale video database.

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

In the following section of this paper, we propose a novel technique that aims to automatically create an optimal and nonredundant summarization, and to support different user requirements for video browsing and content overview by outputting either the optimal set of keyframes, or a summarized version of the original video with the user specified time length.

3 The Proposed System

Our video summarization system uses the SVD as an important basis. The SVD is known for its capabilities of deriving the low dimensional refined feature space from a high dimensional raw feature space, and of capturing the essential structure of a data set in the feature space [7]. To reduce the number of frames to be processed by the SVD, we select a set of frames that are evenly spaced in the input video (one from every ten frames in out implementation). For each frame i in this sampling set, we create an m-dimensional feature vector A_i. Using A_i as a column, we obtain the feature-frame matrix $\mathbf{A} = [A_1 \ A_2 \ \cdots \ A_n]$. Performing SVD on this matrix $\mathbf{A}$ will project each frame i from the m-dimensional raw feature space into a κ-dimensional refined feature space (usually $\kappa \ll m$), where clustering of visually similar frames is performed. Our mathematical analysis has further revealed that in this refined feature space, there is a strong correlation between the degree of visual changes in a frame cluster and the positions at which its constituent frames are projected. For many video images, the degree of visual changes is a good indicator of the amount of visual content contained in the videos. Taking the video's image track only, a static video with almost no changes contains less visual content than a dynamic video with lots of changes. Based on this observation, we define the content value of a frame cluster using the positions of its constituent frames in the refined feature space. Then, in this refined feature space, we find the most static frame cluster, define it as the content unit, and use the content value computed from it as the threshold to cluster the rest of the frames. From each cluster, we select one frame that is closest to the center of the cluster as keyframe. Our approach assigns keyframes and creates the video summary based on the content value rather than relying on shot boundaries of the video sequence. This ensures that the summarized representation of the original video contains little redundancy, and gives equal attention to the same amount of contents.

To support different user requirements for video browsing and content overview, our system is able to output either the optimal set of keyframes, or a summarized motion video of the original video with the user specified time length.

3.1 Construction of Feature Vector

From a wide variety of image features, we selected color histograms to represent video frames. Histograms are very good for detecting overall differences in images, and are cost-effective for computing. Using cost-effective histograms here is to ensure feasibility and scalability of the system in handling long video sequences. In our system, we create three-dimensional histograms in the RGB color space with 5 bins for R,G, and B, respectively, resulting in a total of 125 bins. To incorporate spacial information of the color distribution, we divide each frame into 3×3 blocks, and create a 3D-histogram for each of the blocks. These nine histograms are then concatenated together to form a 1125-dimensional feature vector for the frame. Using the feature vector of frame i as i'th column, we create the feature-frame matrix $\mathbf{A}$ for the video sequence. Since a small image block does not normally contain all kinds of colors, matrix $\mathbf{A}$ is usually sparse. Therefore, SVD algorithms for sparse matrix can be applied here, which is must faster and memory efficient compared to regular SVD algorithms.

3.2 Singular Value Decomposition (SVD)

Given an $m \times n$ matrix $\mathbf{A}$, where $m \geq n$, the SVD of $\mathbf{A}$ is defined as [8]:

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

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 non-negative 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 rank($\mathbf{A}$)=r, then $\mathbf{\Sigma}$ satisfies

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

In our video summarization system, applying SVD to the feature-frame matrix $\mathbf{A}$ can be interpreted as follows. The SVD derives a mapping between the m-dimensional raw feature space spanned by the color histograms and the r-dimensional refined feature space with all of its axes linearly-independent. This mapping maps each column vector i in matrix $\mathbf{A}$, which represents the concatenated histogram of frame 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 the concatenated histogram entry j in each of the video frames, to row vector $\varphi_j = [u_{j1}\ u_{j2}\ \cdots\ u_{jr}]$ of matrix $\mathbf{U}$.

The SVD requires that matrix $\mathbf{A}$'s number of rows m is greater than or equal to its number of columns n. If the number of frames is more than the number of elements in each concatenated histogram, the SVD must be carried out on $\mathbf{A}^T$, and consequently, the role of matrix $\mathbf{U}$ and $\mathbf{V}$, which is explained above, will be exchanged. For simplicity, without loss of generality, only the processing of matrix $\mathbf{A}$ will be described in the following part of this paper.

