A Color-texture Histogram from the HSV Color Space for Video Shot Detection

A. Vadivel
Dept. of CSE
Indian Institute of Technology, Kharagpur
vadi@cc.iitkgp.ernet.in

M. Mohan
Dept. of CSE
Indian Institute of Technology, Kharagpur
mmohan@cse.iitkgp.ernet.in

Shamik Sural
School of IT
Indian Institute of Technology, Kharagpur
shamik@cse.iitkgp.ernet.in

A. K. Majumdar
Dept. of CSE
Indian Institute of Technology, Kharagpur
akmj@cse.iitkgp.ernet.in

Abstract

Color and texture are two important low-level features of a video frame. These features can be used either alone or in combination using appropriate weights for video shot detection. We propose a new soft-decision approach from the HSV color space for modeling combined human visual perception of color and texture in a single feature vector called COLTEX. It shows encouraging results in shot detection for content-based video retrieval applications. We have developed a web-based system for demonstrating our work and for performing user queries.

Keywords: Shot detection, Video retrieval, Color-texture, HSV color space, Soft decision.

1. Introduction

A shot is a continuous sequence of frames captured from the same camera in a video [8,9]. There are different types of shots that include hard cut, dissolve and wipe [1]. Hard cut is an instantaneous transition from one scene to the next [5,17]. Dissolve is a transition from one scene to another in which frames from the first scene gradually merge with the frames from the second scene. Wipe is another common scene break, in which a line moves across the screen, with the new scene appearing behind the line [13,18]. Shot detection is the process of identifying changes in the scene content of a video sequence. This step is fundamental to any kind of video retrieval application since it enables segmentation of a video into its basic components.

There are broadly two approaches to video shot detection, namely, the pixel domain processing and the compressed domain processing [4,12,14]. In the first approach, frame pixels are processed after decoding a compressed video file like MPEG. One such method is pixel comparison, which was introduced to evaluate the differences in intensity or color values of corresponding pixels in two successive frames. The main drawback of this method is that it is sensitive to object and camera movements and noises. Another method is to divide each frame into a number of blocks that are compared against their counterparts in the successive frames. This approach provides a better tolerance to slow and small motions between frames. To further reduce the sensitivity to object and camera movement and thus provide a more robust shot detection technique, color histogram comparison was introduced. Histograms do not depend on the spatial layout of picture but is dependent only on the pixel frequencies [20].

In contrast to the above methods, compressed domain approaches make direct use of the encoded data stored in the MPEG files [7]. Yeo and Liu proposed a method for compressed domain shot detection using a sequence of reduced images extracted from DC coefficients in the Discrete Cosine Transformation (DCT) domain [19]. Another interesting approach was proposed by Lee *et al* [10]. They exploit information from the first few AC coefficients in the transformation domain, and track binary edge maps to segment the video. Macroblock based method also works on compressed MPEG digital video [13]. Entropy-based metrics have recently been used for video shot detection in the compressed domain [3].

With the development of various video shot detection techniques, attempt has been made to compare performance of the different approaches. This includes comparison of automatic shot boundary detection algorithms by Lienhart [11] and by Gargi *et al* [6]. These comparative studies have shown that the pixel domain methods have higher accuracy compared to the compressed domain methods. The compressed domain methods, on the other hand, work faster. Also, of all the pixel domain methods, histogram based techniques perform better than the rest. However, one of the drawbacks of the histogram-based approaches is their sensitivity to the illumination condition of the video. Even in the same shot, variation of the light condition is often detected as a shot change resulting in high false positive rates. In order to handle this situation, we propose a new type of histogram called COLTEX that combines color and texture features in the same vector using soft decision from the HSV color space. While color gives a strong perception of an overall video frame content, texture provides information about further details in each frame.

We explain the method of COLTEX generation in the next section including a short description of the HSV color space properties. In order to demonstrate the effectiveness of our approach and let the readers repeat

our experiments, we have developed a system, which can be accessed freely on the web. Section 3 gives an overview of this video browsing system. Results are given in section 4 and we conclude in the last section of the paper.

