Multimedia Systems 9: 157–168 (2003)
Digital Object Identifier (DOI) 10.1007/s00530-003-0086-3

Multimedia Systems

Video summarization and retrieval using singular value decomposition

Yihong Gong, Xin Liu

NEC Laboratories of America, 10080 North Wolfe Road, SW3-350, Cupertino, CA 95014, USA

Abstract. In this paper, we propose novel video summarization and retrieval systems based on unique properties from singular value decomposition (SVD). Through mathematical analysis, we derive the SVD properties that capture both the temporal and spatial characteristics of the input video in the singular vector space. Using these SVD properties, we are able to summarize a video by outputting a motion video summary with the user-specified length. The motion video summary aims to eliminate visual redundancies while assigning equal show time to equal amounts of visual content for the original video program. On the other hand, the same SVD properties can also be used to categorize and retrieve video shots based on their temporal and spatial characteristics. As an extended application of the derived SVD properties, we propose a system that is able to retrieve video shots according to their degrees of visual changes, color distribution uniformities, and visual similarities.

Key words: Video summarization – Video retrieval – Singular value decomposition – Color histograms

1 Introduction

The widespread distribution of video images in computer systems and networks has presented both excitements and challenges. Video is exciting because it conveys real-world scenes most vividly and faithfully. Handling video is challenging because video images are voluminous, redundant, and their overall contents can not be captured at a glance. With a large video data collection, it is always a painful task to find either the appropriate video sequence, or the desired portions of the video. 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 bandwidths for most home users, people spend minutes or tens of minutes downloading voluminous video images, only to find them irrelevant. To turn video collections into valuable information resources, techniques that enable effective content-based search, and facilitate quick screening on the search result, become indispensable.

With a large amount of video data, presenting the user with a summary of each video program greatly facilitates the task of finding the desired video content. Video content search and summarization are the two essential technologies that complement each other. Video search engines return a set of video data meeting certain criteria, and video summarizers produce video content summaries that enable users to quickly grasp the overall content of each returned video. On the Internet, concise and informative video summaries are particularly important to accommodate limited communication bandwidths. Video search engines serve as an information filter that sifts out an initial set of relevant videos from the database, while video summarizers serve as an information spotter that helps the user to quickly examine through a given video set, and to spot the final set of desired video images.

To date, video summarization is mainly achieved by extracting a set of keyframes from the original video and displaying thumbnails of the keyframes in a storyboard window. The disadvantage of this approach is that keyframes are just a set of static images that contain no spatio-temporal properties of the original video. A video program is a continuous recording of real-world scenes. What distinguishes the video medium from the image medium is the video's capability of depicting the dynamics and the spatio-temporal evolution of the target scene. A set of static keyframes by no means captures these essential video properties, and is indeed a poor representation of general visual contents of a video program.

In this paper, we propose novel video summarization and retrieval systems based on unique properties from singular value decomposition (SVD). Through mathematical analysis, we derive the SVD properties that capture both the temporal and spatial characteristics of the input video in the singular vector space. Using these SVD properties, we are able to summarize a video by outputting a motion video summary with the user-specified length. The motion video summary aims to eliminate visual redundancies while assigning equal show time to an equal amount of visual content for the original video program. The automatically generated motion video summaries are not intended to replace the original videos, but

e-mail: {ygong,xliu}@ ccrl.sj.nec.com

to facilitate better visual content overviews by which the user is able to quickly figure out the general contents of the video collection, and to judge whether the contents are of interest or not. On the other hand, the same SVD properties can also be used to categorize and retrieve video shots based on their temporal and spatial characteristics. As an extended application of the derived SVD properties, we propose a system that is able to retrieve video shots according to their degrees of visual changes, color distribution uniformities, and visual similarities.

In the following, Section 2 describes related work in the literature. Section 3 outlines the proposed video summarization and retrieval systems. Sections 4 to 7 describe the major components of the proposed systems. Sections 8 and 9 present the performance evaluations of the video summarization and retrieval systems, respectively. Section 10 summarizes the paper.

2 Related work

To date, video summarization 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. [2] used the first frame from each shot as a keyframe. Ueda et al. [3] represented each shot using its first and last frames. Ferman and Tekalp [4] clustered the frames in each shot, and selected the frame closest to the center of the largest cluster as the keyframe.

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. [5] proposed to assign keyframes of a variable number according to the activity level of the corresponding scene shot. 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 duplicates and redundancies. In a TV talk show 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 sequence will yield many keyframes that are almost identical. To remove redundancies from keyframes, Yeung et al. [6] selected one keyframe from each shot, performed hierarchical clustering on these keyframes based on their visual similarity, and temporal distance, and then retained only one keyframe for each cluster. Girgensohn and Boreczky [7] 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. In recent clustering-based video summarization Work, Dirk Farin et al. [8] proposed to incorporate domain-knowledge to eliminate the selection of keyframes from irrelevant or uninteresting shots. Examples of such shots include transitional shots from fades and wipes, shots of commercial advertisements, weather forecast charts, etc. The domain-knowledge is incorporated into the clustering process by introducing the feature vectors of uninteresting scenes as additional cluster centers, and frames grabbed by the clusters of these additional cluster centers are excluded from the keyframe selection process.

Apart from the above methods for keyframe selection, summarizing video content using keyframes has its own limitations. A video program is a continuous spatio-temporal recording of real-world events. What distinguishes the video medium from the image medium is the video's capability of depicting the dynamics and the spatio-temporal evolution of the target scene. A set of static keyframes by no means captures these essential video properties, and is indeed a poor representation of general visual contents of a video program. To reduce the spatio-temporal content loss caused by the keyframe presentation, image mosaicing techniques have been utilized to create a panoramic view of the entire scene recorded by each shot [9–11]. For a given video shot, its mosaic is created by warping all the frames into a common coordinate system, and then stitching them together to reconstruct the full background seen by each of the frames in the whole shot. However, a mosaic can be successfully created only when the target scene satisfies several severe conditions, such as that the scene must have no prominent 3D structures, should contain no moving objects, etc. These strict prerequisites have certainly excluded the majority of videos from the application range of mosaic-based video summarization methods.

