Hierarchical temporal video segmentation and content characterization

Bilge Gunsel, Yue Fu, A. Murat Tekalp

Bilge Gunsel, Yue Fu, A. Murat Tekalp, "Hierarchical temporal video segmentation and content characterization," Proc. SPIE 3229, Multimedia Storage and Archiving Systems II, (6 October 1997); doi: 10.1117/12.290364

SPIE.

Event: Voice, Video, and Data Communications, 1997, Dallas, TX, United States

Hierarchical Temporal Video Segmentation and Content Characterization *

Bilge Günsel, Yue Fu and A. Murat Tekalp
Dept. of Electrical Engineering and Center for Electronic Imaging Systems
University of Rochester, Rochester, NY 14627
E-mail: gunsel@ee.rochester.edu

ABSTRACT

This paper addresses the segmentation of a video sequence into shots, specification of edit effects and subsequent characterization of shots in terms of color and motion content. The proposed scheme uses DC images extracted from MPEG compressed video and performs an unsupervised clustering for the extraction of camera shots. The specification of edit effects, such as fade-in/out and dissolve is based on the analysis of distribution of mean value for the luminance components. This step is followed by the representation of visual content of temporal segments in terms of key frames selected by similarity analysis of mean color histograms. For characterization of the similar temporal segments, motion and color characteristics are classified into different categories using a set of different features derived from motion vectors of triangular meshes and mean histograms of video shots.

Key words: Video databases, content-based access, shot detection, edit effects, content characterization.

1 INTRODUCTION

Organization and management of video content is crucial for video databases storing huge amount of digitized data. The management of digital video requires temporal segmentation of a video sequence into its constituent shots, characterization of the content of shots and specification of similarities between the content of video clips. While most of the current systems employ manual or textual keyword-based methods in the analysis of video content, content-based methods are becoming more popular since they promise more practical and natural solutions.

Fundamental problems encountered in the video content representation are: performing temporal segmentation in an automatic way, selection of the minimum number of key frames that represent the visual content and allow quick browsing, and definition of similarities between video clips in terms of content-based quantitative measures. This paper deals with these fundamental problems and presents a method for the representation of video content. The proposed system, shown in Fig.1, have four processing modules that capture the video content information. The first is the temporal segmentation to identify camera shot boundaries at coarse level. At the second processing step shot boundaries are refined by the localization of edit effects. This is achieved by analysis of luminance variations in temporal regions labeled as edit effects. Next, each temporal segment is abstracted into key-frames based on similarity analysis of mean color histograms. In addition to the color, motion characteristics of the segments are integrated with information about camera operation to specify the shot similarities. Following sections describe each processing module and present experimental results obtained on digitized TV sequences.

*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.

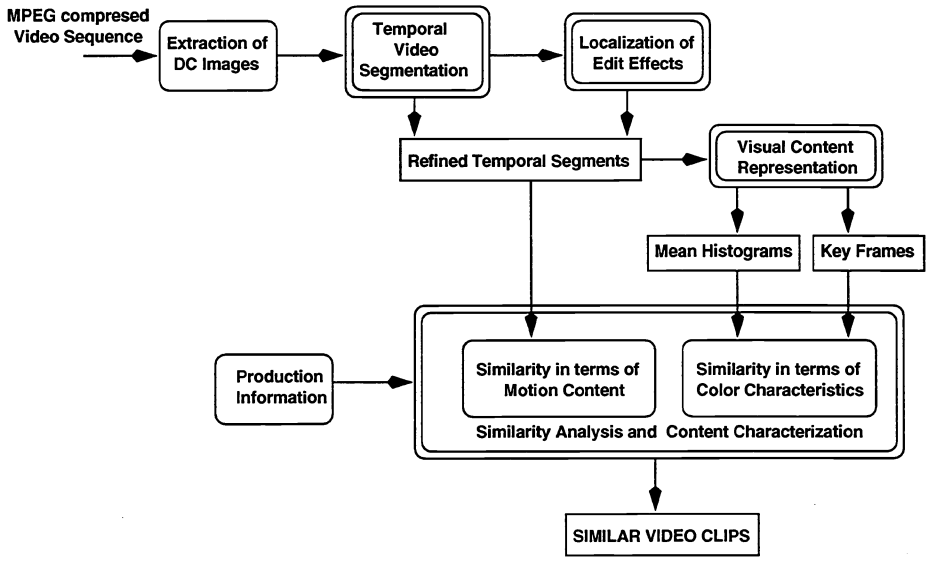

Figure 1: Block diagram of the temporal segmentation and content characterization system.

2 TEMPORAL SEGMENTATION

A video stream is a temporally evolving medium, where content changes occur due to cuts, special effects, and object/camera motion. Temporal video segmentation constitutes the first step in content-based video analysis, and refers to breaking the input video into temporal units based on a certain uniformity criterion.

