Video Rushes Summarization Using Spectral Clustering and Sequence Alignment

Vasileios Chasanis
Department of Computer Science
University of Ioannina
GR 45110
vchasani@cs.uoi.gr

Aristidis Likas
Department of Computer Science
University of Ioannina
GR 45110
arly@cs.uoi.gr

Nikolaos Galatsanos
Department of Computer Science
University of Ioannina
GR 45110
galatsanos@cs.uoi.gr

ABSTRACT

In this paper we describe a system for video rushes summarization. The basic problems of rushes videos are three. First, the presence of useless frames such as colorbars, monochrome frames and frames containing clapboards. Second, the repetition of similar segments produced from multiple takes of the same scene and finally, the efficient representation of the original video in the video summary. In the method we proposed herein, the input video is segmented into shots. Then, colorbars and monochrome frames are removed by checking their edge direction histogram, whereas frames containing clapboards are removed by checking their SIFT descriptors. Next, an enhanced spectral clustering algorithm that both estimates the number of clusters and employs the fast global k-means algorithm in the clustering stage after the eigenvector computation of the similarity matrix is used to extract the key-frames of each shot, to efficiently represent shot content. Similar shots are clustered in one group by comparing their key-frames using a sequence alignment algorithm. Each group is represented from the shot with the largest duration and the final video summary is generated by concatenating frames around the key-frames of each shot. Experiments on TRECVID 2008 Test Data indicate that our method exhibits good performance.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing, Abstracting methods; I.5.3 [**Pattern Recognition**]: Clustering, Algorithms

General Terms

Algorithms, Experimentation, Performance

Keywords

Video summarization, global k-means, key-frame extraction, spectral clustering

TVS'08, October 31, 2008, Vancouver, British Columbia, Canada.
Copyright 2008 ACM 978-1-60558-309-9/08/10 ...$5.00.

1. INTRODUCTION

The huge amount of data produced from several applications such as internet-TV, video on demand and security systems in addition to the production of thousand movies and documentaries every year requires the implementation of efficient summarization, indexing and browsing tools. In this paper, we focus on the problem of video rushes summarization. The main challenges of this task are two. First, the creation of summarization tools that will provide efficient representation and indexing of the original video with minimum resources. Second, the development of methods to recognize similar segments in a video, which could be further used for an efficient video retrieval system.

Three issues should be considered during the rushes summarization process. The first one is that useless frames such as colorbars, monochrome frames and frames that contain clapboards should be removed from the video. The second issue is that similar segments generated from multiple takes of the same scene should be removed keeping only one representative segment. The third issue is the efficient representation of the content of each of the selected representative shots and the creation of the final video summary.

In the method we proposed herein each video is initially segmented into the smallest video segments which are the shots by comparing the normalized histograms of adjacent video frames. Then, an enhanced spectral clustering algorithm is employed for key-frame extraction that both estimates the number of clusters and uses the fast global k-means algorithm in the clustering stage after the eigenvector computation of the similarity matrix. Next, useless frames such as colorbars and monochrome frames are removed by checking their edge direction histogram. Rushes video contain redundant information, since the same scene is taken many times until the desired result is produced. To find similar segments (shots) in the rushes video, the key-frames of shots are compared using a sequence alignment algorithm. These similar shots that describe the same scene are removed and only one of them is kept to contribute to the final video summary. Moreover, key-frames that contain clapboards should be removed from the final representative shots. Comparing the SIFT descriptors of the key-frames of each shot with the SIFT descriptors of a database of clapboards, we are able to check if a key frame contains a clapboard and remove it. Finally, to produce the video summary with duration less than a percentage of the duration of the original video, a number of frames around each key frame of the selected shots are considered to contribute to the final video summary. In *Fig. 1* we summarize the main steps of our approach and the algorithms employed in these steps.

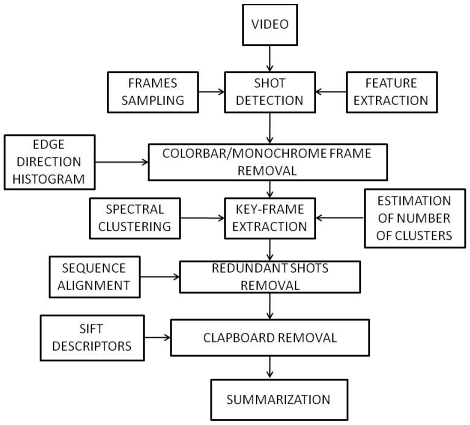


Figure 1: The main steps of our method.