2. COLTEX Generation

2.1. Overview of the HSV Color Space

A three dimensional representation of the HSV color space is a hexacone, where the central vertical axis represents intensity, I. Hue, H, is an angle in the range [0,2π] relative to the red axis with red at angle 0, green at 2π/3, blue at 4π/3 and red again at 2π. Saturation, S, is the purity of color and is measured as a radial distance from the central axis with value between 0 at the center to 1 at the outer surface. For S=0, as one moves higher along the intensity axis, one goes from Black to White through various shades of gray. On the other hand, for a given intensity and hue, if the saturation is changed from 0 to 1, the perceived color changes from a shade of gray to the most pure form of the color represented by its hue. When saturation is near 0, all pixels, even with different hues, look alike. As we increase the saturation towards 1, they tend to get separated out and are visually perceived as the true colors represented by their hues.

Using these properties of the HSV color space, it has been shown that a threshold can be used to determine whether the hue or the intensity is the dominant component based on the saturation of a pixel [15]. Subsequently, we proposed the use of soft decision to determine relative importance of "true color" (hue dominance) and "gray color" (intensity dominance) for each pixel. A shot detection method was developed with a color histogram of true color and gray color components. It was shown that the true color contribution $W_H(S,I)$ and the gray color contribution $W_I(S,I)$ can be defined as follows [16].

$$W_H(S,I) = \begin{cases} S^{0.1(255/I)^{0.85}} & \text{for } I \neq 0 \\ 0 & \text{for } I = 0 \end{cases} \quad (1)$$

$$W_I(S,I) = 1 - W_H(S,I) \quad (2)$$

In this paper, we show how the soft decision approach can be effectively used to combine color and texture for video shot detection. The main contribution of this work is that we consider the neighborhood of each pixel and determine the presence of true colors and gray colors in its proximity. Relative position of true colors captures a color feature while relative position of gray colors captures a texture feature. Thus, by the use of true color and gray color distribution around each pixel together, we effectively capture color and texture in the same feature and hence the name "COLTEX". When used for video shot detection, COLTEX shows high recall and precision values.

2.2. Representation of Color and Texture in COLTEX

In order to capture the distribution of true color and gray color pixels in the proximity of any given pixel we consider the following neighborhood relations:

i. Diagonal– Located to the right bottom of the current pixel.
ii. Vertical– Located to the bottom of the current pixel.
iii. Horizontal– Located to the right of the current pixel.

The features for each of the neighborhoods mentioned above is captured using a two dimensional matrix. Here we explain the process of feature extraction for the diagonal neighborhood. The other two features are extracted in a similar manner.

The diagonal neighborhood matrix DIAG is a square matrix of size NxN where $N=N_1+N_2$; N_1 being the number of components representing true color values and N_2 being the number of components representing gray color values. Since true colors have a range of [0,2π], N_1 is defined as

$$N_1 = \frac{2\pi}{Q_H} + 1 \quad (3)$$

Q_H determines the quantization level of hue values. Since gray colors have a range of [0,255], N_2 is defined as

$$N_2 = \frac{255}{Q_I} + 1 \quad (4)$$

Q_I determines the quantization level of intensity values.

We denote the current pixel as p and the neighboring pixel as q. For both p and q, we calculate true color and gray color contributions using Eqs. (1) and (2) to obtain W_H^p, W_I^p, W_H^q and W_I^q.

Since DIAG represents all possible combinations of N_1 true colors and N_2 gray colors, each pixel and its diagonal neighborhood pixel combination (p, q) gets mapped to four cells of DIAG. If we denote the H and I values of p and q as p_H, p_I, q_H and q_I respectively, then the four cells of DIAG affected by p and q are:

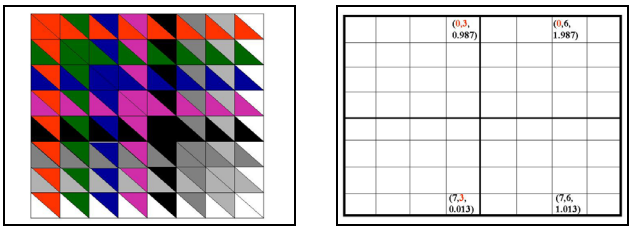
$$\left(\frac{p_H}{Q_H}, \frac{q_H}{Q_H}\right), \left(\frac{p_H}{Q_H}, N_1 + \frac{q_I}{Q_I}\right), \left(N_1 + \frac{p_I}{Q_I}, \frac{q_H}{Q_H}\right) and \left(N_1 + \frac{p_I}{Q_I}, N_1 + \frac{q_I}{Q_I}\right)$$