There have been research efforts that strive to output motion video summaries to accommodate better content overviews. The CueVideo system from IBM provides a fast video playback function which plays long, static shots with a faster speed (higher frame rate), and plays short, dynamic shots with a slower speed (lower frame rate) [12]. However, this variable frame rate playback causes static shots to look more dynamic, and dynamic shots to look more static, therefore it dramatically distorts the temporal characteristics of the video sequence. On the other hand, the Informedia system from CMU provides video skim, that strives to identify and playback only semantically important image segments along with semantically important audio keywords/phrases in the video sequence [13]. The importance of each image segment is measured using a set of heuristic rules that is highly subjective and content-specific. This rule-based summarization system has certainly placed limitations on the handling of diversified video images.

In recent years, content summarization of TV broadcast sports games has received increased attention from researchers in the multimedia community. As sports videos have well defined internal structures in which all the plays are rule-based and predictable, sports video summarization systems can achieve higher levels of content abstraction by exploring domain-specific features and knowledge. Gong et al. [14] developed a soccer game parsing system that classifies each

shot of a soccer video according to its physical location in the field, or the presence/absence of the soccer ball. The soccer field line patterns, players' positions and movements, and the ball's presence/absence were used to recognize the category of each scene shot. Xu et al. [15] suggested that any soccer game is composed of the *play* and *break* states, and developed a system that partitions soccer videos into play/break segments by looking at grass area (green color) ratio in video frames. Zhong and Chang [16] focused on the fact that all the highlights in baseball and tennis games start from pitchings and serves, respectively, and they strove to detect baseball pitching views and tennis serving views by classifying color histograms of keyframes of the scene shots. Rui et al. [17] assumed that exciting segments in baseball games are highly correlated with an announcers' excited speech, and mostly occur right after a baseball batting. Based on these assumptions, they detected baseball highlights based on the analysis of the announcers' speech pitch, and the detection of the baseball batting sound.

3 System overview

In this paper, we propose a novel technique for video summarization and retrieval based on the same framework: color histograms together with singular value decomposition. For video summarization, we strive to create a motion video summary of the original video that: (1) has a user-specified summary length and granularity; (2) contains little visual redundancy; and (3) gives equal attention to equal amounts of visual content. The first goal aims to meet different content overview requirements from a variety of users. The second and third goals are intended to turn the difficult, subjective visual content summarization problem into a feasible and objective one. By common sense, an ideal video summary should be the one that retains only semantically important segments of the given video program. However, finding semantically important video segments requires an overall understanding of the video content, which is beyond our reach given the state of the art of current video analysis and image understanding techniques. The task is also very subjective, and it is hard to find commonly agreed upon criteria for measuring semantic importance. On the other hand, it is relatively easy to measure the activity level of visual content, and to identify duplicates and redundancies in a video sequence. For the purpose of visual content browsing and overview, the video watching time will be largely shortened, and the visual content of the original video will not be dramatically lost if we eliminate those duplicates/redundancies and preserve those visually active contents. Therefore, instead of summarizing videos by heuristically selecting "important" video segments, we choose to create video summaries that meet the above summarization goals using the derived SVD properties. The automatically generated video summaries are not intended to replace the original videos, but to facilitate better visual content overviews, with which the user is able to quickly figure out the general contents of the given videos, and to judge whether they are of interest .

As for video content retrieval, in addition to the basic requirement of retrieving the shots from the database that are visually similar to the sample shot, we also strive to realize the retrieval of shots according to their degrees of visual changes and color distribution uniformities. As demonstrated in Sect. 9, these new video retrieval capabilities are valuable for spotting video shots that either contain visually active contents, or consist of frames with skewed color distributions such as black frames, white frames, frames with very few colors, etc. These shots could either indicate semantically important contents, such as quickly evolving events, important messages displayed on uniformly colored background, etc., or represent visually trivial contents that arise from flashlights, transitional periods such as fades, wipes, etc.