Early attempts to video content analysis mostly employ "manual partitioning" of video into clips where keywords or text are associated with each segment for indexing. Manual segmentation of video have been found inefficient and inadequate, because it is expensive, and the assigned textual attributes tend to be subjective and vary from one user to another. Therefore, visual content descriptors that facilitate automatic annotation of the input video have been developed to complement textual descriptors. Visual-content-based temporal video segmentation is mostly achieved by detection of "camera shots," where each shot refers to a sequence of frames generated during a single operation of the camera. Shot detection is usually performed by computing frame similarity metrics to distinguish "intrashot" variations (introduced by object/camera movement, and changes in lighting) from "intershot" variations (introduced by "cuts" and "gradual transitions").

A number of temporal video segmentation methods that employ different similarity metrics have been suggested for both uncompressed and compressed video. These methods can be broadly divided into three classes: pixel/block comparison methods, intensity/color histogram comparison methods, and methods using DCT coefficients in MPEG encoded video sequences. A vast majority of reported frame similarity comparison methods employ pixel or histogram-based techniques. The pixel-based methods detect dissimilarities between two video frames by comparing the differences in intensity values of corresponding pixels in the two frames. The number of the pixels changed are counted and a camera break is declared if the percentage of the total number of pixels changed exceeds a certain threshold [1]. These methods produce several false alarms, because camera movement, e.g., pan or zoom, and moving objects have the effect of a large number of pixel changes, and hence a wrong segment will be detected. To eliminate this difficulty, methods perform the similarity comparison on video blocks rather than pixels have been proposed. A likelihood ratio approach which divides the video frames into blocks and then compares the corresponding blocks on the basis of the statistical characteristics of their intensity levels (mean values and variances) is suggested [1], [2]. However, the assumpition of uniform second-order statistics over a block is not always valid and it is possible that two corresponding blocks can have the same statistics even though their content are different. The use of intensity/color histograms for comparison is the more favored approach, since the histogram takes into account the global intensity/color characteristics of each frame, thus it is more robust to noise and object/camera motion. The histogram is obtained from the number of pixels

belonging to each intensity level (or three color intensities) in the video frame. A temporal segment is declared if the dissimilarity between the histograms of two frames is greather than a prespecified threshold [1]. However, one threshold is not adequate to discriminate camera breaks from special camera effects. Selecting a small threshold will lead to false detections due to the object/camera motions. On the other hand, to set the content change detection threshold to a big value results in undetected gradual transitions. Several modifications to the basic scheme of frame-by-frame histogram comparison have been proposed to improve detection performance. Otsuji and Tonomura [3] introduced the "projection detection filter" to detect the highest difference in consecutive frame histogram differences over a period of time. Zhang et al. [1] proposed "twin-comparison" approach which performs an extensive motion analysis to check whether an actual content change has occured or not. The twin-comparison method requires two cutoff thresholds, one higher threshold for detecting abrubt transitions and a lower threshold for gradual transitions. To reduce the computational complexity of twin-comparison, Zhang et al. proposed a "multipass" approach [1]. In this approach, a fast pass is made with a lower threshold to provide potential temporal segments. In the second pass the processing is restricted to the vicinity of potential temporal segment boundaries. Different similarity detection algorithms can be applied in different passes.

A number of temporal segmentation methods have also been proposed in compressed-domain [4, 5, 6]. Compression of the video is carried out by dividing the image into a set of (8x8) pixel blocks. The pixels in the blocks are transformed into 64 coefficients using the discrete cosine transform (DCT), and obtained DCT coefficients are quantized and entropy coded. DCT coefficients are analaysed to find highly dissimilar frames corresponding camera breaks. Motion-based temporal segmentation using the motion vectors in the MPEG compressed data as well as the DCT coefficients has also been experimented [7]. These methods considerably reduce the amount of data to be processed when the video sequence stored in compressed format.

The temporal video segmentation methods described above are applicable to all types of video without any a priori knowledge (domain-independent), and called "syntactic" methods. As it is presented, the fundamental drawback of these syntactic methods is that they do not allow automatic management of the content of input video. Becasue, while the assumption of domain independence is valid for computation of similarity metrics, it clearly does not apply to the decision criteria, i.e., selection of the content change detection threshold(s). Reported studies on the statistical behavior of frame differences clearly show [8, 9] that, a threshold that is appropriate for one type of video data will not yield acceptable results for another type of input. This requires supervision of system designer or user. To this effect, in [10] an "unsupervised" temporal segmentation method which eliminates threshold selection and allows automatic management of uncompressed video data has been proposed. The temporal segmentation method used in this paper extends the method proposed in [10] in such a way that dealing with refinement of segmentation boundaries, minimization of false negatives, and elimination of false positives by applying color histogram similarity analysis on DC images at different levels. The DC components of a video frame can be obtained directly from the MPEG compressed video bit steream or can be calculated using the raw video data, simply taking the mean value of intensities on (8x8) image blocks. To use DC components of compressed data increases the segmentation speed significantly and first proposed in [2].