The rest of the paper is organized as follows: In section 2 the procedure for extracting shot key-frames is described. In section 3 the removal of useless frames is presented. In sections 4 and 5 we present methods for the detection of similar segments (shots) and the removal of clapboards respectively, while in section 6 the final summarization process is described. Experiments on TRECVID 2008 Test Data are provided in section 7 and finally, in section 8 we conclude our work.

2. VIDEO REPRESENTATION

The first level of video processing is the segmentation of the video into shots and the extraction of features for each frame. Then, the efficient representation of each shot is required to continue with the summarization process.

2.1 Feature Extraction and Shot Detection

Each video is sampled uniformly keeping only 5 frames per second. Then, for each frame an HSV normalized histogram is used, with 8 bins for hue and 4 bins for each of saturation and value, resulting to $8 \times 4 \times 4$ bins. To detect shot boundaries we calculate the sum of the bin-wise differences of adjacent frames and compare them to a threshold. We use a variation of x^2 to compare the histograms of two frames in order to enhance the difference between the two histograms. Finally the difference between two images I_i, I_j based on their color histograms H_i, H_j is given from the following equation:

$$d(I_i, I_j) = \sum_{k=1}^{128} \frac{(H_i(k) - H_j(k))^2}{H_i(k) + H_j(k)} \ , \tag{1}$$

where k denotes the bin index. A shot boundary is defined at frame I_i if $d(I_i, I_j)$ is greater than a threshold T_{sh}, which in our experiments was set to 0.15. Shots shorter than 1 second were removed.

2.2 Key-frame Extraction

To speed up the summarization process each shot must be represented by unique frames that will capture the whole content of the shot. In this way to compare two shots, we don't use all the frames of each shot but a small number of key-frames that provide a sensible representation of the shot content. To perform key-frame extraction the video frames of a shot are clustered into groups using an improved spectral clustering algorithm. The medoids of the obtained groups are selected as the key-frames of the shot. A medoid is defined as the frame of a group whose average pairwise similarity to all other frames of this group is maximal.

2.2.1 The Typical Spectral Clustering Algorithm

Suppose there is a set of objects $H = H_1, H_2, \ldots, H_N$ to be partitioned into K groups [4], where H_n is the feature vector (normalized color histogram) of the n-th frame.

1. Compute similarity matrix $A \in \mathbb{R}^{N \times N}$ for the pairs of histograms of the data set H.
2. Define D to be the diagonal matrix whose (i, i) element is the sum of the A's i-th row and construct the Laplacian matrix $L = I - D^{-1/2}AD^{-1/2}$.
3. Compute the K principal eigenvectors $x_1, x_2, \ldots, x_K$ of matrix L to build an $N \times K$ matrix $X = [x_1\ x_2 \ldots\ x_K]$.
4. Renormalize each row of X to have unit length and form matrix Y so that:
$$y_{ij} = x_{ij}/(\sum_j x_{ij}^2)^{1/2} \ . \tag{2}$$
5. Cluster the rows of Y into K groups using k-means.
6. Finally, assign object H_i to cluster j if and only if row i of the matrix Y has been assigned to cluster j.

The distance function we consider is the Euclidean distance between the histograms of the frames. As a result each element of the similarity matrix A is computed as follows:

$$a(i, j) = 1 - \sqrt{\sum_{h \in bins} (H_i(h) - H_j(h))^2} \ . \tag{3}$$

2.2.2 Estimation of the Number of Clusters Using Spectral Clustering

Assume we wish to partition dataset H into disjoint subsets $(H^1, \ldots, H^K)$, and let $X = [X_1, \ldots, X_K] \in \mathbb{R}^{N \times K}$ denote the partition matrix, where X_j is the binary indicator vector for set H^j such that:

$$\begin{array}{l} X(i,j) = 1 \ : \ if\ i \in H^j \\ X(i,j) = 0 \ : \ otherwise \end{array} \ . \tag{4}$$

The optimal partition is defined as the optimal solution to the following problem [7]:

$$\begin{array}{c} \max\limits_X \ trace(X^T L X) \\ s.t.\ X^T X = I_K \text{ and } X(i,j) \in \{0,1\} \end{array} \ , \tag{5}$$

where L is the Laplacian matrix defined in section 2.2.1. The spectral clustering algorithm (for K clusters) provides solution to the following relaxed optimization problem:

$$\begin{array}{c} \max\limits_Y \ trace(Y^T L Y) \\ s.t.\ Y^T Y = I_K \end{array} \ . \tag{6}$$

