VIDEO SCENE CHANGE DETECTION USING THE GENERALIZED SEQUENCE TRACE

Cüneyt Taşkiran and Edward J. Delp

Video and Image Processing Laboratory *(VIPER)*
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47907-1285

ABSTRACT

We propose an algorithm to detect scene changes in a video sequence in the compressed domain. We define a feature vector extracted from each frame that we call the generalized trace. We examine various ways of processing the generalized trace to determine the temporal location of scene changes in a video stream.

1. INTRODUCTION

Due to the rapid advances in compression technology and imaging hardware, the expansion of low-cost storage media, and the explosion of the Internet, the availability of digital video “for everyone” is now possible. The demand for digital video is also increasing in areas such as video teleconferencing, multimedia authoring systems, education, and video-on-demand systems. Some have even identified the widespread use of digital video and images as the second revolution in communication, the first being the invention of printing.

The technology for organizing and searching images and video based on their content is still in its infancy. This is especially true in multimedia applications where the difficulty of searching and editing data is often the largest cost factor. The first step to extract content-based information from a video sequence is detecting scene changes or shot boundaries. A *shot* is defined as a collection of contiguous frames grouped together that depict a single scene or camera operation or contain a distinct event or action. Once shots are identified, representative key frames may be extracted and the shots may be clustered to obtain hierarchical views of the video.

Recently, detecting shots in the compressed video domain has gained a lot of attention. Patel and Sethi [1] have used a χ^2 technique based on intensity histograms of I frames to detect scene changes. Shen and Sethi [2] have applied the technique which was proposed by Zabih, *et. al.* [3] to the compressed domain by combining it with their compressed domain edge detection algorithm. This method uses the number of entering and exiting edge pixels to find scene changes. Zhang, *et. al.* [4] used the normalized inner product of vectors consisting of predetermined collections of DCT coefficients from a number of preset regions in a frame. They then use a global threshold to detect scene changes. Yeo and Liu [5] detect scene changes by using both pixel differences and luminance histograms based on DC-images extracted from the compressed video stream.

Working directly in the DCT domain has a number of advantages: First, by removing the decoding/coding step and working with a much lower data rate, computational complexity is greatly reduced. Secondly, a great deal of information, such as motion vectors and block averages, which may be of use in detecting shots is available in the data stream. Lastly, manipulation in the compressed domain provides the flexibility to accommodate dynamic resources and heterogenous quality of service requirements, which is particulary important for Internet and video-on-demand applications.

2. THE GENERALIZED TRACE

Given a video stream, V, composed of N frames, $\{f_i\}$, we define the *generalized sequence trace* as follows: Let $\vec{x}_i = [x_{1i}\, x_{2i}\, \cdots\, x_{ni}]^T$ be a feature vector extracted from the pair of frames $\{f_i, f_{i+1}\}$. For this work we have used two features; hence $n = 2$. The generalized trace, d, for V is then defined as

$$d_i = \| \vec{x}_i - \vec{x}_{i+1} \|_2 \qquad (1)$$

The image formed by the DC coefficients of the DCT for a frame in a MPEG sequence is known as the *dc-image*. The sequence of dc-images corresponding to the original video stream is known as the *dc-sequence*. The dc-image can be obtained directly from the MPEG stream for the intracoded I frames, but the DC coefficients are not directly available for the intercoded B and P frames because motion compensation is used for these frames. Motion vectors may be used to estimate the DC coefficients. We have used the method described in [6] to estimate the dc-images for B and P frames.

After the dc-sequence is obtained, the luminance histogram of each dc-image in the sequence is then obtained. The

This work was supported by a grant from the Rockwell Foundation. Address all correspondence to E. J. Delp, ace@ecn.purdue.edu, http://www.ece.purdue.edu/∼ace or +1 765 494 1740.

