Content-Based Video Abstraction *

Bilge Günsel and A. Murat Tekalp

Department of Electrical Engineering and Center for Electronic Imaging Systems
University of Rochester, Rochester, NY 14627-0126
E-mail: {gunsel,tekalp }@ee.rochester.edu

Abstract

This paper addresses automatic scene change detection, key-frame selection, and similarity ranking which constitute the main steps of a content-based video abstraction system. Unlike other methods, the proposed algorithm performs scene change detection and key-frame selection in one step. We treat scene change detection as a two-class classification problem and employ automatic threshold selection techniques originally developed for image binarization. A quantitative measure for retrieval of similar scenes according to their color content is also defined. The described scheme can be applied to both uncompressed and MPEG compressed video, and can be implemented in real-time. Performance of the algorithm has been analyzed on real TV sequences, and comparison with some previously introduced techniques are provided.

Key words: Content-based access, video content summarization, video databases.

1. Introduction

Recent advances in hardware and signal processing techniques are enabling interactive digital video applications with new functionalities, including random access to desired video scenes, automatic recording of important scenes of one program while watching another program, quick browsing of simultaneous broadcast TV programs by using visual summaries, and so on. Automatic video abstraction refers to processing of video sources to extract description schemes and visual summaries to enable these functionalities.

Scene change detection and visual summarization constitute two of the main steps of video content representation for abstraction. The abstraction system should also provide the necessary cues to a search engine in order to allow retrieval of similar shots. Thus, fundamental problems in video content representation and abstraction are: automatic scene change detection (temporal video segmentation), selection of minimum number of key frames that represent the visual content of the scene, and definition of similarity measures between video scenes in terms of content-based quantitative descriptors. Current techniques mostly require intensive supervision and high computational resources. Other difficulties may arise if the video source is in compressed format. This paper proposes a practical system for automatic video abstraction. The proposed scheme can be applied to both uncompressed and MPEG compressed video sources, and can be implemented in real-time.

2. Detection and Visual Summarization of Video Scenes

Unlike most of the reported work in the literature, we propose an algorithm that performs scene change detection and key-frame selection in one step by using a two dimensional feature vector. The scheme is based on color similarity analysis in the YUV space, because color is a consistent characteristic of each shot [1] and also can be extracted from uncompressed as well as the compressed video (e.g., DC images [2]).

Let $\boldsymbol{f_t} = (f_t^1\ f_t^2)^T$ be a two dimensional feature vector representing the color content of a video frame at time t. We define the first component of the feature vector as the color histogram difference between the successive frames at t and $t+1$, given by

$$f_t^1 = \sum_{i=0}^{G-1}(|H_{t+1}^Y(i) - H_t^Y(i)| + |H_{t+1}^U(i) - H_t^U(i)| + |H_{t+1}^V(i) - H_t^V(i)|). \quad (1)$$

In Eqn. 1, $H_t^Y(i)$, $H_t^U(i)$, and $H_t^V(i)$ denote the color histogram of Y, U, and V components, respectively, of frame t, and i denotes one of the G bins. The component f_t^1 simply provides a quantitative measure

*This work is supported by a National Science Foundation SIUCRC grant and a New York State Science and Technology Foundation grant to the Center for Electronic Imaging Systems at the University of Rochester.

of similarity of color between successive frames. The component f_t^2 is defined as the difference between the color histogram of current frame t and the mean histogram of previous k frames, given by

$$f_t^2 = \sum_{i=0}^{G-1}(|H_t^Y(i) - H_{mean_k}^Y(i)| + |H_t^U(i) - H_{mean_k}^U(i)| + |H_t^V(i) - H_{mean_k}^V(i)|) \quad (2)$$

where the means of histograms of Y, U and V components are defined as:

$$\begin{aligned} H_{mean_k}^Y(i) &= \frac{1}{k}\sum_{f=1}^{k} H_f^Y(i), \\ H_{mean_k}^U(i) &= \frac{1}{k}\sum_{f=1}^{k} H_f^U(i), \\ H_{mean_k}^V(i) &= \frac{1}{k}\sum_{f=1}^{k} H_f^V(i). \end{aligned} \quad (3)$$

Thus, f_t^2 includes information about color similarities of the last k frames. Note that k is not a prespecified constant, and varies adaptively depending on the content. For example, k is small in a sports clip, while it is large for a news clip.