The quanta of update for these cells are

$$w_H^p + w_H^q, w_H^p + w_I^q, w_I^p + w_H^q \; and \; w_I^p + w_I^q$$

In Figure 1(a) we show the layout of the DIAG matrix. Each cell represents a combination of current pixel color (true or gray) and neighboring pixel color (true or gray). We use 4 components for true colors and 4 components for gray colors. The color of the current pixel is represented in the lower triangle of each cell while the

color of its neighboring pixel is represented in the upper triangle. In table I we show the matrix components that are updated along with the quantum of update for a number of combinations of current pixel and neighborhood pixel HSV values. The corresponding components for the first row are shown in Figure 1(b) to explain the actual process of updating the matrix.



1(a) **1(b)**

Figure 1. (a) Logical View of COLTEX Matrix (b) Relationship between True Color – Gray Color Components and Weights.

Table I. True and Gray Color Contributions with respect to Index Values for (4,4) Combination.

hsv_p	hsv_q	Matrix Components			
		true, true, contr.	true, gray, contr.	gray, true, contr.	gray, gray, contr.
0.4, 0.9,200	1.6π, 0.0,180	0,3, 0.987	0,6, 1.987	7,3, 0.013	7,6, 1.013
1.6π, 0.0,180	2π, 0.5,100	3,3, 0.987	3,5, 0.013	6,3, 1.987	6,5, 1.013
2π, 0.5,100	0.4, 0.9,200	3,0, 1.845	3,7, 0.871	5,0, 1.129	6,5, 0.155

The complete algorithm for updating the three 2-D matrices, namely, Diagonal neighborhood matrix DIAG, Horizontal neighborhood matrix HORZ and Vertical neighborhood matrix VERT is shown in Figure 2.

For each frame, the COLTEX histogram is generated as a linear representation of the three 2-D matrices DIAG, HORZ and VERT. We perform shot detection by comparing the COLTEX vectors of adjacent frames in a video. The comparison is done using a standard distance metric. For any given metric, if the distance exceeds a threshold, a shot change is detected at that frame position.

For row = 1 to IMAGE HEIGHT
For col=1 to IMAGE WIDTH

Read H(row, col), S(row, col), I(row, col)
Read H(row+1, col+1), S(row+1, col+1), I(row+1,col+1)
Read H(row, col+1), S(row, col+1), I(row, col+1)
Read H(row+1, col), S(row+1, co1), I(row+1, col)

Determine w_H *(row, col),* w_I *(row, col)*
Determine w_H *(row+1, col+1),* w_I *(row+1, col+1)*
Determine w_H *(row, col+1),* w_I *(row, col+1)*
Determine w_H *(row+1, col),* w_I *(row+1, col)*

//Update the diagonal Matrix DIAG as follows

$$DIAG\left[\frac{H(row,col)}{Q_H}\right]\left[\frac{H(row+1,col+1)}{Q_H}\right]=$$

$$DIAG\left[\frac{H(row,col)}{Q_H}\right]\left[\frac{H(row+1,col+1)}{Q_H}\right]+$$
$$w_H(row,col)+w_H(row+1,col+1)$$

$$DIAG\left[\frac{H(row,col)}{Q_H}\right]\left[N_1+\frac{I(row+1,col+1)}{Q_I}\right]=$$

$$DIAG\left[\frac{H(row,col)}{Q_H}\right]\left[N_1+\frac{I(row+1,col+1)}{Q_I}\right]+$$
$$w_H(row,col)+w_I(row+1,col+1)$$

$$DIAG\left[N_1+\frac{I(row,col)}{Q_I}\right]\left[\frac{H(row+1,col+1)}{Q_H}\right]=$$

$$DIAG\left[N_1+\frac{I(row,col)}{Q_I}\right]\left[\frac{H(row+1,col+1)}{Q_H}\right]+$$
$$w_I(row,col)+w_H(row+1,col+1)$$

$$DIAG\left[N_1+\frac{I(row,col)}{Q_I}\right]\left[N_1+\frac{I(row+1,col+1)}{Q_I}\right]=$$

$$DIAG\left[N_1+\frac{I(row,col)}{Q_I}\right]\left[N_1+\frac{I(row+1,col+1)}{Q_I}\right]+$$
$$w_I(row,col)+w_I(row+1,col+1)$$

// Update Horizontal matrix HORZ and Vertical matrix VERT similar to diagonal matrix DIAG with $(row+1,col+1)$ *replaced by* $(row,col+1)$ *and* $(row+1,col)$*, respectively.*

Figure 2. COLTEX Generation Algorithm.