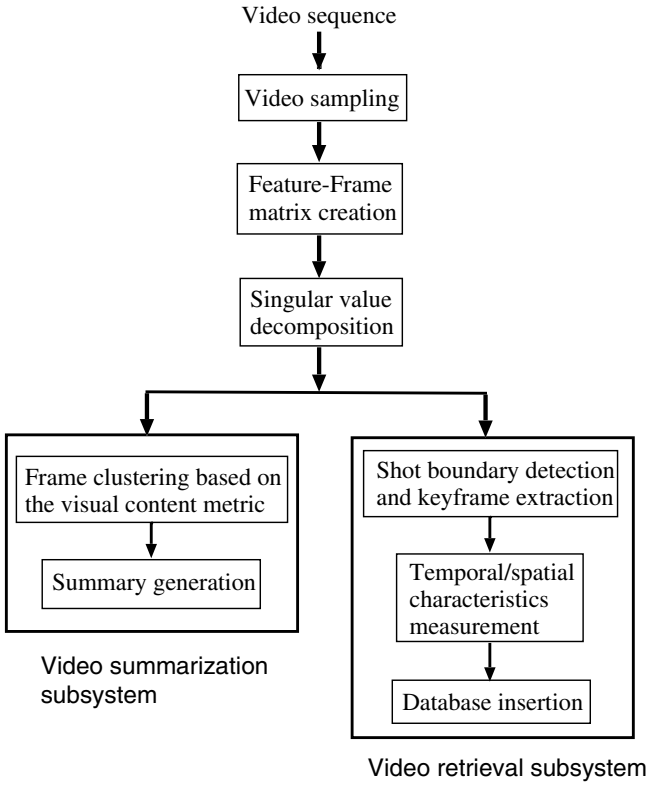

Fig. 1. Block diagram of the video summarization and retrieval systems

Figure 1 shows a block diagram of the proposed video summarization and retrieval systems. To reduce the number of frames to be processed by the SVD, we roughly sample the input video sequence with a fixed rate of five frames/second. Our experiments have shown that this sampling rate is sufficient for video programs without many dramatic motions, such as news, documentaries, talk shows, etc. For each frame i in the sampling set, we create an m-dimensional feature vector A_i. Using A_i as column vector i, we obtain the feature-frame matrix $\mathbf{A} = [A_1 \ A_2 \ \cdots \ A_n]$. Performing SVD on matrix $\mathbf{A}$ will project each frame i from the m-dimensional raw feature space into a τ-dimensional singular vector space (usually $\tau \ll m$). Through mathematical analysis, we derive unique SVD properties that capture both the spatial and temporal characteristics of the input video in the singular vector space. By using these SVD properties, we are able to achieve the goals we set for both the video summarization and retrieval at the beginning of this section.

As shown in Fig. 1, both the video summarization and retrieval systems share the operations of video sampling, feature-frame matrix creation, and singular value decomposition. The

operations become different for the two systems after the video frames are projected into the singular vector space. However, both systems make use of the same set of singular vectors, and no further feature extraction processes are involved in their subsequent operations. The details of the major operations in Fig. 1 are described in subsequent sections.

4 Feature-frame matrix creation

The video summarization and retrieval systems start from feature-frame matrix creation. From a wide variety of image features, we selected color histograms to represent each video frame. As demonstrated in Sect. 7, the combination of color histograms and SVD captures the information of color distribution uniformity for each frame. Further, histograms are very good for detecting overall differences in images [18], and are cost-effective for computing. Using cost-effective histograms here ensures feasibility and processing speed of the system in handling long video sequences. In our system implementation, we create three-dimensional histograms in the RGB color space with five bins for R, G, and B, respectively, resulting in a total of 125 bins. To incorporate spatial 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 the ith 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 matrices can be applied here, which are much faster and memory-efficient as compared to regular SVD algorithms.

5 Singular value decomposition (SVD)

Given an $m \times n$ matrix $\mathbf{A}$, where $m \geq n$, the SVD of $\mathbf{A}$ is defined as [19]:

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

where $\mathbf{U} = [u_{ij}]$ is an $m \times n$ column-orthonormal matrix whose columns are called left singular vectors; $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ 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 and retrieval systems, 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 singular vector space with all of its axes linearly-independent. This mapping projects each column vector $A_i = [a_{1i}\ a_{2i}\ \cdots\ a_{mi}]^T$ of matrix $\mathbf{A}$, which represents the concatenated histograms of frame i, to column vector $\psi_i = [v_{i1}\ v_{i2}\ \cdots\ v_{ir}]^T$ of matrix $\mathbf{V}^T$, and projects each row vector j of 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 $\mathbf{U}$.

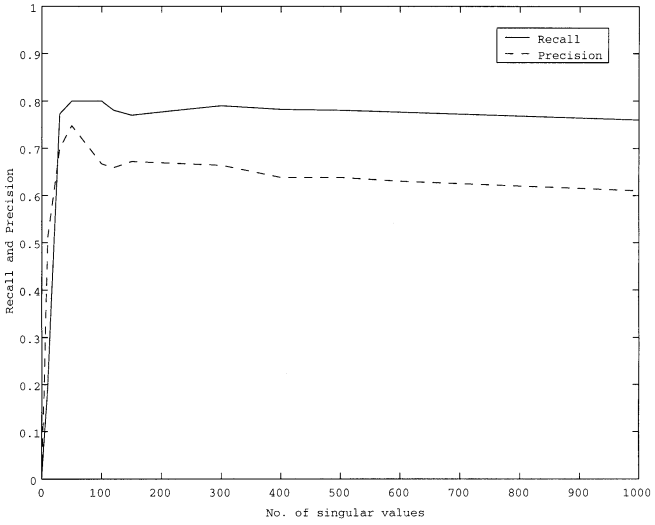

Fig. 2. Performance evaluation of the SVD-based shot boundary detection using τ as a parameter

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 dimension reduction property that has been widely utilized by many applications in many areas (see [20] 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* $\mathbf{A}_\tau$ *(*$\tau \leq r$*) defined below is the closest rank-*τ *matrix to* $\mathbf{A}$ *for the Euclidean and Frobenius norms.*

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

The use of τ-largest singular values to approximate the original matrix with Eq. (3) has more implications than just dimension reduction. Discarding small singular values is equivalent to discarding linearly semi-dependent or practically non-essential axes of the singular vector space. In our case, axes with small singular values usually capture either non-essential color variations or noise within the video sequence. The truncated SVD, in one sense, captures the most salient 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 τ-dimensional partial singular vector space. Similarity comparison between frames in this partial singular vector space will certainly yield better results than in the raw feature space.

To demonstrate the impact of discarding small singular values on the similarity comparison between video frames, we

implemented a shot boundary detection method in the singular vector space, and evaluated the performance of the method using τ as a parameter. More precisely, for each frame i, we take the column vector ψ_i of matrix $\mathbf{V}^T$ as its feature vector, and use Eq. (4) as the similarity metric to compare it with frame $j = i + 1$. Once the difference between frame i and j exceeds the predefined threshold, we declare the detection of a shot boundary:

$$\mathrm{D}(\psi_i, \psi_j) = \sqrt{\sum_{k=1}^{\tau} \sigma_k (v_{ik} - v_{jk})^2} \tag{4}$$

In fact, Eq. (4) defines a Euclidean distance weighted by the singular values σ_k. τ is the parameter that specifies how many singular values are to be used in the metric.