The proposed system labels a video frame t as scene change point if $f_t^1 > T$, where T is the scene change detection threshold. Also the frame t is selected as a key-frame. If $f_t^1 < T$, then the system checks whether $f_t^2 < T$, and labels the frame as a key-frame if $f_t^2 > T$. Note that, f_t^2 can take values greater than T even if $f_t^1 < T$ at smooth scene change points, e.g., edit effects. The second check provides uniform (in terms of color content) temporal segments in which f_t^2 remains less than T, and the first frame of each uniform region is selected as a key-frame.

Previous work shows that thresholding of histogram differences (f_t^1) using a single threshold is not adequate to detect smoothly changing scenes (e.g., dissolves). Multithresholding techniques which require more than one pass over the video are proposed to overcome this problem [1]. However, none of these techniques brings a practical solution to deal with the key-frame selection problem. In our scheme, the first feature f_t^1 allows detection of abrupt scene changes. The role of the second feature, f_t^2, is twofold: it guarantees detection of significant changes in a scene, and allows specification of uniform temporal video regions in which the first frame of each segment is selected as a key-frame. Clearly, in our scheme, the number of selected key-frames may be greater than the number of shots. However, these extra key-frames help provide a better visual summary of clips. Note that, if so desired, they can be eliminated by using two thresholds, where the threshold on f_t^2 is greater than T. Observe that our proposed scheme employs a single threshold T.

3. Automatic Threshold Specification

Automatic selection of an appropriate scene change threshold, T, is also addressed in this paper. We treat the scene change detection problem as a two-class classification problem and suggest to employ automatic threshold selection techniques originally developed for image binarization. Note that unlike the classical case, we deal with histogram differences here rather than gray levels. We found the method of Otsu, which is a global point-dependent thresholding technique, most useful to specify the scene change detection threshold automatically. It is shown in [3] that Otsu method, which is based on discriminant analysis, provides high performance in terms of uniformity of thresholded regions (uniform temporal segments) and correctness of segmentation boundaries (accurately localized shot boundaries).

Theoretically Otsu method aims to minimize the weighted sum of class variances to specify the optimal threshold. Therefore, one way to determine optimal threshold is minimization of μ with respect to T where μ is formulated as $\mu = \sigma_B^2/\sigma_{tot}^2$ [3]. Here, σ_B^2 and σ_{tot}^2 denote between-class variance and the total variance, respectively. Thus the optimal threshold $T^\star$ can be specified as

$$T^\star = arg \min_{T \epsilon D} \mu, \qquad D: \{f_t^1, t = 1, 2, .., \#frames\} \quad (4)$$

where

$$\begin{aligned} \sigma_{tot}^2 &= \sum_{i=0}^{d}(i - \mu_{tot})^2 p_i, \qquad \mu_{tot} = \sum_{i=0}^{d} i p_i, \\ \sigma_B^2 &= \omega_1\omega_2(\mu_1\mu_2)^2, \qquad \omega_1 = \sum_{i=0}^{T} p_i, \qquad \omega_2 = 1 - \omega_1, \\ \mu_2 &= \frac{\mu_{tot} - \mu_T}{1 - \omega_1}, \qquad \mu_1 = \frac{\mu_T}{\omega_1}, \qquad \mu_T = \sum_{i=0}^{T} i p_i. \end{aligned} \quad (5)$$

Here d is the largest distance computed between two video frames, and p_i is the probability of occurrence of a histogram distance value i.

4. Retrieval of Similar Scenes

Similarity analysis aims to define a quantitative measure for the similarity of scenes according to certain criteria, which is color content in our work. The defined quantitative measure will be provided to search engines to allow retrieval of scenes according to ranked similarities. To this effect, an $N \times N$ similarity matrix $\boldsymbol{S} = [S_{ij}]$ is constructed [4]. Here N is the total number of detected scene changes and S_{ij} denotes the similarity between the ith and jth scenes. Note that the video scenes correspond only to temporal segments detected by the feature f_t^1. $\boldsymbol{S}$ is a positive symmetric matrix and S_{ij} values are derived as:

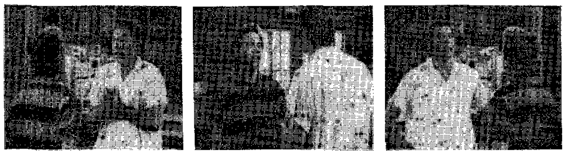
$$S_{ij} = \sum_{k=1}^{P} \sum_{l=1}^{R} \max s_{kl} \qquad (6)$$