3. Web-based Video Retrieval System

In this section we briefly describe some of the important components of a video retrieval system developed by us (http://www.videodb.iitkgp.ernet.in/coltext/index.php).

Video Shot Detection and Key Frame Extraction: For all the available MPEG sequences, shots are detected using COLTEX with a combination of local and global threshold [8]. Key frames are then extracted from the detected shots. The key-frames of all the shots from all the video files form a database of representative images in the video retrieval system.

Query Specification: A query in the system is specified by an example frame. 20 random key frames are initially displayed. The number of key frames to be retrieved and displayed can be chosen as a user input. Users can select any of the displayed frames as the example image by clicking on it.

Display of Result Set: The nearest neighbor result set is retrieved from the video database based on the clicked query frame and displayed as shown in Figure 3. The retrieval process considers the parameter values selected in the options boxes.

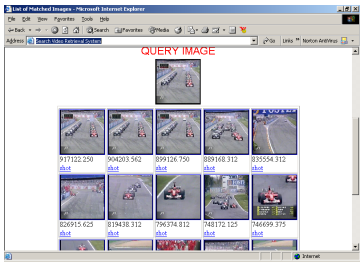


Figure 3. Result set display in the web-based video retrieval system.

Shot Playback: Once the set of similar frames is displayed to the user, the corresponding shot may be played by clicking on a link available at the bottom of each key frame. The shot is played in a custom video player, which opens on a new HTML page as shown in Figure 4. The client requires Java Media Framework to run the player. On this page, we provide buttons to play the next shot, the previous shot or the entire video file along with a play/pause button and a status bar.

External Video Upload: Users are often interested in retrieving video images similar to their own input video frames. To facilitate this, we provide a utility to upload an external video file in our system. We perform on-line shot detection and key frame extraction from the uploaded video. The key frames are then displayed as possible example images. *To the best of our knowledge, this feature is unique in our system and is not available in any other video retrieval system available in the public domain.*

Figure 4. Custom media player running a retrieved shot.

4. Results

The performance of any feature-based shot detection technique is dependent on the distance metric used. We first compare the results of four distance metrics, namely, Bin-to-bin distance (BTB), Chi-square test histogram difference (CHI), Histogram Intersection (HI) distance and the Vector cosine angle distance (VCAD). The four frame distance measures (fd) between two frames with N pixels having histogram representations h_1 and h_2 are defined as follows:

$$fd_{BTB}(h_1,h_2) = \frac{1}{N}\sum_i |h_1[i]-h_2[i]| \quad (5)$$

$$fd_{CHI}(h_1,h_2) = \frac{1}{N^2}\sum_i \frac{|h_1[i]-h_2[i]|}{h_2[i]} \text{ for } h_2[i]!=0 \quad (6a)$$

$$= \frac{1}{N^2}\sum_i \frac{|h_1[i]-h_2[i]|}{h_1[i]} \text{ for } h_2[i]=0 \quad (6b)$$

$$fd_{HI}(h_1,h_2) = 1 - \frac{\sum_i \min(h_1[i],h_2[i])}{N} \quad (7)$$

$$fd_{VCAD}(h_1,h_2) = \frac{h_1 \bullet h_2}{\|h_1\|\|h_2\|} \quad (8)$$

$h_1 \bullet h_2$ represents dot product of h_1 and h_2 while $\|h_1\|$ represents norm of the histogram h_1.

In Figure 5, we show the variation in precision of shot detection for a number of recall values. Here Recall and Precision are defined as:

$$\text{Recall} = \frac{NoOfTrueShotsDetected}{TotalNoOfTrueShots} \quad (9a)$$

$$\text{Precision} = \frac{NoOfTrueShotsDetected}{TotalNoOfShotsDetected} \quad (9b)$$

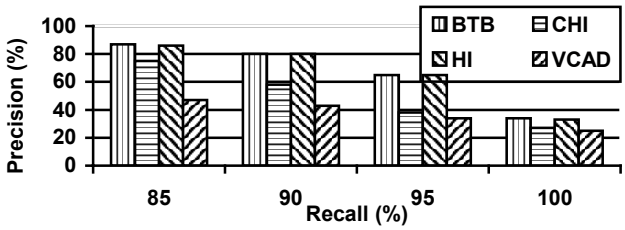


Figure 5. Precision of COLTEX based shot detection for different frame-to-frame distance measures.