Relaxing Y into the continuous domain turns the discrete problem into a continuous optimization problem. The optimal solution is attained at $Y = U_K$, where the columns u_i of $U_k, i = 1, \ldots, K$, are the eigenvectors corresponding to the ordered top K largest eigenvalues λ_i of L. Since it holds that [7]:

$$\lambda_1 + \lambda_2 + \ldots + \lambda_K = \max_{Y^T Y = I_K} \quad trace(Y^T LY) \ , \qquad (7)$$

the optimization criterion that also quantifies the quality of the solution for K clusters and its corresponding difference for successive values of K are respectively given by:

$$\begin{array}{l} sol(K) = \lambda_1 + \lambda_2 + \ldots + \lambda_K \\ sol(K+1) - sol(K) = \lambda_{K+1} \end{array} \ . \qquad (8)$$

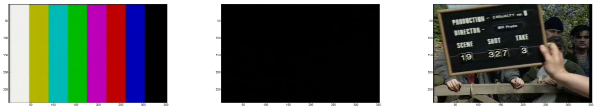
When the improvement in this optimization criterion (i.e. the value of the λ_{K+1} eigenvalue) is below a threshold, improvement by the addition of cluster K+1 is considered negligible, thus the estimate of the number of clusters is assumed to be K. The threshold value that is used in all our experiments was fixed to Th=0.005 with very good results.

2.2.3 *Fast global k-means algorithm*

In our method, in the fifth step of the spectral clustering algorithm instead of using the typical k-means approach, we have used the fast version of the very efficient global k-means algorithm [1]. Global k-means in an incremental deterministic clustering algorithm that overcomes the important initialization problem of the typical k-means approach. Using the global k-means, the obtained key frames usually provide a sensible representation of shot content. Next we briefly review the global k-means algorithm. Suppose we are given a data set $X = x_1, \ldots, x_N$, $x_n \in R^d$ to be partitioned into K disjoint clusters $C_1, C_2, \ldots, C_K$.

This algorithm is incremental in nature. It is based on the idea that the optimal partition into K groups can be obtained through local search (using k-means) starting from an initial state with i) the K-1 centers placed at the optimal positions for the (K-1)-clustering problem and ii) the remaining K-th center placed at an appropriate position within the dataset. Based on this idea, the K-clustering problem is incrementally solved as follows. Starting with k = 1, find the optimal solution which is the centroid of the data set X. To solve the problem with two clusters, the k-means algorithm is executed N times (where N is the size of the data set) from the following initial positions of the cluster centers: the first cluster center is always placed at the optimal position for the problem with k = 1, whereas the second center at execution n is initially placed at the position of data x_n. The best solution obtained after the N executions of k-means is considered as the solution for k = 2. In general if we want to solve the problem with k clusters, N runs of the k-means algorithm are performed, where each run n starts with the k-1 centers initially placed at the positions corresponding to the the solution obtained for the (k-1)-clustering problem, while the k-th center is initially placed at the position of data x_n. A great benefit of this algorithm is that it provides the solutions for all k-clustering problems with $k \leq K$.

The fast global k-means algorithm reduces the computational cost of the global k-means algorithm without significant loss in the quality of the solution [1]. Initially, a new cluster center is placed at position x_n and an upper bound E_n of the final clustering error obtained. The initial position of the new cluster center is selected as the point x_i for which E_n is minimum and the k-means runs only once for each k.

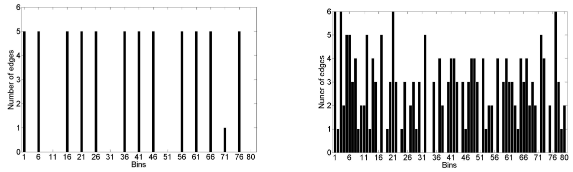


(a) Colorbar (b) Monochrome (c) Clapboard

Figure 2: Useless frames.

3. USELESS FRAMES DETECTION

Video rushes contain many useless frames such as colorbars and monochrome frames (see *Fig. 2(a), 2(b)*), which are not necessary for the final summarization and should be removed. The shot detection algorithm usually isolates colorbars or monochrome frames into single shots, thus to speed up the implementation process the first key-frame of each shot is checked and if it is defined as useless frame, the corresponding shot is removed from the summarization process. To check whether a key-frame is useless or not we calculate its edge direction histogram [3]. The key-frame is first subdivided into sub-images, and local edge histograms for each of these sub-images is computed. Edges are grouped into five categories: vertical, horizontal, 45 diagonal, 135 diagonal, and isotropic (nonorientation specific). Thus, each local histogram has five bins corresponding to the above five categories resulting in a 80-bin histogram for the whole frame.