The SVD has the following important property that has been widely utilized for text indexing and retrieval (See [9] for proof).

Theorem 1 *Let the SVD of matrix* $\mathbf{A}$ *be given by Eq.(1),* $\mathbf{U} = [U_1 U_2 \cdots U_n]$, $\mathbf{V} = [V_1 V_2 \cdots V_n]$, *and rank(*$\mathbf{A}$*)=r. Matrix* $\mathbf{A}_\kappa$ *(*$\kappa \leq r$*) defined below is the closest rank-*κ *matrix to* $\mathbf{A}$ *for the Euclidean and Frobenius norms.*

$$\mathbf{A}_\kappa = \sum_{i=1}^{\kappa} U_i \cdot \sigma_i \cdot V_i^T \tag{3}$$

The use of κ-largest singular values to approximate the original matrix with Eq.(3) has significant implications. Discarding small singular values is equivalent to discarding linearly semi-dependent or practically nonessential axes of the feature space. The truncated SVD, in one sense, captures most of the important underlying structure in the association of histograms and video frames, yet at the same time removes the noise or trivial variations in video frames. Minor differences between histograms will be ignored, and video frames with similar color distribution patterns will be mapped near to each other in the κ-dimensional refined feature space. From analogy with the SVD-based text clustering and retrieval [7], clustering visually similar frames in this refined feature space will certainly yields better results than in the raw feature space. The value of κ is a design parameter. Our experiments show that $\kappa = 150$ gives satisfactory video summarization results.

The above discussion has led us to the definition of the following distance metric between frame i and j for the frame clustering purpose:

$$\mathrm{D}(\psi_i, \psi_j) = \sqrt{\sum_{l=1}^{\kappa} \sigma_l (v_{il} - v_{jl})^2} \tag{4}$$

where ψ_i, ψ_j are the vectors representing frames i, j in the refined feature space, respectively, and σ_l's are the singular values from the SVD.

3.3 SVD-Based Video Summarization

Besides the above SVD properties, we have further discovered the following SVD feature which constitutes the basis of our video summarization system (See the appendix of this paper for proof).

Theorem 2 *Let the SVD of* $\mathbf{A}$ *be given by Eq.(1),* $\mathbf{A} = [A_1 \cdots A_i \cdots A_n]$, $\mathbf{V}^T = [\psi_1 \cdots \psi_i \cdots \psi_n]$. *Define the distance of* ψ_i *to the origin of the refined feature space as:*

$$||\psi_i|| = \sqrt{\sum_{j=1}^{rank(\mathbf{A})} v_{ij}^2} \, . \tag{5}$$

If rank($\mathbf{A}$*)=n, then, from the orthonormal property of matrix* $\mathbf{V}$*, we have* $||\psi_i||^2 = 1$*, where* $i = 1, 2, \ldots, n$*. Let* $\mathbf{A}' = [A_1 \cdots \overbrace{A_i^{(1)} \cdots A_i^{(k)}}^{k} \cdots A_n]$ *be the matrix obtained by duplicating column vector* A_i *in* $\mathbf{A}$ k *times (*$A_i^{(1)} = \cdots = A_i^{(k)} = A_i$*), and* $\mathbf{V}'^T = [\psi'_1 \cdots \overbrace{\phi'_1 \cdots \phi'_k}^{k} \cdots \psi'_n]$ *be the corresponding right singular vector matrix obtained from the SVD. Then,* $||\phi'_j||^2 = 1/k$*, where* $j = 1, 2, \ldots, k$*.*

The above theorem indicates that, if a column vector A_i of matrix $\mathbf{A}$ is linearly-independent, the SVD operation will project it into the vector ψ_i whose distance defined by Eq.(5) is one in the refined feature space. When A_i has some duplicates $A_i^{(j)}$, distance of its projected vector ϕ'_j decreases. The more duplicates A_i has, the shorter distance ϕ'_j holds. Translating this property into the video domain, it can be inferred that, in the refined feature space, frames in a static video segment (e.g., shots of anchor persons, weather maps) will be projected into the points closer to the origin, while frames in a video segment containing a lot of changes (e.g., shots containing moving objects, camera pan and zoom) will be projected into the points farther from the origin. In other words, by looking at the location at which a video is projected, we can roughly tell the degree of visual changes of the video.