The proposed temporal segmentation scheme first computes color histogram-based similarity differences for DC components of each video frame pair and than, automatically clusters the similarities into two classes, i.e., "camera shot boundary frame" and "interior shot frame" using 2-classes K-means algorithm. The main advantage of this temporal segmentation method is: it does not require the selection of "scene-change detection" threshold. The performance index used is Euclidean distance, and the initial cluster means for the K-means algorithm are selected arbitrarily. From among the different comparison metrics reported in the literature [11], we have selected the histogram difference:

$$HD_{(k,k+1)} = \sum_{i=0}^{G-1} \left(|H^Y_{k+1}(i) - H^Y_k(i)| + |H^U_{k+1}(i) - H^U_k(i)| + |H^V_{k+1}(i) - H^V_k(i)| \right) \quad (1)$$

Here, $H^Y_k(i), H^U_k(i), H^V_k(i)$ and $H^Y_{k+1}(i), H^U_{k+1}(i), H^V_{k+1}(i)$ denote the Y, U, and V components of the color histogram of the kth and $(k+1)$th frames in the sequence, respectively; i denotes one of the G bins.

If there is a "clean cut" between two video shots, the algorithm labels only one video frame as shot boundary. However, transitions between shots are usually smoothed by edit effects, i.e. fade-in/out and dissolve. In this case the algorithm labels a number of temporarily adjacent/close frames as shot boundary, thus roughly provides the location of edit effects.

3 LOCALIZATION OF EDIT EFFECTS

We refine shot boundaries and localize the edit effects, such as fade-in/out and dissolve with a finer analysis. To this effect, the algorithm first labels the frames between two successive scene changes as "scene change" if the two scene change points are closer than a prespecified frame number T_e. This step roughly provides the placement and the length of edit effects through the sequence. We call a temporal region that begins T_e frames prior and ends T_e frames after from a scene change label cluster as "potential edit effect" region. Next, the distribution of mean luminance value within the potential edit effect regions is analysed. The mean luminance value of frame k, I_{mean_k}, can be defined by the equation:

$$I_{mean_k} = \sum_{j=1}^{M \times N} I_k^Y(j) \tag{2}$$

where $I_k^Y(j)$ is the luminance value at the jth pixel of $M \times N$ video frame k. It is expected that the distribution of mean luminance versus frame number will be in the form of parabole if the edit effect is fade in and out while it will be a half parabole in either of them. For dissolve type of edit effects the ideal distribution will be uniform.

In [12], a method for detecting fade regions is proposed. The method basically computes the mean of the current and previous frames, and the relative mean change between these means. A fade is declared if the relative mean change exceeds a prespecified threshold. Alternatively, a fade is declared if the relative mean change is less than the first predetermined threshold but greater than a second predetermined threshold and the magnitute of the difference between the means is greater than a third predetermined threshold. In [13] and [5] it is proposed to use variance of intensity values as the indication for dissolve and fade in/out in uncompressed and compressed domains, respectively. It is claimed that the variance curve in the dissolve region shows a clear parabolic shape, while it is a half parabole in fade in/out. The only difference between fade and dissolve type of edit effects is, in the fade case the valley of the curve goes down deeper to almost zero. Theoretically this is true but in practice because of noise, object motions and other effects, it is difficult to specify the threshold that discriminates the fade from dissolve. However, our method investigates the distribution of mean luminance value computed for each frame which shows a parabolic shape for fade in-out and remains uniform for dissolve type of edit effects. The computational complexity of the mean value analysis is also lower than the variance analysis.

4 SUMMARIZATION OF VISUAL CONTENT

Detected camera shots are used as input for the visual content representation module that provides a small set of representative frames, usually called key-frames. Such key-frames are displayed as a visual abstraction for browsing of shot content. Furthermore, similarities between shots can be defined by the similarities of their key-frames

The work on key-frame selection is in its infancy. The challenge in the key frame extraction is identifying the frames that maintain the dynamic nature of video content, which is a difficult task. A common way to represent the visual content of a video sequence with still frames is the "storyboard representation," in which time ordered thumbnail images are presented to provide a summary of the content. These representative key frames can be chosen at regular time intervals or one can be selected for each camera shot after temporal segmentation [14, 10]. Similarly, key frames can be presented in a "scene transition graph," which summarizes the visual content in a temporal story structure [8]. Another alternative reported in the literature is "video posters," which depict the visual summary of the video sequence in a graphic layout pattern that is selected according to relative "dominance" of the shots included in each story unit [15].