The shot boundary detection method is evaluated using two common measures, recall and precision, which are defined as follows:

$$Recall = \frac{\text{No. of correctly detected boundaries}}{\text{No. of true boundaries}}$$

$$Precision = \frac{\text{No. of correctly detected boundaries}}{\text{No. of totally detected boundaries}}$$

The evaluation was performed using a set of TV video programs with a total length of three hours. These video programs consist of a great variety of scene categories taken from news, documentaries, movies, TV commercials, etc. All the video programs are in MPEG1 format with a frame size of 352×240 pixels. Figure 2 shows the evaluation result with the value of τ as a parameter. For each given τ value, we empirically determined the shot boundary detection threshold so that the best recall and precision were obtained. It can be seen from the figure that when τ takes values below 30, the shot boundary detection yields poor recall and precision (below 0.5). When τ equals 50, both the recall and precision reach their maximum. When τ further increases, the recall decreases slightly and then flattens out, whereas the precision decreases by 10% and then stabilizes at that level. Since the raw feature vector of each video frame has 1125 dimensions, theoretically the SVD operation can produce up to 1125 singular values. However, because these 1125 dimensions are not fully independent of each other, the SVD seldom produces 1125 non-zero singular values. Our experiments have shown that the singular values from rank 600 and onward are either very small or could be ignored.

Based on the above experiments, we set the value of τ to 50, and use this value as well as Eq. (4) as the similarity metric for both the video summarization and the retrieval systems. With $\tau = 50$, the best threshold for the shot boundary detection equals 8000.

For comparison, we also implemented the same shot boundary detection method using the raw feature vectors. That is, instead of using the column vector ψ_i of matrix $\mathbf{V}^T$, we use the column vector A_i of matrix $\mathbf{A}$ for each frame i. The Euclidean distance between two raw feature vectors is used as the similarity metric. The performance evaluation using the same test video set yielded a 73% recall and 67% precision. This performance is compatible to that of the singular vector-based method using the full set of singular values. This comparison is further evidence for the impact of discarding small singular values on the similarity comparison between video frames.

6 SVD-based video summarization

Besides the SVD properties described in the above section, we have derived the following SVD feature, which constitutes the basis of our video summarization system (see Appendix 1 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 norm of* $\psi_i = [v_{i1}\ v_{i2}\ \cdots\ v_{in}]^T$ *in the singular vector 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 norm defined by Eq. (5) is one in the singular vector space. When A_i has some duplicates $A_i^{(j)}$, the norm of its projected vector ϕ'_j decreases. The more duplicates A_i has, the smaller norm ϕ'_j holds. Translating this property into the video domain, it can be inferred that, in the singular vector 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 those points furthest from the origin. In other words, by looking at the location at which a video segment is projected, we can roughly tell the degree of visual changes of the video segment.

Consider a video and ignore its audio signals. From the viewpoint of visual content, a static video with little visual change contains less visual content than a dynamic video with lots of changes. In other words, the degree of visual changes in a video segment is a good indicator of the amount of visual content conveyed by the segment. Since the degree of visual changes of a given video segment has a strong correlation with the location of its corresponding cluster $\mathbf{S}_i$ in the singular vector space, we define the following quantity as a metric of visual content contained in cluster (video segment) $\mathbf{S}_i$:

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

Using the above visual content metric, we strive to group video frames into units with approximately equal amounts of visual content in the singular vector space. More precisely, in the singular vector space, we first find the most static frame cluster, define it as the visual content unit, and then use the value of the visual content metric computed from it as the threshold to cluster the rest of the frames. The main operations of the video summarization system are given as follows (Fig. 1).

Main process

Step 1. Roughly sample the input video sequence with a fixed rate.
Step 2. Create the feature-frame matrix $\mathbf{A}$ using the frames in the sampling set (see Sect. 4).
Step 3. Perform the SVD on $\mathbf{A}$ to obtain matrix $\mathbf{V}^T$ in which each column vector ψ_i represents frame i in the singular vector space.
Step 4. In the singular vector space, find the most static cluster, compute the value of its visual content metric using Eq. (6), and cluster the rest of the frames into units that have approximately the same amount of visual content as the most static cluster.
Step 5. For each cluster $\mathbf{S}_i$ obtained, find the longest shot Θ_i contained in the cluster. Discard the cluster whose Θ_i is shorter than one second. Output a summarized motion video with the user-specified time length.

In Step 4 of the above operation, finding the most static cluster is equivalent to finding the cluster closest to the origin of the singular vector space. Referring to the notation in Theorems 1 and 2, the entire clustering process in Step 4 can be described as follows.

Clustering

1. In the singular vector space, sort all the vectors ψ_i in ascending order of their norms 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 has the shortest norm 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

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

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

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

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:

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

4. If there exist 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 the 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 visual content measure 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 that are very close to each other from being separated. With Condition 2, a long video shot with large visual variations will be clustered into more than one cluster, and consequently, will be assigned more than one slot in the summary. On the other hand, with the combination of Conditions 2 and 3, video shots with very similar visual content will be clustered together, and only one slot will be assigned to this group of video shots. These characteristics exactly meet our goals set for the video summarization system.

In the main process, Step 5 forms another unique characteristic of our video summarization system: it is able to output a summarized motion video of the original video sequence with the user-specified time length and granularity. The 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 (the granularity) 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 $N = T_{\mathrm{len}}/T_{\min}$. 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 $L = T_{\mathrm{len}}/C$ 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 $L = T_{\min}$ to each selected shot.
3. From each selected shot, take the first L seconds of the shot, and concatenate these video segments in their original time order to form the motion video summary.

Given the user's input T_{len} and $T_{\min}$, the maximum number of shots the summarized video can include equals $N = T_{\mathrm{len}}/T_{\min}$. 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 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 (or summary with a small granularity), which consists of more shots that are shorter in length, while a large value for $T_{\min}$ will produce a depth-centric video summary (or summary with a large granularity), which consists of fewer shots

that are 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.