luminance histogram is a valuable tool in comparing two images and has been used extensively in detecting scene changes [1], [5]. We then define our first feature to be the the dissimilarity measure based on the histogram intersection, H, of the dc-images as

$$x_{1i} = 1 - H(h_i, h_{i+1}) = 1 - \frac{\sum_{j=1}^{K} min(h_i(j), h_{i+1}(j))}{\sum_{j=1}^{K} h_{i+1}(j)} \tag{2}$$

where h_i and h_{i+1} are the luminance histograms for frames f_i and f_{i+1}, respectively, and K is the number of bins used. It was shown in [7] that if the two images that are compared have the same number of pixels, T, that is if

$$\sum_{j=1}^{K} h_i(j) = \sum_{j=1}^{K} h_{i+1}(j) = T \tag{3}$$

then the histogram intersection-based dissimilarity measure is equivalent to the city-block metric. Hence, when comparing dc-images, the first feature may be written as

$$x_{1i} = \frac{1}{2T} \sum_{j=1}^{K} |h_i(j) - h_{i+1}(j)| \tag{4}$$

The second feature that we have used is the absolute value of the difference of standard deviations, for the luminance component of the dc-images, i.e., $x_{2i} = |\sigma_i - \sigma_{i+1}|$ where

$$\sigma_i^2 = \frac{1}{T-1} \sum_i \sum_j (Y_i(i,j) - \mu)^2 \tag{5}$$

and μ is the mean value of the luminance of the dc-image.

These two features were chosen for a number of reasons: First, they are easy to extract. In addition, having both histogram-based and pixel-based techniques is desirable because they complement each other's weaknesses. Pixel-based techniques may give false alarms when there are moving objects and camera movement in the frames. Histogram-based techniques are fairly immune to these effects but they may miss scene changes if the luminance distribution of the frames do not change significantly. Another way to compare the similarity of two frames, which has been used often in the literature, is the χ^2 similarity measure[1]. We have not included it in our feature list because it generally has a high false alarm rate.

The generalized trace for two videoclips are shown in Fig.1 and Fig.2. Here, the trace has been normalized such that the maximum value equals 1. The wide peaks in the generalized trace plot of the "tv2" sequence correspond to scene changes where there is first a fade out to a black frame and then a fade in to another scene. Narrow peaks correspond to cuts. Dissolve scene changes fall somewhere in between, as seen from two dissolves near frames 490 and 600.

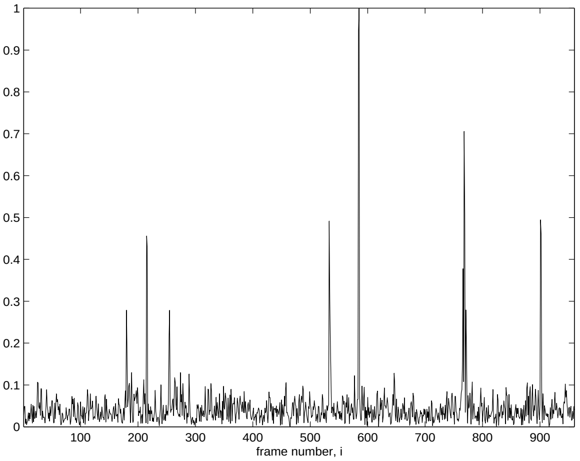

Figure 1: The generalized trace for "tv1"

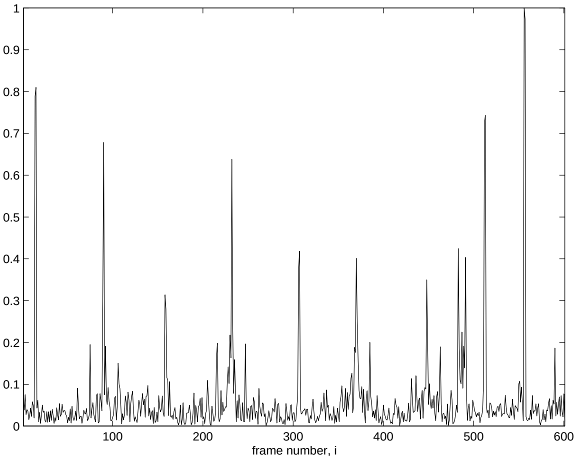

Figure 2: The generalized trace for "tv2"

3. SCENE CHANGE DETECTION

Generally, after a dissimilarity measure is derived from the video sequence, most work use some type of global thresholding technique to detect the scene changes [4],[1]. Simple as this approach may be, *a priori* selection of the threshold is a problem since scene change is a local activity. Considering this fact, others have used sliding windows to process the data and detect shots [5]. We approach the problem differently. Noting that the edges of the generalized trace correspond to scene changes in the video stream, we pose the scene change detection problem as a one dimensional edge detection problem. A number of techniques are available to detect the edges of a 1D signal. We have chosen to use a method based on mathematical morphology because morphological techniques are also useful in later stages where the detected scenes changes are to be classified.