We propose a key frame selection scheme in which key-frames are extracted at shot level based on color characteristics of the shot. Initially shots between two edit effects are considered as "coarsest story units." The

idea behind this is: edit effects roughly specify the "contextual" transitions, which is a reasonable assumption for video production. Then, within a story unit, the first frame of each shot is always selected as the first key-frame. Following frames in the shot are compared against the "mean color histogram" of the previous frames sequentially as they are processed. The Y, U, and V components of mean color histogram of first k frames for ith bin is defined as:

$$H^Y_{mean_k}(i) = \frac{1}{k}\sum_{f=1}^{k} H^Y_f(i) \quad (3)$$

$$H^U_{mean_k}(i) = \frac{1}{k}\sum_{f=1}^{k} H^U_f(i)$$

$$H^V_{mean_k}(i) = \frac{1}{k}\sum_{f=1}^{k} H^V_f(i)$$

where $i = 1, 2, .., 256$ in the paper.

Next, the scheme computes the histogram difference between the color histogram of current frame and the mean color histogram. This histogram difference can be defined as:

$$MHD_{(k+1)} = \sum_{i=0}^{G-1} (|H^Y_{k+1}(i) - H^Y_{mean_k}(i)| + |H^U_{k+1}(i) - H^U_{mean_k}(i)| + |H^V_{k+1}(i) - H^V_{mean_k}(i)|) \quad (4)$$

where $k+1$ is the current frame. If significant color content change occurs between the current frame and the previous mean histogram, i.e., if $MHD_{(k+1)}$ is greater than a prespecified threshold T_c, the current frame is selected as a new key-frame. The mean histogram computation is reinitialized at each new key-frame. This process will be iterated until the last frame of the shot is reached. It should be pointed out that mean histograms are robust to object motion. The scheme allows us to capture any significant action by a key-frame, while static shots will result in only a single key-frame. The scheme also provides one or more key-frames that represent the content of missed shots, thus minimize the temporal segmentation error coming from the first step. In other words, we perform a hierarchical temporal segmentation that provides the camera shots at the coarse level and produces stationary temporal segments at the finer level. One may argue the similarity between the shot detection methods with multiple threshold and our scheme. The advantage of our algorithm is we provide direct access to camera shots at coarse level and this step does not require the selection of any threshold, however, at the finer level, we aim to give the access to the content of temporal segments that contain any significant action, whether the segment corresponds to a shot or not. Because it is well known that a shot may include different camera motions to capture different actions and an access to individual actions is important for the visual summarization and analysis of video content. Furthermore, unlike the specification of proper shot detection threshold(s), selection of the threshold T_c is not crucial, since it only serves for detection of all significant actions.

5 SHOT SIMILARITY ANALYSIS

Our method for characterizing a shot aims to classify shots into 5 classes based on color and motion characteristics. These classes are: 1) stationary shot, 2) shot including object motions, 3) shot including zoom type camera motion, 4) shot including pan type camera motion, and 5) others. The mean color histograms assigned to each stationary temporal segment are used in the analysis of color characteristics. The number of key-frames that selected to represent the visual content of the shot gives information about degree of the color variations within the shot. Content-based triangular meshes are used to derive motion characteristics of the shot. The production type information, such as length and location of the shot, is also used to support decisions.

The algorithm first checks the number of key frames assigned to each individual shot. If the number of key frames is equal to 1, than the "stationary shot" label is assigned to the shot without any motion analysis. If the number of key frames is greater than 1 but still remains small with respect to the length of the shot, and the motion characteristics display low motion, than the shot is labeled as "shot including object motions." When the

number of key frames is large and motion characteristics of shot is high/smooth, shot is classified as "including camera motion" if a camera motion can be estimated, otherwise "others" label is assigned to the shot.

We use 2-D triangular meshes in motion analysis since they provide [16] all motion characteristics of the video frames as a function of spatial node configurations and node motion trajectories. In [17] we have used 2-D triangular meshes for the representation and indexing of video objects. The content-based mesh-based motion analysis consists of the initial mesh design, node motion estimation, mesh tracking and refinement, and motion ineterpretation steps. We initially design a triangular mesh for the first frame of each temporal segment and track the mesh through the segment. The initial mesh node points are placed on spatial edges in such a way that the node density is higher in spatial high-gradient and rapid motion regions. It is aimed to keep this distribution consistent with the video content through the tracking. Thus, variaciance in the number and localization of the mesh nodes frame to frame allows us to specify motion characteristics of temporal segments as stationary, high or low. The node motion vectors for the results presented in this paper is obtained by hierarchical Lucas&Kanade algorithm.