7 SVD-based video retrieval

The video retrieval system shares the operations of video sampling, feature-frame matrix creation, and singular value decomposition with the video summarization system. As camera shots are considered to be a natural and appropriate processing unit for indexing and retrieval, we segment the input video into individual camera shots using the shot boundary detection method described in Sect. 5. From each camera shot, we extract a keyframe and use it for matching and retrieval of visually similar shots. In the singular vector space, the following three metrics are defined to measure visual similarity between shots, degree of visual changes, and uniformity of color distributions of each shot.

Similarity metric

As demonstrated in Sect. 5, the similarity metric using the singular vectors (Eq. (4)) is superior to that using the raw feature vectors for shot boundary detection. By analogy, the superiority of Eq. (4) should also apply for matching and retrieval of visually similar keyframes. However, for Eq. (4) to accurately measure the similarity among keyframes from different video sequences, all the keyframes must be processed by the same SVD . To handle this problem, we propose the following scheme, consisting of the database creation and updating stages.

1. **Database creation**, which builds a video database from scratch with the following steps: (a) Collect all the video sequences to be indexed by the database. (b) Segment each sequence into individual shots, and record the boundary of each shot in the database. (c) Extract a keyframe from each shot, and create the raw feature vector (see Sect. 4) for each keyframe. (d) Use the raw feature vectors of all the keyframes together to create the feature-frame matrix $\mathbf{A}$, and perform the SVD on matrix $\mathbf{A}$. Store the matrices $\mathbf{U}$, $\mathbf{\Sigma}$, and $\mathbf{V}^T$ into the video database. (e) Use column vector ψ_i from $\mathbf{V}^T$ as the feature vector of keyframe i for matching and retrieval.
2. **Database updating**, which adds new keyframes to the existing video database using the following folding-in technique [21]. Let A_x be the raw feature vector of a new keyframe x. The projected feature vector ψ_x of frame x in the singular vector space is computed by

$$\psi_x = \mathbf{\Sigma}^{-1}\mathbf{U}^T A_x \tag{9}$$

If the similarity metric Eq. (4) uses only τ singular values, then in the above equation, only the top τ singular values from matrix $\mathbf{\Sigma}$ need to be used to compute ψ_x, which produces ψ_x as a τ-dimension vector.

In the above database updating stage, instead of recomputing the SVD, the folding-in technique is used to add new keyframes to the database. Folding-in is based on the existing singular vector space obtained at the database creation stage, and hence new keyframes have no effect on the representation of the pre-existing keyframes stored in the database. Folding-in requires much less time and memory for computing, but can have deteriorating effects on the representation of the new keyframes. Therefore, folding-in is a preferable choice when a small number of new keyframes need to be added into the database, whereas recomputing the SVD for the whole keyframe set should be performed when new keyframes to be added exceed a certain percentage (i.e. 20%) of the database population.

Color distribution uniformity

For each keyframe extracted from the corresponding camera shot, its color distribution uniformity can be measured using the following SVD property (see Appendix 2 for proof).

Theorem 3. *Assume that the SVD of* $\mathbf{A}$ *be given by Eq. (1),* $\mathbf{A} = [A_1 \cdots A_i \cdots A_n]$, $\mathbf{V}^T = [\psi_1 \cdots \psi_i \cdots \psi_n]$. *Let* $A_i = [a_{1i}\ a_{2i}\ \cdots\ a_{mi}]^T$, *and* $\psi_i = [v_{i1}\ v_{i2}\ \cdots\ v_{in}]^T$. *Define the singular value weighted norm of* ψ_i *as:*

$$||\psi_i||_\Sigma = \sqrt{\sum_{j=1}^{rank(\mathbf{A})} \sigma_j^2 v_{ij}^2} \tag{10}$$

Then, $||\psi_i||_\Sigma^2 = A_i \cdot A_i = \sum_{j=1}^m a_{ji}^2$

In the above theorem, because A_i is the concatenated histograms of frame i, the sum of its elements a_{ji} equals a constant value $\sum_{j=1}^m a_{ji} = P$ (the number of pixels in the frame). Hence, $||\psi_i||_\Sigma^2$ reaches the minimum when $a_{1i} = a_{2i} = \cdots = a_{mi} = \frac{P}{m}$, while it reaches the maximum when one element $a_{ki} = P$ and the remaining elements all equal zero. In other words, the norm $||\psi_i||_\Sigma^2$ is proportional to the uniformity of the color distribution of frame i. This norm becomes the shortest when frame i has a complete uniform color distribution, and it becomes the longest when frame i consists of only one color. $||\psi_i||_\Sigma^2$ is a measurement that can be computed independent of other video sequences. Therefore, in contrast to the similarity metric, there is no need to compute the SVD for all the keyframes stored in the database before we can compute $||\psi_i||_\Sigma^2$ for each frame i.

Degree of visual changes

Besides the similarity and color distribution measures, it is also desirable to measure the temporal characteristics of each shot. Theorem 2, which was used to derive the visual content metric in Sect. 6, can be used here to measure the degree of visual changes for each shot. Let ψ_i be the projected feature vector of frame i in the singular vector space. Since the norm $||\psi_i||^2$ is strongly related to the visual variation level of the shot to which frame i belongs, the quantity $\sum_{\psi_i \in \mathbf{S}_i} ||\psi_i||^2$ can be used as a metric to measure the degree of visual changes of shot $\mathbf{S}_i$. Similar to the color distribution metric, this quantity can be computed independent of other video sequences, and it is not necessary to compute a second SVD incorporating all the video sequence.

Table 1. Evaluation for video summarization

Time Length	Total Shots	Similar shots Properly merged	Dynamic shots Assigned more slots
120 min.	683	79%	86%

Table 2. Evaluation for video retrieval

	Recall	Precision
Similarity retrieval	76%	72%
Color Uniformity	91%	94%
Dynamic Degree	94%	95%

8 Evaluation of video summarization

Conducting an objective and meaningful evaluation for a video summarization method is particularly 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 an 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 summaries, making the creation of any agreed-upon performance metrics even more difficult.