where s_{ij} is a similarity metric between two uniform temporal segments provided by the feature f_t^2, P and R denote the number of these uniform segments included in the scene i and j, respectively. Recall that a scene may include more than one uniform segment. Similarities between uniform temporal segments, s_{ij}'s, are defined by the similarities of their key-frames. Hence we have used the color histogram differences between the key-frames of uniform segments, thus maximum similarity corresponds to minimum histogram difference. Similarity matrix $\boldsymbol{S}$ provides a quantitative measure that can be used to rank retrieval outcomes. Note that the similarity measure depends on the histogram differences that are already computed for scene change detection. Alternatively, in [5], clustering of camera shots based on the similarity measures of visual primitives such as color luminance correlation has been proposed. In [6], a "time-constrained clustering," which employs two similarity metrics to take into account both visual characteristics and temporal locality of shots, has been proposed. In [7] maximization over the similarities between each frame of each shot is proposed.

5. Results

Results are obtained in the YUV color space on decompressed MPEG1 video sequences including digitized TV programs such as sitcoms, sports (basketball), cartoons and commercials. The sampling rate is 15 frames/sec and the size of video frames is 240 × 180. For each program type, a scene change detection threshold is specified by applying the Otsu method over a 1 minute recorded program. It is assumed that the longest shot (mostly a sitcom scene) will be shorter than 40 sec. The automatically specified thresholds are given at the second row of Table-1. Note that the smallest threshold is obtained for the sports program as 14%, which means a scene change is declared when the histogram difference between two frames is greater than 14% of the maximum difference. Obviously the maximum difference is proportional to the bin size and the size of the video frame. Thus for a specific type of program, we just need to recompute the threshold when the image size or bin size is changed. Our extensive experiments show that specified thresholds are appropriate to detect scene changes of the same type of programs digitized from different channels. Note that a Sparc20 workstation is capable of computing Otsu thresholds in real time. The number of key-frames detected by the proposed method (histogram difference plus difference from mean histogram - HD+MHD) for 3500 frames/program is given in the second row of Table-1. These key-frames are selected using the specified Otsu thresholds.

Performance of the developed algorithm is compared to 2-class clustering method introduced in [8]. The third row of Table-1 gives the cluster means obtained by 2-class K-means clustering of histogram differences (f_t^1) of the same 3500 frames/program. Corresponding key frame numbers are shown at the fifth row of Table-1. Both the proposed method (HD+MHD) and the clustering detect reasonable number of key frames and reduce the number of frames at least 17 times (the number of original scene changes is given at the last row of Table-1). However clustering requires to process the entire sequence therefore it is appropriate to off-line applications such as indexing of video tapes recorded by VCR or indexing of home type of video.



©*Courtesy NBC Television*

S	(1)	(2)	(3)
(1)	0.0	0.558	0.361
(2)	0.558	0.0	0.549
(3)	0.361	0.549	0.0

Figure 1: From left to right; key-frames representing three different camera shots, 1360th frame (1), 1456th frame (2), 1549th frame (3). Computed similarity matrix between three shots.

Table-2 summarizes the scene change detection and key-frame selection results obtained on different TV programs, each 3500 frames. For each program type, percentage of hits, false positives, false negatives, and the number of selected key-frames are reported for three different methods. The first method corresponds to scene change detection by thresholding color histogram differences (HD) and the first frame of each scene is selected as a key-frame. The second method, HD+MHD, is the proposed scheme. As it is described in Section 2, this method selects the first frame of each uniform segment as a key-frame. Note that both methods employ thresholds specified by Otsu thresholding. The third method, clusters the histogram differences computed within the entire sequence (3500 frames) into two classes, "scene change" and "no scene change" and labels the first frame of each scene as a key-frame. Note that the threshold specified for sitcoms is used for the entire sequence, although sitcom sequence includes commercials in it. According to the results obtained by HD and HD+MHD, it can be concluded that HD+MHD increases the number of hits and allows detection of smooth scene changes (decreases the number of false negatives). However, it also increases the number of false positives that results in extra key-frames which mostly correspond to the scenes including more than one uniform temporal segment. This is an advantage to provide a better visual summary, because the selected key-frames correspond to meaningfull discontinuities (i.e., exit/entry of objects). It is also concluded that the HD+MHD method provides high performance for all types of programs. Clustering gives very high performance for cartoons, however fails on the sport programs and provides less accuracy for the Sitcom&Commercial. This is because scene changes of cartoon are mostly clean cuts; therefore, the distribution of color histogram differences is a bimodal function which increases the performance of 2-class clustering as well as the other methods. Note that localization of edit effects is out of the scope of this paper; however, we can easily integrate the edit effect detection method proposed in [4] with our system.