Our camera motion estimation method is inspired by the Hough transform based video tomography scheme proposed by Akutsu et al. [18]. Unlike Akutsu et al., however, our point correspondences are obtained by tracked mesh node points as opposed to pairwise block matching. We compute a set of camera motion parameters for each temporal segment/shot, since the temporal partitioning method discussed in Section 2 ans 3 will map individual camera motions into different temporal segments. The x and y components of the matching mesh node point pairs (within a particular temporal segment) are transformed to the Hough space separately. The transformation of the x coordinate pairs is modeled by

$$\rho = x_t cos\theta + x_{t-1} sin\theta, \;\; t = 2, ..., VOP_{num} \tag{5}$$

where x_t and x_{t-1} are the x coordinates of a tracked node point at times t and $t-1$, respectively, and (ρ, θ) denotes coordinates of the transformed node in the Hough space. After all matching mesh node point pairs are transformed into the $\rho - \theta$ space through Eqn. 5, the camera parameters can be determined from $(\rho_{max}, \theta_{max})$, which denotes (ρ, θ) pair receiving the maximum number of votes. The zoom and panning parameters, p_{zoom} and p_{pan}, are given by

$$p_{zoom} = tan(\theta_{max}), \quad p_{pan} = F\alpha_x = \frac{\rho_{max}}{cos\theta_{max}} \tag{6}$$

where F is the focal length of the camera, and α_x is angle of the rotation of the camera around the $y-$axis. In the same way, the tilt parameter $F\alpha_y$ can be computed from the peak value of the Hough transformed y-components, where α_y is the rotation of camera around the $x-$axis. Camera motion parameters are added into the content representation as a function of camera focal length. We limit the camera motion analysis with "pan, zoom" and "fix," since these camera motions and their combinations cover most of the camera motions used in video production.

Besides the decision tree classifier employed for shot classification, we aim to order the degree of visual similarities between shots. Since it is desired to find similarities for visual content, one way to follow is to make an analysis only using the key frames specified at the visual representation step [19]. However, in our similarity analysis, rather than performing new computations on key frames/original video frames, we prefer to use the mean color histograms assigned to each temporal segment, since they represent the color content of the shot adequately. To this effect, an $N \times N$ similarity matrix which describes the color similarities is constructed. Here N is the total number of detected camera shots. Let S = $[S_{ij}]$ denotes the $N \times N$ similarity matrix and S_{ij} denotes the similarity between the ith and jth shots. The elements of similarity matrix, S_{ij} values, are derived as:

$$S_{ij} = \sum_{k=1}^{P} \sum_{l=1}^{R} \max s_{kl} \tag{7}$$

where s_{ij} is a similarity metric between two subshots, P and R denote the number of subshots included in the shot i and j, respectively. We have used the histogram differences between the mean color histograms of subshots as a similarity metric, thus maximum similarity corresponds minimum histogram difference.

In [4], clustering of shots based on the similarity measures of visual primitives such as color luminance correlation has been proposed. In [5], 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 [19] key-frames are used in the representation of each video shot and shot similarities has been defined based-on the similarities between the key-frames.

6 RESULTS

Results are obtained in YUV color space on a 32 minutes MPEG1 compressed video data including recorded TV programs, such as sitcom and commercials. First the temporal segmentation method is applied on the DC images obtained from MPEG1 sequence. Next, edit effects are localized by labeling the scene change points closer than T_e =8 frames. The parameter T_e is specified assuming that the shortest camera shot cannot be shorter than 0.5 sec, which gives us the T_e is equal to 8, since the TV sequence consists of 240×180 video frames digitized with a sampling rate 15 frames/sec. In order to label the extracted edit effects as fade in/out or dissolve, the analysis of mean luminance distribution, described in Section 3, is performed within the potential edit effect regions, i.e., 8 frames prior and 8 frames after from each scene change detection cluster. Fig.2(a) through (e) illustrates five frames selected from a fade in-out type edit effect region. The length of the specified potential edit effect region is equal to 30 frames (from 3280 to 3310). The distribution of the mean luminance value versus the frame number within this region is depicted by the solid plot shown in Fig.2(f). The distribution has a parabolic form. For comparison purposes, the distribution of the intensity variances versus frame number is also plotted by dashed line in the same graph. Note that these values are normalized. Table-1 presents the temporal segmentation and edit effect localization results obtained in YUV color space employing color histogram differences. The number of coarse story units obtained is equal to 47. As it is seen, unsupervised K-means clustering performs quite good in the detection of cuts and fade in/out type edit effects, but fails on dissolve type edit effect regions. However, these results are acceptable since missing segments are detected with 100% accuracy at the mean histogram analysis step. The similar performance has been obtained for uncomprressed video while the processing speed for DC images is significantly shorter.

Fig.3(a) depicts the transitions between shots and subshots. Here a 60 frames length sitcom shot consisting of two subshots (SSH-1 and SSH-2) is laying between two other sitcom shots. The shot is divided into two subshots searching the stationary color regions by the mean histogram analysis defined in Section 4. Thus the frames 1282 and 1325 are selected as the visual representative key frames for the shot and the Y, U, and V components of the mean histograms assigned to subshot SSH-1 and SSH-2 are plotted in Fig.3(b) and 2(c), respectively.