With the above observations in mind, we choose to evaluate our video summarization system using the objectives set at the beginning of Sect. 3, which are (1) adjustable summary length and granularity, (2) redundancy reduction, and (3) equal attention to an equal amount of visual content. The realization of the first objective is obvious from the algorithm itself. The redundancy reduction can be measured by showing how many visually similar or duplicated shots have been deprived, and how many static shots with little visual change have been shortened. For the objective of equal attention to an equal amount of visual content, we show the percentage of long, dynamic shots that have been assigned more play time in the summaries produced. We used a portion of the video data set described in Sect. 5 that excludes TV commercials and many short video programs which are not very meaningful for summarization. The test data set has a total of two hours in length, and consists of news reports, documentaries, political debates, talk shows, and live coverage of breaking events.

Figure 3 details a one minute summary of a six-minute news report covering the Clinton–Lewinsky scandal. The sequence consists of 29 shots, and Fig. 3 displays the 15 major shots. Each row on the left hand rectangle represents a shot in the original video, and the number of frames in each row is proportional to the time length of the corresponding shot. The same row on the right hand rectangle depicts the number of slots assigned to the corresponding shot. Each slot has an equal play time that are calculated using the algorithm described in Sect. 6. In our experiment, the thirteenth shot (represented by row 13) was detected as the most static shot, and was used as the visual content unit to cluster the rest of the 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 static and visually similar, they were clustered together, and were assigned only one slot in the summary (row 1 on 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 three clusters together with shot 10, and assigned three slots. Similarly, as shot 5 contains many visual changes, it was also assigned two slots. Figure 3 demonstrates that this one minute motion video summary has largely met the objectives we set for video summaries.

Table 1 shows the overall evaluation result on the two hour test video set. The result shows that our video summarization method has an 86% accuracy for segmenting long, dynamic shots into multiple clusters, hence giving more attention to this type of shot. For merging visually similar or duplicated shots, our method shows a 79% accuracy. The failure of merging some visually similar shots occurs mainly when two shots have similar color distributions but different illumination conditions, or position shifts of the main objects (e.g. anchor persons, buildings).

9 Evaluation of video retrieval

The video retrieval system is evaluated using the same three hour video set used for testing the shot boundary detection method. Each video program in the test set has been segmented into individual shots, and this shot segmentation process has yielded 1100 camera shots, and hence produced 1100 keyframes.

The video retrieval system supports three basic types of video search: search based on the visual similarity, the color distribution uniformity, and the degree of visual changes. Any combination of the three basic search types are also supported. For similarity-based video search, the user must present a sample frame of the desired video shot. The system matches the sample frame with all the keyframes stored in the database, and retrieves 16 shots whose keyframes are among the top 16 matches. On the other hand, searching shots based on the color distribution uniformity and the degree of visual changes are mainly realized using the predefined thresholds. With the color distribution uniformity metric defined by Eq. (10), keyframes with a value above 19,000 are composed of very few colors, with their histograms containing many empty bins. In contrast, keyframes with a value below 4000 are composed of many colors, and relatively uniform color distributions are observed in their histograms. On the other hand, using the visual change metric defined by Eq. (5), shots with the value above 2000 can be categorized as the dynamic shots containing high degrees of visual changes, while shots with the value below 1000 can be categorized as static shots with very little change.

Figure 4 shows examples of video retrieval based on the visual similarity, the color distribution uniformity, and the degree of visual changes. Figures 4(a), (b), and (c) display the top 16 returns from the database using the respective retrieval methods. In Figure 4(a), a frame from a scene of the Clinton testimony before Kenn Starr is used as a sample frame (displayed in the top large window). and the system returns the top 16 matches, 12 of which belong to the other shots of the Clinton testimony. The rest of the four images are similar either with the overall color layout and distribution (thirteenth and fourteenth images), or with the main object (fifteenth and sixteenth images). In Figure 4(b), retrieval of shots with skew