From the viewpoint of content value, a static video with little visual change contains less visual content than a dynamic video with lots of changes. Since the degree of visual changes of a video segment has a strong correlation with the location of its corresponding cluster $\mathbf{S}_i$ in the refined feature space, we define the following quantity as a measure of content value contained in cluster (video segment) $\mathbf{S}_i$:

$$\mathrm{CON}(\mathbf{S}_i) = \sum_{\psi_i \in \mathbf{S}_i} ||\psi_i||^2 \tag{6}$$

With the above content measurement, the details of the summarization process are given as follows.

3.4 Operational Detail

Our video summarization system consists of the following major processing steps:

Main Process:

Step 1. Select frames with a fixed interval (=10 in our implementation) from the input video sequence, and create the feature-frame matrix $\mathbf{A}$ using these selected frames.

Step 2. Perform the SVD on $\mathbf{A}$ to obtain matrix $\mathbf{V}^T$ whose each column vector ψ_i represents frame i in the refined feature space.

Step 3. In the refined feature space, find the most static cluster, compute the content value of this cluster using Eq.(6), and use this value as the threshold to cluster the rest of the frames.

Step 4. For each obtained cluster $\mathbf{S}_i$, find the longest video shot Θ_i contained in the cluster. Discard the cluster whose Θ_i is shorter than one second.

Step 5. According to the user's request, output either a set of keyframes which each represents a frame cluster, or a summarized video with the user specified time length.

In Step 3 of the above operation, finding the most static cluster is equivalent to finding the cluster closest to the origin of the refined feature space. Referring to the notations in Theorem 1 and Theorem 2, the entire clustering process can be described as follows:

Clustering:

1. In the refined feature space, sort all the vectors ψ_i in ascending order using the distance defined by Eq.(5). Initialize all the vectors as unclustered vectors, and set cluster counter $C = 1$.

2. Among the unclustered vectors, select the one that is closest to the origin as the seed to form cluster $\mathbf{S}_C$. Set the average internal distance of the cluster $\overline{\mathrm{R}}(\mathbf{S}_C) = 0$, and the frame count $P_C = 1$.

3. For each unclustered vector ψ_i, calculate its minimum distance to cluster $\mathbf{S}_C$, which is defined as:

$$d_{min}(\psi_i, \mathbf{S}_C) = \min_{\psi_k \in \mathbf{S}_C} \mathrm{D}(\psi_i, \psi_k) \tag{7}$$

where $\mathrm{D}(\psi_i, \psi_k)$ is defined by Eq.(4). If cluster counter $C = 1$, go to Case (a); otherwise, go to Case (b).

(a) add frame ψ_i to cluster $\mathbf{S}_1$ if

$$\overline{\mathrm{R}}(\mathbf{S}_1) = 0 \qquad \text{or}$$
$$d_{min}(\psi_i, \mathbf{S}_1)/\overline{\mathrm{R}}(\mathbf{S}_1) < 5.0$$

(b) add frame ψ_i to cluster $\mathbf{S}_C$ if

$$\overline{\mathrm{R}}(\mathbf{S}_C) = 0 \qquad \text{or}$$
$$\mathrm{CON}(\mathbf{S}_C) < \mathrm{CON}(\mathbf{S}_1) \qquad \text{or}$$
$$d_{min}(\psi_i, \mathbf{S}_C)/\overline{\mathrm{R}}(\mathbf{S}_C) < 2.0$$

If frame ψ_i is added to cluster $\mathbf{S}_C$, increment frame count P_C by one, update the content value $\mathrm{CON}(\mathbf{S}_C)$ using Eq.(6), and update $\overline{\mathrm{R}}(\mathbf{S}_C)$ as follows:

$$\overline{\mathrm{R}}(\mathbf{S}_C) = \frac{(P_C - 1)\overline{\mathrm{R}}(\mathbf{S}_C) + d_{min}(\psi_i, \mathbf{S}_C)}{P_C} \qquad (8)$$

4. If there exit unclustered points, increment the cluster counter C by one, go to Step 2; otherwise, terminate the operation.