We have performed the mean histogram analysis within a 40 frames length camera shot detected by the unsupervised temporal segmentation. Fig.4(a) through (d) illustrate the key frames selected by the mean histogram analysis. Since this shot is labeled as "including camera motion," the last frame of the shot has also been added into the key frames to present a complete visual representation. In this example, the number of detected key frames is high with respect to the total length of the shot thus a mesh-based motion analysis has also been applied for motion characterization. Fig.4(e) through (h) display the initial and tracked meshes through the shot. The number of initial mesh node points (Fig.4(e)) is equal to 219 and, since the shot has high motion components, the number of nodes does not decrease significantly through the shot. The shot is classified as having high motion and high color variations, teherefore, a camera motion analysis based on the method described in Section 5 is performed and the camera motion is estimated as zoom with zooming parameter $p_{zoom} = 0.709$.

Fig.5(a) through (e) illustrate five key frames visually representing the five temporal segments extracted performing temporal segmentation followed by edit effect localization and mean histogram analysis. The first two key frames represent the subshots of a 60 frames length sitcom shot and the rest three frames are the key-frames selected for three stationary temporal segment. All of the five temporal segments are placed in the same coarse story unit and a color mean histogram is computed and assigned to each of them. Table-2 presents the 5×5 matrix that gives the s_{kl} values, i.e., histogram distances between representative mean histograms, for these temporal segments. As it is observed, the similarity values are consistent with the visual similarities between the segments. Note that these values are normalized and the smallest histogram distance specify the most similar segment. Visual similarities between four camera shots are than computed by using Eq.(7).

7 CONCLUSIONS

The paper presents a method to detect camera shot boundaries, specify edit effects, extract visual representation, as well as the color and motion characteristics of shots. It is concluded that the color-based shot characterization method is successful and thus can form a good basis for color-based indexing and retrieval of video clips. Analysis of motion characteristics of shots by tracked 2-D triangular meshes provides a compact representation of motion through the clip- which is an important functionality for video database management. Our current work plan includes integration of the proposed frame-based video content analysis techniques with object-based techniques [17] to desing a video indexing/retrieval system that supports a wide array of object- /frame-based queries.

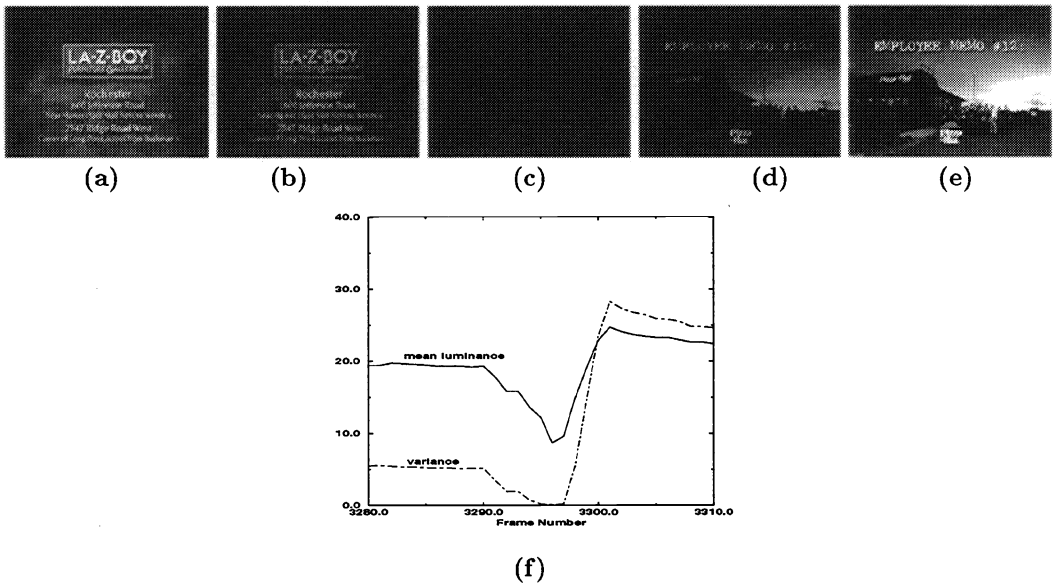

Figure 2: (a), (b), (c), (d), (e) Five frames selected from a fade in-out type edit effect region. (f) The distribution of the mean luminance value (solid line) and intensity variances (dashed line) versus frame number, within the fade in-out effect region.

Table-1 Temporal segmentation and edit effect localization results obtained in YUV space using histogram differences. Original frame size (240x180), sampling rate is 15 frames/sec, necessary CPU time at a SPARC 20 is 7.92×10^{-4} sec/frame for unsupervised temporal segmentation.