(a) Colorbar (b) Normal frame

Figure 3: Edge direction histograms.

In *Fig. 3* we provide the edge direction histograms for a colorbar (a) and a normal frame (b). The edge direction histogram of a colorbar produces peaks in vertical and horizontal bins whereas the other bins are close to zero. The bins of the edge direction histogram of a monochrome frame are all close to zero. Thus a colorbar or monochrome frame is detected if the difference between the sum of all bins of the edge histogram and the sum of the vertical and horizontal bins is lower that a threshold T_{edh}:

$$\sum_{k=1}^{128} E_i(k) - \sum_{m=0}^{15} E_i(5m+1) - \sum_{m=0}^{15} E_i(5m+2) < T_{edh} \ , \quad (9)$$

where E_i is the edge direction histogram of frame I_i and $E_i(5m+1)$, $E_i(5m+2)$, $m = 0, \ldots 15$ are the vertical and horizontal bins of the histogram respectively. In our experiments T_{edh} was set to 10.

4. REDUNDANT INFORMATION REMOVAL

Rushes often contain repetitive information, since the same scene is usually taken many times until the desired result is produced. Our goal is to detect similar segments which in our case are shots and keep only one representative for each group of similar shots that will be further analyzed to contribute to the final summary.

4.1 Visual Shot Similarity Metric

Once we have removed the shots that correspond to colorbars or monochrome frames we need to suggest a proper visual shot similarity metric. Suppose we are given two shots S_i and S_j and $KF_i = \{KF_i^1,\ KF_i^2, \ldots,\ KF_i^m\}$, $KF_j = \{KF_j^1,\ KF_j^2, \ldots,\ KF_j^n\}$ their corresponding key-frame sets. An $m \times n$ similarity matrix SM is constructed with elements:

$$SM(m,n) = VisSim(KF_i^m, KF_j^n)\ , \tag{10}$$

where $VisSim$ is the visual similarity between two frames I_i and I_j given by the following equation:

$$VisSim(I_i, I_j) = 1 - d(I_i, I_j)\ , \tag{11}$$

with $d(I_i, I_j)$ defined in equation 1 and $VisSim \in [0,1]$.

Taken into consideration that in rushes two shots that describe the same scene are similar, we expect that their key frames will follow the same order. Thus, it is expected that either a segment of one shot or the whole shot will also appear in the other shot. To find similar segments in two shots we use a sequence alignment algorithm between the sets of their key-frames. In this way a key-frame is "matched" with the most similar (visually) key-frame of the the other set of key-frames, while also taking into consideration the temporal order of key-frames. Suppose that the first shot describes the following time ordered events $E_1, E_2, E_3, E_4, E_5, E_6$ and the second shot describes events E_2, E_3, E_5, E_6. An optimal alignment of the two shots is presented in *Fig.* 4.

$$\begin{array}{lcl} Seq_1 & : & E_1E_2E_3E_4E_5E_6 \\ Seq_2 & : & E_2E_3E_5E_6 \end{array}$$

Seq_1	E_1	E_2	E_3	E_4	E_5	E_6
Seq_2	–	E_2	E_3	–	E_5	E_6

Figure 4: Sequence alignment example

The score of the sequence alignment constitutes the final shot similarity metric. To align two sequences we use the "Smith-Waterman" algorithm [6]. This method requires a substitution matrix which in our case is given by similarity matrix SM. The score of each alignment is normalized to be in range of 0-1.

$$Score_N = Score/min(n_{kf1}, n_{kf2})\ , \tag{12}$$

where n_{kf1}, n_{kf2} are the numbers of key-frames of the two shots under alignment respectively.

4.2 Repetitive shot detection

To find groups of repetitive and similar shots we compared each shot with the next three. If one of the three shots is similar with the shot under consideration then all the shots between these two shots and the shots under comparison, form a group. If none of the shots is similar then a new group of shots is considered and the algorithm continues until all shots are examined. Two shots are considered similar if the score of the sequence alignment of their sets of key-frames exceeds a predefined threshold which in our experiments was set to 0.88 (experimentally selected using TRECVID 2007 Development Data). Finally, the shot of each group with the largest duration is selected as the representative of this group.