In the above operations, it should be noticed that different conditions are used for growing the first and the rest of clusters. The first cluster relies on the distance variation $d_{min}(\psi_i, \mathbf{S}_1)/\overline{\mathrm{R}}(\mathbf{S}_1)$ as its growing condition, while the remaining clusters examine the content value as well as the distance variation in the growing process. Condition 2 in Case (b) ensures that the cluster under processing contains the same amount of visual content as the first cluster, while Condition 3 prevents two frames which are very close to each other from being separated. With Condition 2, a long video shot with large visual variations may be clustered into more than one cluster, and consequently, will be assigned more than one keyframes. On the other hand, with the combination of Condition 2 and 3, video shots with very similar visual contents will be clustered together, and only one keyframe will be assigned to this group of video shots. This characteristic exactly meets our goals set at the beginning of this section.

In **Main Process**, Step 5 forms another unique characteristic of our proposed system: it is able to output either the optimal set of keyframes, or a summarized version of the original video with the user specified time length. When the keyframe output is selected by the user, the system performs the SVD and clustering operations described above. From each obtained cluster, the system selects one frame whose feature vector is closest to the center of the cluster as keyframe. The output of a summarized video requires more operations. Our system composes a summarized video according to the two user inputs: the time length of the summarized video T_{len}, and the minimum time length each shot should be displayed in the summarized video T_{min}. The process consists of the following main operations:

Summary Composition:

1. Let C be the number of clusters obtained from the above **Clustering** process, and $N = T_{len}/T_{min}$. For each cluster $\mathbf{S}_i$, find the longest video shot Θ_i.

2. If $C \leq N$, go to Case (i); otherwise, go to Case (ii).

   (i) Select all the shots Θ_i where $i = 1, 2, \ldots, C$, and assign an equal time length T_{len}/C to each of the shots.

   (ii) Sort shots Θ_i in descending order by length, select the top N shots, and assign an equal time length T_{min} to each selected shot.

3. Sort the selected shots by the time code, based on this sorted order, get from each selected shot a portion of the assigned time length, and insert it into the the summarized video.

Given the user's input T_{len} and T_{min}, the maximum number of video shots the summarized video can include equals $N = T_{len}/T_{min}$. If the total number of shots $C \leq N$, then all the shots will be assigned a slot in the summarized video (Case (i)); otherwise, the shots will be selected in descending order of the length to fill the summarized video. Here, the parameter T_{min} can be considered as a control knob for the user to select between depth-centric and breadth-centric summarization. A small value for T_{min} will produce a breadth-centric video summary that consists of more shots that each is shorter in length, while a large value for T_{min} will produce a depth-centric video summary that consists of less shots that each is longer in length. Moreover, because the clustering process is performed such that all the resultant clusters contain approximately the same amount of visual content, it is natural to assign the same time length to each selected shot to form the summarized video.

4 Experimental Results and Summary

We have implemented our video summarization system using C++, and tested the system using different types of video sequences. The test video sequences consist of news reports, documentary, political debate, and live coverage of a breaking event, each of which last for from 5 to 30 minutes.

Figure 1 shows the summarization result on a 5-minute news report covering the Clinton-Lewinsky