	CUTS		DISSOLVES		FADE-IN/OUT		WIPES		OVERALL	
	#	%	#	%	#	%	#	%	#	%
HITS	479	78.3	14	26.4	33	97.1	1	100	527	81.8
MISSES	77	12.3	39	73.6	1	2.9	0	0	117	18.1
FALSE ALARMS	56	9.2	0	0	0	0	0	0	56	9.6

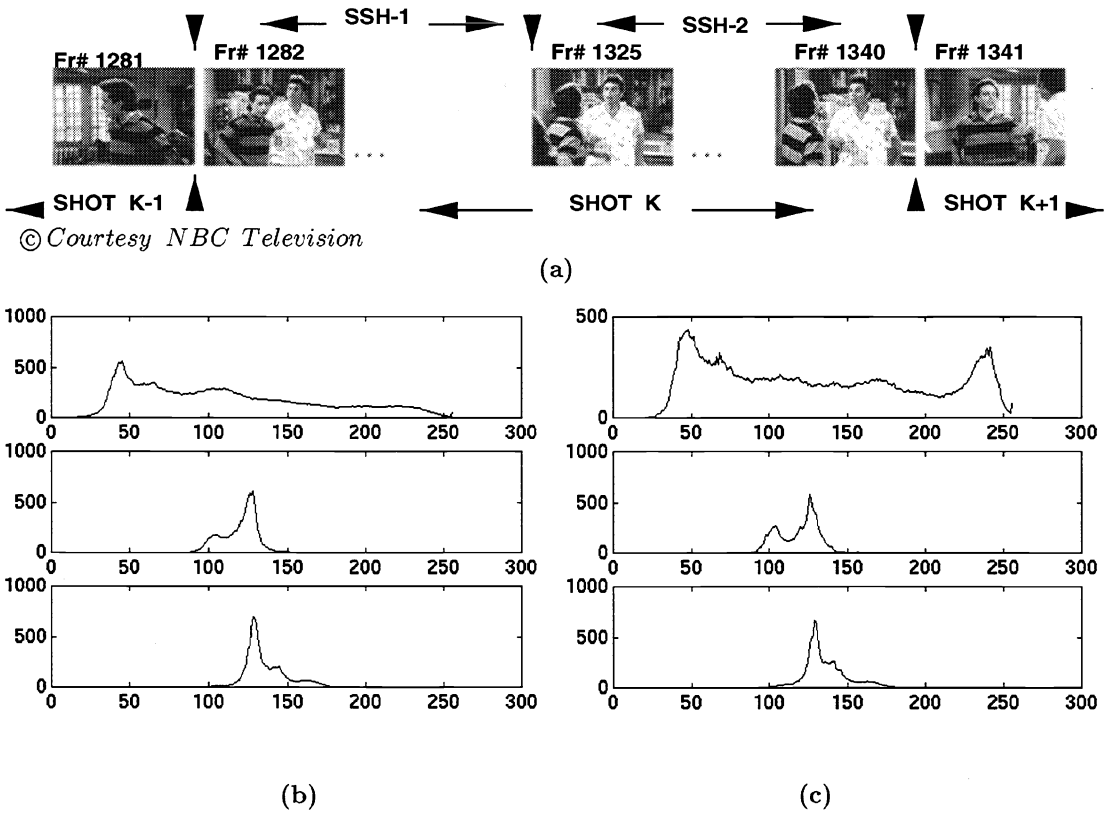

Figure 3: (a) Transitions between shots and subshots. 1282nd and 1325th frames are selected as key frames for the visual representation of a 60 frames length sitcom shot. (b) and (c) depict the Y, U, V components of the mean histograms computed for the first and the second subshots, respectively.

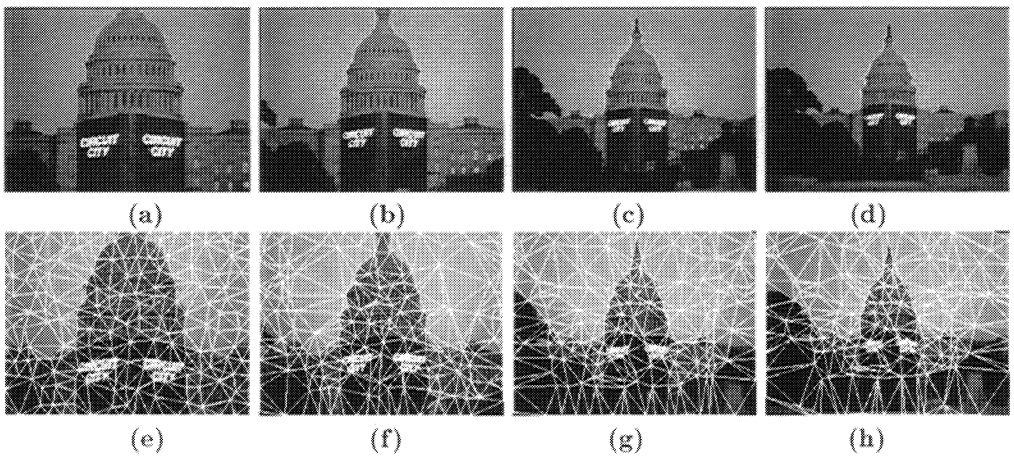

Figure 4: Key frames selected for the visual representation of a shot 40 frames in legth. The shot is labeled as "camera motion shot" since it includes zoom. (a) 14738th frame (firstframe). (b) 14747th frame. (c) 14768th frame. (d) The last frame of the sequence (14776th frame). (e), (f), (g), (h) Tracked meshes corresponding to the key frames shown at (a), (b),(c), and (d), respectively.