Fig. 1 illustrates three key-frames and gives the similarity matrix $\boldsymbol{S}$ computed between the scene contents represented by these key-frames. Similarities are computed using Eqn. 6. It can be concluded that the similarity ranking results are consistent with human perception. Fig. 2 illustrates first-frame of nine different video scenes retrieved according to the color similarities with the clip represented by the key frame shown at the upper left corner of the figure. It should be pointed out that the ranked retrieval outcomes are consistent with human perception except the commercial clip (6th outcome). Note that among 202 scenes the number of sitcom scenes is 43 in this experiment.

We have performed extensive experiments using the DC images obtained from MPEG1 compressed data. Current results show that the developed method can be applicable to DC images and computation time decreases significantly. Obviously accuracy of working in uncompressed domain is higher but results obtained on DC images of sitcom and cartoon are quite good. However, none of the methods provides adequate performance for sport programs. Future work includes to improve the performance in compressed domain by using more than one threshold.

References

[1] B. Furht, S.W. Smoliar, and H. Zhang. *Video and Image Processing in Multimedia Systems.* Kluwer Academic Publishers, 1995.

[2] B.-L. Yeo and B. Liu. Rapid scene analysis on compressed video. *IEEE Trans. on Circuits and Systems for Video Technology*, 5:533–544, Dec. 1995.

[3] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen. A survey of thresholding techniques. *CVGIP*, 41:233–260, 1988.

[4] B. Günsel, Y. Fu, and A. M. Tekalp. Hierarchical temporal video segmentation and content characterization. In *Proc. SPIE:Storage and Retrieval*, pages 953–963, vol. 3024, Dallas, USA, November 1997.

[5] F. Arman, A. Hsu, and M.Y. Chiu. Image processing on compressed data for large video databases. In *Proc. 1st ACM Int. Conf. on Multimedia*, pages 267–272, CA, 1993.

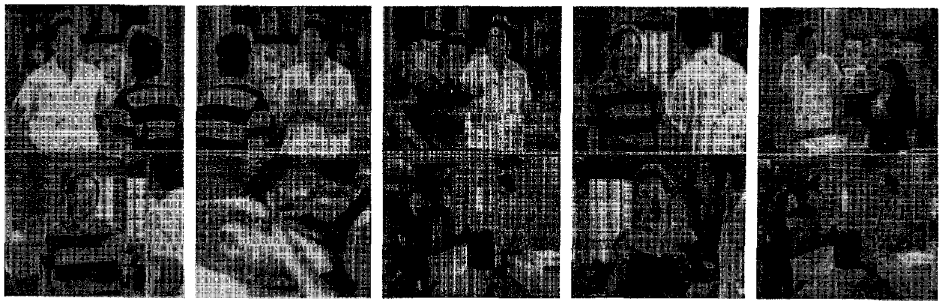
[6] S. F. Chang and J. R. Smith. Extracting multi-dimensional signal features for content-based visual query. In *Proc. SPIE*, volume 2501, pages 995–1006, 1995.

[7] H.J. Zhang, J. Wu, D. Zhong, and S.W. Somaliar. An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, 30, No.4:643–658, April, 1997.

[8] B. Günsel, A.M. Ferman, and A. M. Tekalp. Video indexing through integration of syntactic and semantic features. In *Proc. IEEE Workshop on Applications of Computer Vision*, pages 90–95, Florida, USA, 1996.

Table-1 Automatically specified scene change detection thresholds, cluster means and the number of respective key frames.

Program Type	**Sitcom**	**Sport**	**Cartoon**
Otsu Threshold	23%	14%	31%
Nr.of Key-Frms.	202	150	75
Cluster Means	C1=0.168, C2=0.108	C1=0.147, C2=1.117	C1=0.202, C2=1.197
Nr. of Key-Frms.	134	45	66
Nr. of original scenes	126	55	64



© *Courtesy NBC Television*

Figure 2: First nine outcomes of retrieval according to color indexes. Query example is the key-frame shown at the upper left corner of the figure.

Table-2 Scene change detection and key-frame selection results obtained by histogram difference thresholding, the proposed algorithm and 2-class K-means clustering.

Program Type	**Method**	**Hits (%)**	**False+ (%)**	**False- (%)**	**Key-Fr.#**
Sitcom & Commer.	HD	86	16	14	129
	HD + MHD	92	46	8	202
	Clustering	88	3	12	134
Sport	HD	96	24	4	71
	HD + MHD	98	65	2	150
	Clustering	75	11	25	45
Cartoon	HD	89	0.0	11	47
	HD + MHD	97	17	3	76
	Clustering	96	2	4	66