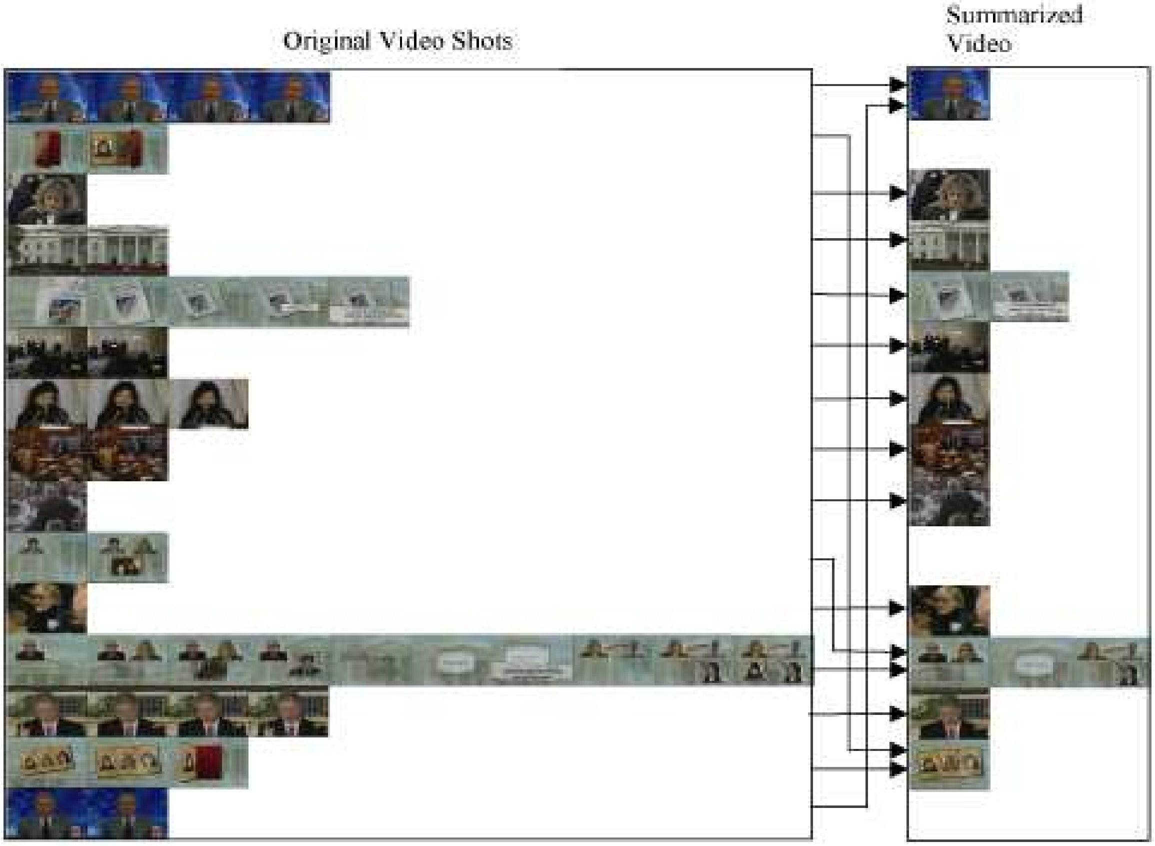

Figure 1: A video summarization result.

scandal. The sequence consists of 29 shots, and Figure 1 displays the 15 major shots. Each row in the left hand rectangle represents a shot in the original video, and the number of the frames in each row is proportional to the time length of the corresponding shot. The same row in the right hand rectangle depicts the keyframes assigned to the corresponding shot. In our experiment, the 13'th shot (represented by row 13) was detected as the most static shot, and was used as the content unit to cluster the rest shots. The anchor person appeared two times, one at the beginning (row 1), the other at the end (row 15) of the whole sequence. However, as the two shots are quite similar in terms of visual content, and contain little motion, they were clustered together, and were assigned only one keyframe (row 1 in the right hand rectangle). The similar situation occurs for shot 2 and 14 as well. Shot 12 is the longest shot, and contains lots of changes in the whole sequence. It was clustered into 3 clusters together with shot 10, and were assigned three keyframes. Similarly, as shot 5 contains many visual changes, it was also assigned two keyframes. Besides the set of keyframe output, a motion video summary with the user specified time length can be also generated by our system. Our experiment shows that a 30-second motion video summary contains most of the major shots from the original video sequence.

Table 1 shows the detailed evaluation results on five different video sequences. The political debate video contains many shots that display either the same speakers, or the same global view of the studio. As shown in the table, most of the visually similar shots have been properly merged by our system. The live coverage video consists of many long and important shots that gradually unveil the ongoing event in the field. Most of these shots have been assigned more than one keyframe (or slot) in the generated summary. For the documentary video, the number of incorrect shot merger is higher than in other videos. This is mainly caused by some shots in the sequence which have very similar color distributions but different visual contents.

References

[1] Y. Tonomura, A. Akutsu, K. Otsuji, and T. sadakata, "Videomap and videospaceicon: Tools for anatomizing video content," in *Proc. ACM INTERCHI'93*, 1993.