The building blocks of morphological methods are the two operations of erosion and dilation[8]. There are various ways to extend the basic binary erosion and dilation operations to multilevel, i.e. grayscale functions [9]. Based on the umbra representation of signals, we have used the following extension methodology: Let d_i, $i = 0, \ldots, N-2$ be the generalized trace, as defined above, and let s_j, $j = 0, \ldots,$-$L-1$ be another signal, known as the structuring element (SE), with the assumption that $N-1 > L$. The *gray-scale erosion* of d_i by the SE is then defined as

$$(d \ominus s)(i) = \min_{j=0,\ldots,L-1} d(i+j) - s(j) \qquad (6)$$

for $i = 0, \ldots, N-L-1$. Similarly, the *gray-scale dilation* is defined as

$$(d \oplus s)(i) = \max_{j=i-L+1,\ldots,i} d(j) + s(i-j) \qquad (7)$$

for $i = L-1, L, \ldots, N-2$. When the length of the SE, L, is small, the shape of the SE is not important, hence we have chosen a SE having a constant value of 1. Also, we have used a relatively small SE of length $L = 3$ for ease of computation.

Based on these two operations, we define the gradient by dilation as $g^+ = (d \oplus s) - d$ and the gradient by erosion as $g^- = d - (d \ominus s)$. The *morphological laplacian*, $\Lambda(d)$, is then given by

$$\Lambda(d) = g^+ - g^- = ((d \oplus s) - d) - (d - (d \ominus s)) \qquad (8)$$

The morphological laplacian for the "tv2" sequence is shown in Fig.3. It can be shown that $\Lambda(d)$ is an approximation to the second derivative of d [10]. The zero crossings of Λ indicate the location of the edges of the generalized trace, and hence where scene changes occur. In order to isolate the zero crossings due to edges from the ones due to noise, we proceed as follows: Suppose there is a zero crossing between the i'th and $i+1$'th frames, i.e. $\Lambda_i \cdot \Lambda_{i+1} < 0$. If $|\Lambda_i - \Lambda_{i+1}| > t$, where t is a threshold, then we indicate that there is an edge at the i'th frame. One can find the best threshold automatically by varying t and counting the edges detected. A plot which is useful for such a procedure is shown in Fig.4. We see that the best value of t is approximately 0.4. After the edges are detected, we check for spurious edge points based on the fact that edges that are less than D frames apart cannot possibly correspond to scene changes. We have used the value $D = 10$ in this work.

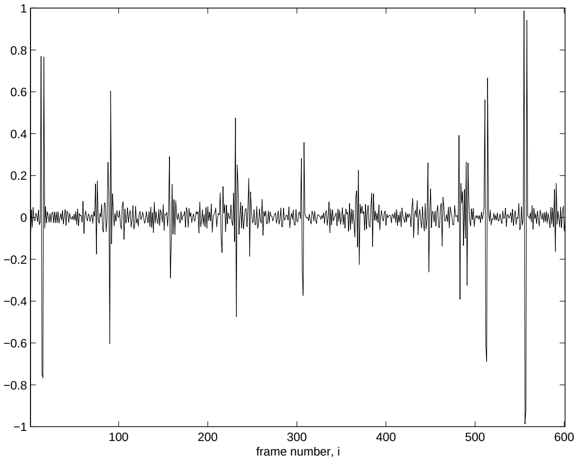

Figure 3: The morphological laplacian for the "tv2" sequence

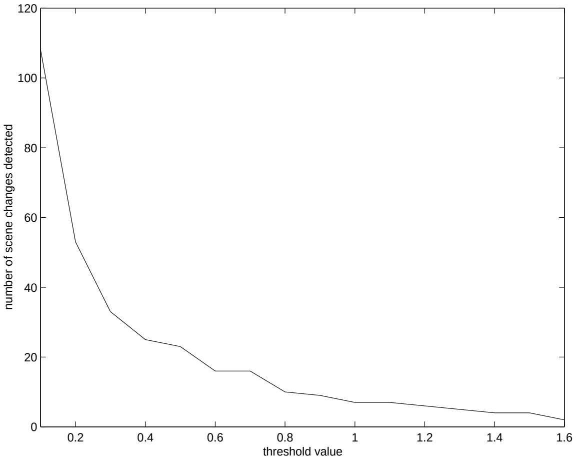

Figure 4: t versus the number of edges detected for "tv2" sequence

4. RESULTS

The results of using the proposed algorithm on the "tv1" and "tv2" sequences are shown in Fig.5 and Fig.6, respectively. All of the scene changes for the "tv1" sequence are detected whereas one is missed in the "tv2" sequence. The missed change was a dissolve type of scene change near frame 600. This was a dissolve from an object to a close up view of the same object so there was little change in the color content and object position. Additional features have to be considered to make the generalized trace more sensitive to such subtle changes.