The test data set consists of 20 MPEG videos of different categories like movie, soccer, advertisement, cartoon, car racing and news as shown in Table II. The video sequences contain all the three types of shots, namely, hard cut, dissolve and wipe.

Table II. Details of the Test Data.

Type of Video	No. of Video Clips	Total No. of Frames	Total No. of Shots
Advertisement	3	5670	238
Movie	11	30340	521
Cartoon	2	7770	57
Soccer Game	1	2460	27
Formula Racing	1	3715	37
News	1	2665	57
Documentary	1	2910	57
Total	20	58530	994

It is seen from Figure 5, that the histogram intersection based distance and the bin-to-bin distance have similar retrieval performance and are far higher than CHI and VCAD. In the subsequent experiments we use HI as the distance measure with COLTEX.

We next perform accuracy comparison with a number of other standard shot detection methods. This includes a standard color-based detection [8], a motion vector based detection [2], combination of color and motion, and a recently proposed method using temporal slices analysis [12]. For the temporal slices analysis approach, we consider the L1 norm, which was shown to have higher precision by its authors. The results are plotted in Figure 6. It is seen that performance of standard color histogram is better than motion-based detection. A combination of color and motion improves the accuracy which is almost the same as the temporal slices approach. However, the performance of the COLTEX histogram is much better than all these approaches. COLTEX histogram shows a precision of about 85% even when the recall is 85%. All the other approaches achieve a precision of only 50% at such recall values.

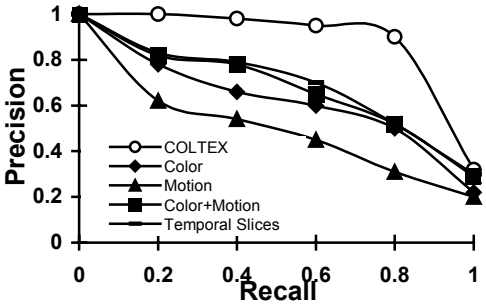


Figure 6. Comparison of Recall Vs. Precision of COLTEX with other standard techniques.

Besides the accuracy of shot detection, we compare the performance of the COLTEX histogram with the other approaches in terms of the processing efficiency. In Table III, we show the average time of feature extraction on a Pentium IV 1.8 GHz computer running Linux.

Table III. Comparison of feature extraction time.

Shot Detection Method	Per Frame Feature Extraction Time (s)
Color	0.794
Motion	0.623
Color + Motion	0.820
Temporal Slices Analysis	0.878
COLTEX	0.814

From the above table, it is seen that, the COLTEX feature extraction time for each frame is comparable with the combined color and motion approach and is less than temporal slices analysis technique. Color histogram approach and the motion vector approach take slightly less time. Overall, the COLTEX histogram shows an encouraging speed-accuracy performance.

5. Discussions and Conclusions

Video data management and information retrieval is an exciting and challenging research area. We have proposed a shot-change detection method using COLTEX - a novel color-texture histogram generated from the HSV color space with the help of soft decision. Since the COLTEX histogram has a better shot detection performance, we are extending our research to the compressed domain. We have developed a web based video shot detection and retrieval system that is available free for use. One of the key features of this application is the ability of the users to load their own video clippings, which will be processed by our system for on-line shot boundary detection and subsequent retrieval of images similar to the key frames. The application has been developed using a modular approach so that if other shot boundary detection

algorithms are implemented, we can seamlessly integrate them in the existing system.

In a recently proposed approach to video shot detection, very high recall and precision has been reported using a compressed domain approach [1]. We would like to implement this method and test on the same data set to make an objective comparison with our approach. In order to make our web-based application even more useful, we are in the process of adding a larger number of new MPEG files to have at least a few thousand minutes of video available for retrieval.

Acknowledgement

The work done by Shamik Sural is supported by research grants from the Department of Science and Technology, India, under Grant No. SR/FTP/ETA-20/2003 and by a grant from IIT Kharagpur under ISIRD scheme No. IIT/SRIC/ISIRD/2002-2003.