[2] H. Ueda, T. Miyatake, and S. Yoshizawa, "Impact: An interactive natural-motion-picture dedicated multimedia authoring system," in *Proc. ACM SIGCHI'91*, (New Orleans), Apr. 1991.

Table 1: Experimental Evaluation

Video Contents	**Time Length**	**Total Shots**	**Shots Merged**	**Properly Merged**	**Shots Separated**	**Properly Separated**
Political debate	8 min.	37	23	21	2	2
Documentary	10 min.	66	34	28	5	4
News report	5 min.	29	4	4	1	1
News report	12 min.	60	14	11	8	6
Live event coverage	30 min.	32	6	6	13	10

[3] A. Fermain and A. Tekalp, "Multiscale content extraction and representation for video indexing," in *Prc. SPIE 3229 on Multimedia Storage and Archiving Systems II*, 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, 1998.

[5] M. Yeung, B. Yeo, W. Wolf, and B. Liu, "Video browsing using clustering and scene transitions on compressed sequences," in *Proc, SPIE on Multimedia Computing and Networking*, vol. 2417, 1995.

[6] A. Girgensohn and J. Boreczky, "Time-constrained keyframe selection technique," in *Proc. IEEE Multimedia Computing and Systems (ICMCS'99)*, 1999.

[7] 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, pp. 391–407, 1990.

[8] W. Press and et al., *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge, England: Cambridge University Press, 2 ed., 1992.

[9] G. Golub and C. Loan, *Matrix Computations.* Baltimore: Johns-Hopkins, 2 ed., 1989.

Appendix: Proof of Theorem 2

Assume that except for k duplicates of A_i, the rest of the column vectors in $\mathbf{A}'$ are all linearly-independent. By performing the SVD on $\mathbf{A}'$, and, without loss of generality, by conducting some permutations, we have matrix ${\mathbf{V}'}^T$ as shown in Figure 2. In the figure, the row vectors from 1 to n are the right singular vectors whose corresponding singular values are non-zero, and each column vector $\phi'_j = [y_{j1}\ y_{j2}\ \cdots\ y_{j(n+k-1)}]$, $j = 1, \ldots, k$, in the hatched rectangle area corresponds to $A_i^{(j)}$ in $\mathbf{A}'$. Because the SVD projects the identical column vectors in $\mathbf{A}'$ to the same point in the refined feature space, the following condition holds:

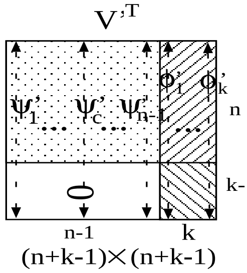

Figure 2: The structure of matrix ${\mathbf{V}'}^T$ with some permutations.

$$y_{1s} = y_{2s} = \cdots = y_{ks} \text{ where } 1 \leq s \leq n \tag{9}$$

Because ${\mathbf{V}'}^T$ is an orthonormal matrix,

$$\phi'_a \cdot \phi'_b = \delta_{ab} \tag{10}$$

$$\psi'_c \cdot \phi'_d = 0 \tag{11}$$

where $\phi'_a, \phi'_b, \phi'_d$ represent any column vectors from the hatched rectangle area, and ψ'_c represents any column vector in the dotted rectangle area. From Eq.(9) and Eq.(10), the condition in Eq.(9) does not hold for $n < s \leq n + k - 1$. From Eq.(9) and Eq.(11), Elements of $n+1$ to $n+k-1$ in each vector ψ'_c all equal zero. From the orthonormal property of ${\mathbf{V}'}^T$

$$\sum_{i=1}^{k} \sum_{s=n+1}^{n+k-1} y_{is}^2 = k - 1 \tag{12}$$

$$\sum_{i=1}^{k} \sum_{s=1}^{n+k-1} y_{is}^2 = k \tag{13}$$

subtracting Eq.(12) from Eq.(13), we have

$$\sum_{i=1}^{k} \sum_{s=1}^{n} y_{is}^2 = 1 \tag{14}$$

From Eq.(9) and Eq.(14), we have

$$||\phi'_j||^2 = 1/k, \text{ where } 1 \leq j \leq k \tag{15}$$