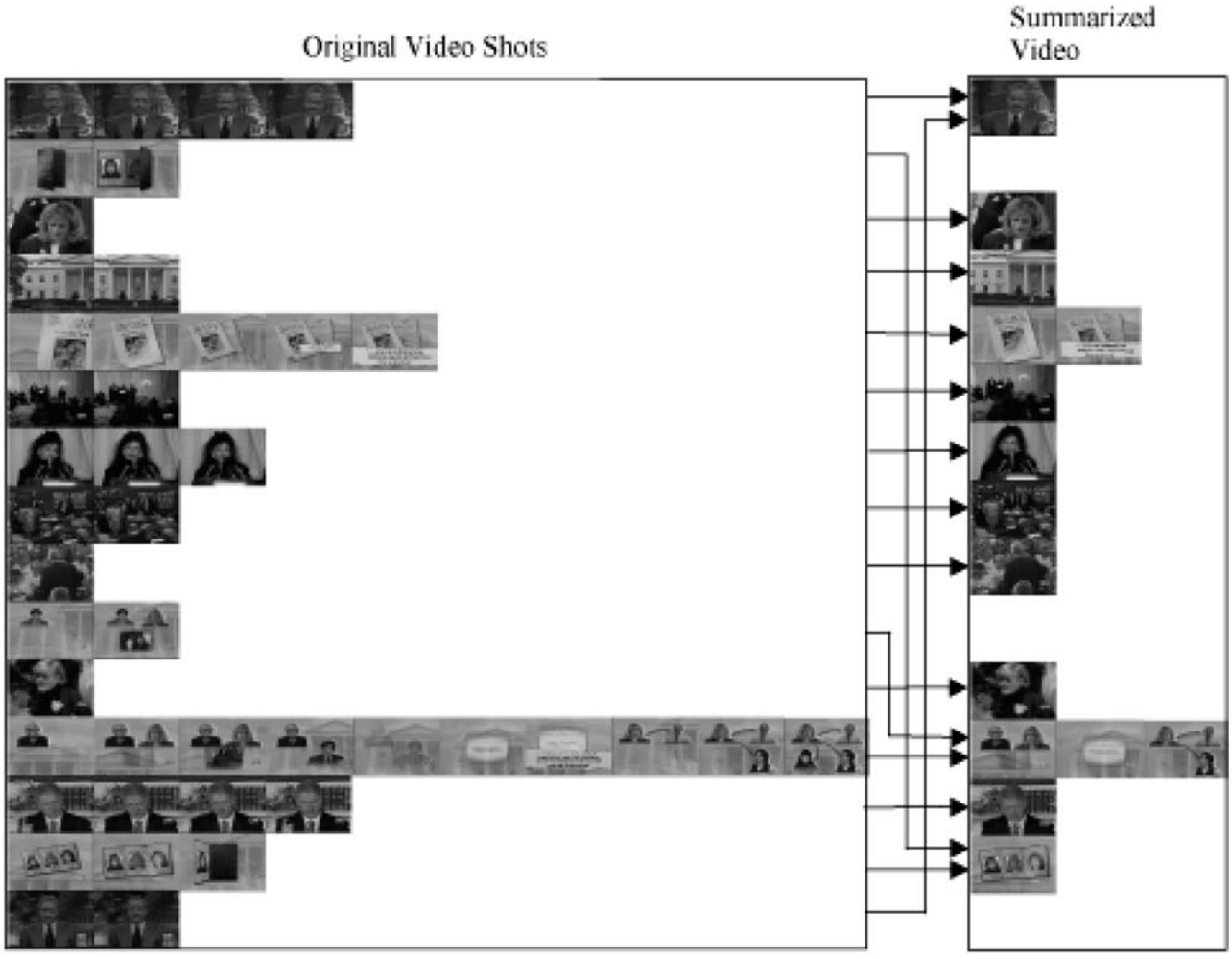

Fig. 3. A video summarization result

color distributions is conducted, and the 16 keyframes shown in the picture are either black-white images, or composed of very few colors. In Figure 4(c), the results of retrieving dynamic shots are displayed, and they represent shots that contain dissolves, fast moving objects, camera panning, etc. For the second and third types of video retrieval, if no other constraints are imposed, the system will return all the keyframes that satisfy the predefined threshold. If more than 16 keyframes are returned from the database, the arrow buttons at the bottom of the window can be used to move to the previous, or the next, page of the returned keyframes.

The video retrieval system is evaluated using the recall and precision whose definitions are analogous with those defined in Sect. 5. Table 2 shows experimental results using the three hour test video set. It is clear from the table that, while similarity-based video retrieval has obtained reasonable performance, video retrieval based on the color uniformity and degree of visual changes have achieved impressive recall and precision. This result has demonstrated the effectiveness of those SVD derived metrics described in Sect. 7.

10 Summary and discussion

In this paper, we have proposed a novel framework that realizes the video summarization and retrieval by sharing the same features and the same singular vector space. Through mathematical analyses, we have derived the SVD properties that capture both the temporal and spatial characteristics of the input video sequence in the singular vector space. Using these SVD properties, we are able to categorize and retrieve video shots according to their degrees of visual changes, color distribution uniformities, and visual similarities. On the other hand, these SVD properties also enable us to derive a metric to measure the visual content value of a given video segment. Using this metric, we are able to group video frames into clusters with approximately the same visual content value. With this clustering technique, we strive to generate video summaries that (1) have an adjustable length and granularity controllable by users, (2) contain little redundancy, and (3) give equal attention to the same amount of visual content.

In the experimental evaluations, we have used the above objectives to measure the effectiveness of the video summarization system, and used the common recall and precision metrics to evaluate the performance of the video retrieval system. The evaluation results have shown that the video summarization system has certainly met our objectives, and the

a

Fig. 4a–c. Video retrieval examples

video retrieval system has revealed the effectiveness of retrieving video shots based on the visual similarity, the color distribution uniformity, and the degree of visual changes.

The SVD and certain SVD properties (e.g. Theorems 1 and 2) can be applied to other image features as long as the features have a fixed number of dimensions. The advantages of using SVD will become more obvious with high dimension features (e.g. more than 100 dimensions), because SVD can tell which dimensions are important and which are not. As showcased in Sect. 5 and in [21], in singular vector spaces, discarding unimportant dimensions is more than just the dimension detection; in many cases, it improves system performance.

Our future work includes the improvement of the summary composition process so that the generated motion video summary will have a better audio effect, the testing of the singular value decomposition on different color spaces to compare them with the RGB color space. We are also going to apply the SVD to other image features such as edges, textures, etc., and conduct performance comparisons with the histogram+SVD combination.

Appendix 1: 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 Fig. 5. In the figure, the row vectors from 1 to n are the right sin-

b 

c 

Fig. 4a–c. (continued)

gular 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:

$$y_{1s} = y_{2s} = \cdots = y_{ks} \quad \text{where } 1 \le s \le n \tag{11}$$

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

$$\phi'_a \cdot \phi'_b = \delta_{ab} \tag{12}$$

$$\psi'_c \cdot \phi'_d = 0 \tag{13}$$

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. (11) and Eq. (12), the condition in Eq. (11) does not hold for $n < s \le n+k-1$. From Eq. (11) and Eq. (13), elements of $n+1$ to $n+k-1$ in