©*Courtesy NBC Television*

(a) (b) (c) (d) (e)

Figure 5: (a), (b) Two frames selected for the visual representation of the 60 frame length sitcom shot that resides in 1282-1340. First frames of the "stationary" shots reside in (c) 1341-1359, (d) 1360-1455, and (e) 1456-1547.

Table-2 Normalized similarity values between five temporal segments.

	SSH-1.1	SSH-1.2	S-2	S-3	S-4	S-5.1	S-5.2	S-5.3
SSH-1.1	0.0	1.6966	2.2476	1.4955	2.2620	4.2036	3.9821	3.7389
SSH-1.2	1.6966	0.0	3.1337	0.9394	2.4398	4.5823	4.2645	4.2341
S-2	2.2476	3.1337	0.0	2.7165	1.9428	4.9104	4.8705	4.5479
S-3	1.4955	0.9394	2.7165	0.0	2.1463	4.5247	4.2466	3.9904
S-4	2.2620	2.4398	1.9428	2.1463	0.0	5.2581	5.0500	4.7093
S-5.1	4.2036	4.5823	4.9104	4.5247	5.2581	0.0	2.0099	2.7815
S-5.2	3.9821	4.2645	4.8705	4.2466	5.0500	2.0099	0.0	1.7546
S-5.3	3.7389	4.2341	4.5479	3.9904	4.7093	2.7815	1.7546	0.0

ACKNOWLEDGEMENTS

Authors would like to thank A. M. Ferman who implemented the first version of temporal segmentation program. Special thanks to P.J.L. van Beek for providing the mesh tracking program.

References

[1] H.J. Zhang, A. Kankanhalli, and S.W. Somaliar. Automatic partitioning of full-motion video. *ACM/Springer Multimedia Systems*, 1(1):10–28, 1993.

[2] H. Nagel. Formation of an object concept by analysis systematic time variation in the optically perceptible environment. *Computer Graphics and Image Processing*, 7:149–194, 1978.

[3] K. Otsuji and Y. Tonomura. Projection detection filter for video cut detection. In *Proc. of ACM Multimedia'93*, pages 75–82, Anaheim, CA, 1993.

[4] 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.

[5] J. Meng, Y. Juan, and S. F. Chang. Scene change detection in a MPEG compressed video sequence. In *Proc. SPIE*, volume 2419, pages 14–25, 1995.

[6] 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.

[7] H. J. Zhang, C. Y. Low, Y. Gong, and S. W. Somaliar. Video parsing using compressed data. In *Proc. IS&T/SPIE, Image and Video Processing*, pages 142–149, 1994.

[8] M.M. Yeung and B. Liu. Efficient matching and clustering of video shots. In *Proc. IEEE International Conference on Image Processing*, pages 338–341, 1995.

[9] D. C. Coll and G. K. Choma. Image activity characteristics in broadcast television. *IEEE Trans. on Comm.*, pages 1201–1206, Oct. 1976.

[10] 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.

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

[12] A. M. Alattar. *US Patent.* No: 5245436, 1993.

[13] A.M. Alattar. Detecting and compressing dissolve regions in video sequences with DVI multimedia image compression algorithm. In *Proc. of ACM Multimedia'93*, pages 39–46, 1993.

[14] H. J. Zhang and S. W. Somaliar. Developing power tools for video indexing and retrieval. In *Proc. SPIE Storage and Retrieval for Image and Video Database II*, San Jose, USA, 1994.

[15] M. M. Yeung and B. L. Yeo. Video content characterization and compaction for digital library applications. In *Proc. SPIE: Storage and Retrieval for Image and Video Databases*, pages 45–58, vol. 3022, San Jose, USA, February 1997.

[16] P.J.L. van Beek and A.M. Tekalp. Object-based video coding using forward tracking 2-d mesh layers. In *Proc. SPIE:VCIP*, pages 699–710, vol. 3024, CA, USA, 1997.

[17] B. Günsel, A. M. Tekalp, and P.J.L. van Beek. Object-based video indexing for virtual studio productions. In *Proc. IEEE Int. Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[18] A. Akutsu and Y. Tonomura. Video tomography: An efficient method for camerawork extraction and motion analysis. In *Proc. of ACM Multimedia 94*, pages 349–356, CA, 1994.

[19] 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.