5. CONCLUSION

We have proposed a new method of detecting scene changes in the compressed domain. After feature vectors are extracted for DC-images, the generalized sequence trace was obtained. We are investigating the use of more features and using the generalized trace for identfying scene content. A postscript version of this paper is available at

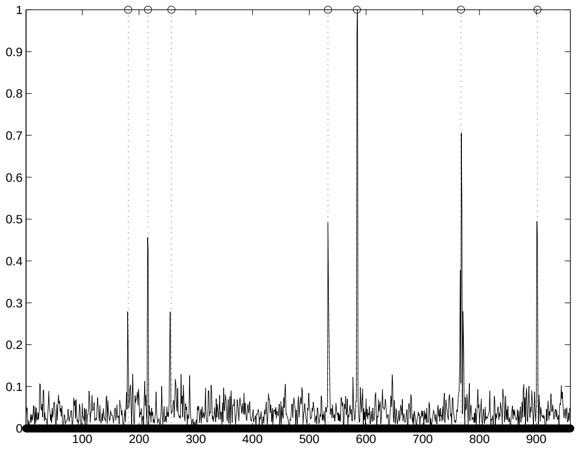

Figure 5: Detected scene changes for “tv1” sequence

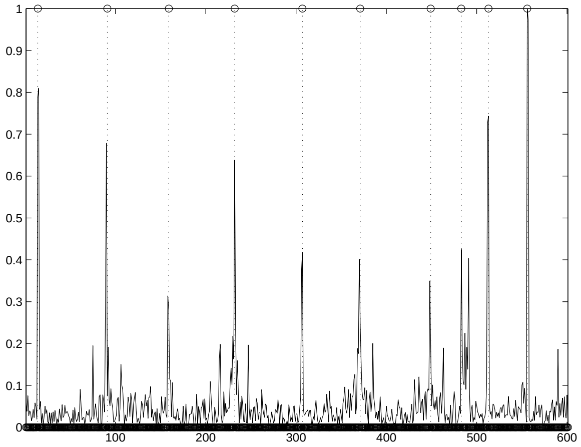

Figure 6: Detected scene changes for “tv2” sequence

`ftp://skynet.ecn.purdue.edu/pub/dist/delp/icassp98-gentrace`.

6. REFERENCES

[1] N. Patel and I. Sethi, “Video shot detection and characterization for video databases,” to appear in *Pattern Recognition*.

[2] B. Shen and I. Sethi, “Cut detection via compressed domain edge detection,” *Proceedings of the IEEE Workshop on Nonlinear Signal and Image Processing*, September 1997.

[3] R. Zabih, J. Miller, and K. Mai, “A feature-based algorithm for detecting and classifying scene breaks,” *Proceedings of the ACM International Conference on Multimedia*, pp.189-200, San Francisco, November 1995

[4] H. Zhang, C. Low, and S. Smoliar, “Video parsing and browsing using compressed data,” *SPIE Conference on Multimedia Tools and Applications* Vol. 1, No. 1, pp. 89-11 1995

[5] B. Yeo and B. Liu, “Rapid scene analysis on compressed video,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 5, No. 6, pp. 533-544, December 1995.

[6] K. Shen and E. J. Delp, “A fast algorithm for video parsing using MPEG compressed sequences,“ *Proceedings of the IEEE International Conference on Image Processing*, Oct. 1995, Washington, D.C., pp.252-255

[7] M. J. Swain and D. H. Ballard, “Color indexing,” *International Journal of Computer Vision*, Vol. 7 No. 1, pp.11-32, 1991

[8] P. Maragos and R. Schafer, “Morphological systems for multidimensional signal processing,” *Proceedings of the the IEEE*, vol. 78, No. 4, pp. 719-739 April 1990

[9] C. Chu and E. J. Delp, “Impulsive noise supression and background normalization of electrocardiogram signals using morphological operators,” *IEEE Transactions on Biomedical Engineering*, vol. 36, No. 2, Feb. 1989

[10] J. Casas and L. Torres, “Strong edge features for image coding,” in *Mathematical Morphology and Its Applications to Image and Signal Processing*, P. Maragos, R. Schafer, and M. Butt (ed.s), Kluwer 1996