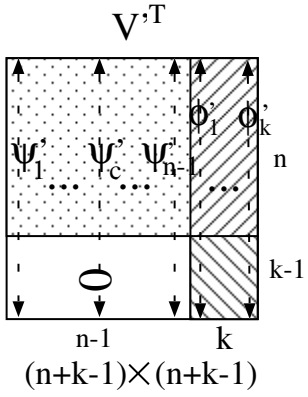

Fig. 5. The structure of matrix $\mathbf{V'}^T$ with some permutations

each vector ψ'_c all equal zero. From the orthonormal property of $\mathbf{V'}^T$

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

$$\sum_{i=1}^{k}\sum_{s=1}^{n+k-1} y_{is}^2 = k \tag{15}$$

subtracting Eq. (14) from Eq. (15), we have

$$\sum_{i=1}^{k}\sum_{s=1}^{n} y_{is}^2 = 1 \tag{16}$$

From Eq. (11) and Eq. (16), we have

$$||\phi'_j||^2 = 1/k, \quad \text{where } 1 \le j \le k \tag{17}$$

Appendix 2: Proof of Theorem 3

Assume that the SVD of $\mathbf{A}$ is as given by Eq. (1). Let $\mathbf{A} = [A_1 \cdots A_i \cdots A_n]$, $\mathbf{V}^T = [\psi_1 \cdots \psi_i \cdots \psi_n]$, $A_i = [a_{1i}\ a_{2i}\ \cdots\ a_{mi}]^T$, and $\psi_i = [v_{i1}\ v_{i2}\ \cdots\ v_{in}]^T$. Then we have

$$\mathbf{A}^T\mathbf{A} = \mathbf{V\Sigma U}^T \cdot \mathbf{U\Sigma V}^T = \mathbf{V\Sigma}^2\mathbf{V}^T \tag{18}$$

The last step makes use of the orthonormal property of matrix $\mathbf{U}$. In the above equation, the diagonal element i of the left-hand side matrix $\mathbf{A}^T\mathbf{A}$ equals $A_i \cdot A_i = \sum_i^m a_{ji}^2$, and the diagonal element i of the right-hand side matrix $\mathbf{V\Sigma}^2\mathbf{V}^T$ equals $\sum_{j=1}^n \sigma_j^2 v_{ij}^2$. Therefore, we have $||\psi_i||_\Sigma^2 = A_i \cdot A_i = \sum_{j=1}^m a_{ji}^2$.

References

1. Chang S-F, Eleftheriadis A, McClintock R (1998) Next-generation content representation, creation and searching for new media applications in education. IEEE Proc (special issue on multimedia signal processing) 86:884–904
2. Tonomura Y, Akutsu A, Otsuji K, Sadakata T (1993) Videomap and videospaceicon: Tools for anatomizing video Content. Proceedings ACM INTERCHI'93, Amsterdam, The Netherlands
3. Ueda H, Miyatake T, Yoshizawa S (1991) Impact: an interactive natural-motion-picture dedicated multimedia authoring system. Proceedings ACM SIGCHI'91, New Orleans, LA
4. Fermain A, Tekalp A (1997) Multiscale content extraction and representation for video indexing. Proc SPIE 3229: pp 23–31
5. DeMenthon D, Kobla V, Doermann D (1998) Video summarization by curve simplification. Technical Report LAMP-TR-018, Language and Media Processing laboratory, University of Maryland, MD
6. Yeung M, Yeo B, Wolf W, Liu B (1995) Video browsing using clustering and scene transitions on compressed sequences. Proc SPIE 2417: pp 399–413
7. Girgensohn A, Boreczky J (1999) Time-constrained keyframe selection technique. Proc IEEE Multimedia Computing and Systems (ICMCS'99), Florence, Italy
8. Farin D, Effelsberg W, de With PH (2002) Robust clustering-based video summarization with integration of domain knowledge. Proceedings IEEE International Conference an Multimedia and Expo, Lausanne, Switzerland
9. Irani M, P. Anandan) (1998) Video indexing based on mosaic representations. IEEE Trans Pattern Analy Machine Intell 86(5): pp 905–921
10. Gelgon M, Bouthemy P (1998) Determining a structured spatio-temporal representation of video content for efficient visualization and indexing. Proccedings European Conference on Computer Vision (ECCV), Freiburg, Germany
11. Aner A, Tang L, Kender JR (2002) A method and browser for cross-referenced video summaries. Proceedings IEEE International Conference on Multimedia and Expo, Lausanne, Switzerland
12. Ponceleon D, Amir A, Srinivasan S, Mahmood T, Petkovic D (1999) Cuevideo: Automated multimedia indexing and retrieval. Proceedings ACM Multimedia, Orlando, FL
13. Smith MA, Kanade T (1997) Video skimming and characterization through the combination of image and language understanding techniques. Proceedings CVPR'97, Puerto Rico, pp 775–781
14. Gong Y, Sin LT, Chuan CH, Zhang H, Sakauchi M (1995) Automatic parsing of tv soccer programs. IEEE International Conference on Multimedia Computing and Systems, Boston, Massachusetts, pp 167–174
15. Xu P, Xie L, Chang SF, Divakaran A,Vetro A, Sun H (2001) Algorithms and system for segmentation and structure analysis in soccer video. IEEE Conference on Multimedia and Expo, Tokyo, Japan,pp 928–931
16. Zhong D, Chang SF (2001) Structure analysis of sports video using domain models. IEEE Conference on Multimedia and Expo, Tokyo, Japan, pp 920–923
17. Rui Y, Gupta A, Acero A (2000) Automatically extracting highlights for tv baseball programs. Eighth ACM International Conference on Multimedia, Los Angeles, California, pp 105–115
18. Swain MJ (1993) Interactive indexing into image databases. SPIE 1908: pp 95-103
19. Press W et al. (1992) Numerical recipes in C: the art of scientific computing, 2nd ed. Cambridge University Press,Cambridge
20. Golub G, Loan C (1989) Matrix computations, 2nd ed. Johns-Hopkins, Baltimore
21. Deerwester S, Dumais S, Furnas G, Landauer T, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Infor Sci 41:391–407