References

[1] J. Bescos. Real-time Shot Change Detection over online MPEG-2 Video. *IEEE Transactions on Circuits and Systems for Video Technology*, CSVT-14, pages 475-483, 2004.

[2] E. Bruno and D. Pellerin. Video Shot Detection based on Temporal Linear Prediction of Motion. In *Proc. IEEE International Conference on Multimedia and Exposition*, Lausanne, Switzerland, pages 289-292, 2002.

[3] Z. Cernekova, C. Nikou and I. Pitas. Shot Detection in Video Sequences using Entropy-based Metrics. In *Proc. IEEE International Conference on Image Processing*, Rochester, NY, pages 421-424, 2002.

[4] J. Fan, A. K. Elmagarmid, X. Zhu, G. A. Walidand L. Wu. Classview: Hierarchical Video Shot Classification, Indexing and Accessing, *IEEE Transactions on Multimedia*, 6, pages 70-86, 2004.

[5] W. A. C. Fernando, C. N. Canagarajah and D. R. Bull. Fade and Dissolve Detection in Compressed Video Sequences. In *Proc. IEEE International Conference on Image Processing*, pages 299-303, 1999.

[6] U. Gargi, R. Kasturi and S. H. Strayer. Performance Characterization of Video Shot-change Detection Methods. *IEEE Transactions on Circuits and Systems for Video Technology*, CSVT-10(1), pages 1-13, 2000.

[7] A. Hanjalic. Shot-boundary Detection: Unraveled and Resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, CSVT-12, pages 90-104, 2002.

[8] I. Koprinska and S. Carrato. Temporal Video Segmentation, a Survey. *Signal Processing, Image Communication*-16, pages 450-477, 2001.

[9] T.C.T. Kuo and A.L.P. Chen. Content based Query Processing for Video Databases. *IEEE Transactions on Multimedia* 2(1), pages 240-254, 2000.

[10] S.-W. Lee, Y.-M. Kim and S.-W Choi. Fast Scene Change Detection using Direct Feature Extraction from MPEG Compressed Videos. *IEEE Transactions on Multimedia*, 2(4), pages 240-254, 2000.

[11] R. Lienhart. Comparison of Automatic Shot Boundary Detection Algorithms. In *Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases*, volume 3656, pages 290-301, 1998.

[12] C.-W. Ngo, T.-C. Pong and H.-J. Zhang. On Clustering and Retrieval of Video Shots through Temporal Slices Analysis, *IEEE Transactions on Multimedia*, 4(4), pages 446- 458, 2002.

[13] S.-C. Pei and Y.-Z. Chou. Effective Wipe Detection in MPEG Compressed Video using Macroblock type Information. *IEEE Transactions on Multimedia*, 4(3), pages 309-319, 2002.

[14] E. Sahouria and A. Zakhor. Content Analysis of Video using Principal Components. *IEEE Transactions on Circuits and Systems for Video Technology*, CSVT-9, pages 1290-1298, 1999.

[15] S. Sural, G. Qian and S. Pramanik. Segmentation and Histogram Generation using the HSV Color Space for Content-based Image Retrieval. In *Proc. IEEE International Conference on Image Processing*, pages 589-592, 2002.

[16] S. Sural, M.Mohan and A.K. Majumdar. A Soft Decision Histogram from the HSV Color Space for Video Shot Detection. In S. Deb, editor, *Video Data Management and Information Retrieval*, Idea Group Publishing, Hershey, PA, USA, 2004 (to appear).

[17] B. T. Truong, C. Dorai and S. Venkatesh. New Enhancements to Cut, Fade and Dissolve Detection Processes in Video Segmentation. In Proc. *ACM Multimedia*, pages 219-227, 2000.

[18] J. Wei, M.S. Drew and Z.N. Li. Video Dissolve and Wipe Detection via Spatio-temporal Images of Chromatic Histogram Differences. In *Proc. IEEE International Conference on Image Processing*, pages 929-932, 2000.

[19] B.-L.Yeo and B. Liu. Rapid Scene Analysis on Compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, CSVT-5, pages 533-544, 1995.

[20] R. Zabih, J. Miller and K. Mai. A Feature-based Algorithm for Detecting Cuts and Classifying Scene Breaks. In *Proc. ACM Multimedia*, pages 189-200, 1995.