5. CLAPBOARD REMOVAL

So far, we have selected unique and non-repetitive shots which are represented by their key-frames. Rushes also contain clapboards to indicate the current number of the shot (see *Fig. 2(c)*). These frames should not be included in the final summarization, thus they have to be removed. These clapboards usually appear at the beginning of each shot take. To detect clapboards we compute for each key-frame the scale-invariant feature transforms (SIFT) [2]. Using the TRECVID 2007 Development Data, a database of approximately 150 frames containing only clapboards was generated and their SIFT descriptors were calculated. In order to detect whether a key-frame contains a clapboard, we compute its SIFT descriptors and compare them with the SIFT descriptors of the database. If the number of matching descriptors is over a predefined threshold, this key-frame is characterized as clapboard and the cluster corresponding to this key-frame is removed from the shot. Having checked all the key-frames of a shot and having removed those key-frames characterized as clapboards and their corresponding clusters, we extract new key-frames for the shot using the method described in section 2.

6. SUMMARIZATION

The final stage of our summarization method involves the production of the final video summary. The method we described so far has produced unique, non-repetitive shots that are represented from their time-ordered key-frames. A number of frames around each key frame of the selected shots are considered to contribute to the final video summary. The goal of the rushes summarization process is to create a video summary with duration less than p% of the original video duration. Once the repetitive shots have been detected (Section 4), the shot with the largest duration is selected as their representative. The duration of a group of similar shots is referred as T_{all}. We want the duration of the summarized video T_{sum}, for the specific group, to be p% of T_{all}. Suppose that this shot is represented from k key-frames. A duration of $T_{kf} = T_{sum}/k$ is assigned to each key-frame. Finally, sampling every 3 frames, the $\lfloor T_{kf}/2 \rfloor$ preceding and $\lfloor T_{kf}/2 \rfloor$ following frames of each key-frame are selected to summarize the shot under consideration.

7. EXPERIMENTS

We have tested our method on TRECVID 2008 Test Data, under the Rushes Summarization task of TRECVID 2008. The performance of our method was tested on 40 videos. The goal of this task is to produce video summaries with duration less than or equal to $p = 2\%$ of the duration of the original video. Precise details of the results can be found in

Table 1: Performance of our video rushes summarization method.

	Our method		All	
	Mean	Median	Avg.(Mean)	Avg.(Median)
DU (secs)	25.07	28.00	27.01	28.25
XD (secs)	6.64	5.17	4.69	3.93
TT (secs)	39.86	41.33	40.76	39.91
VT (secs)	27.57	30.33	29.31	30.47
IN (0-1)	0.53	0.56	0.44	0.44
JU (1-5)	3.31	3.33	3.17	3.21
RE (1-5)	3.16	3.33	3.3	3.36
TE (1-5)	2.50	2.33	2.76	2.75

[5]. In *Table 1* we present the scores of our method for the different measures: (DU) - duration of the summary (secs), (XD) - difference between target and actual summary size (target-actual) (secs), (TT) - total time spent judging the inclusions (secs), (VT) - total video play time (versus pause) judging the inclusions (secs), (IN) - fraction of inclusions found in the summary (0 - 1), (JU) - Summary contained lots of junk, (RE) - Summary contained lots of duplicate video, (TE) - Summary had a pleasant tempo/rhythm [5]. We also present the average mean and median for all groups participated in the same task. It is worth mentioning that the proposed key-frame extraction algorithm efficiently summarizes the content of a shot, which is indicated from the high fraction of inclusions found in the summary (IN). In what concerns the removal of useless frames (JU), we observe that the results of our method are above the average. However, clapboard removal could be further investigated and improved. Finally, the identification of repetitive information (RE) also needs improvement as indicated from the results.

8. CONCLUSION

In this paper we presented a method for video rushes summarization. Useless frames are removed from each video using edge direction histograms for colorbars-monochrome frames and SIFT descriptors for clapboards. Redundant segments of the video are also removed using sequence alignment keeping only one representative. Also, an improved spectral clustering algorithm was used to extract the key-frames of each shot. Our system exhibited good performance in the Rushes Summarization task of TRECVID 2008. Some issues like clapboard detection and identification of repetitive information should be further improved in the future using more sophisticated approaches.

Acknowledgments

This research project (PENED) is co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%).

9. REFERENCES

[1] A. Likas, N. Vlassis, and J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(6):451–461, Feb 2003.

[2] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[3] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715, Jun 2001.

[4] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm, 2001.

[5] P. Over, A. F. Smeaton, and G. Awad. The TRECVid 2008 BBC rushes summarization evaluation. In *TVS '08: Proceedings of the International Workshop on TRECVID Video Summarization*, pages 1–20, New York, NY, USA, 2008. ACM.

[6] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[7] E. P. Xing and M. I. Jordan. On semidefinite relaxation for normalized k-cut and connections to spectral clustering. Technical Report UCB/CSD-03-1265, EECS Department, University of California, Berkeley, Jun 2003.