Content-based Search of Video Using Color, Texture, and Motion

Yining Deng and B. S. Manjunath
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560
deng@iplab.ece.ucsb.edu, manj@ece.ucsb.edu

Abstract

We present an implementation of a system for content-based search and retrieval of video based on low-level visual features. Currently the system consists of three parts, automatic video partition, feature extraction, video search and retrieval. Three primary features, color, texture and motion are used for indexing. They are represented by color histogram, Gabor texture features, and motion histogram. Most of the processing is done directly in the MPEG compressed domain. Testing on sports and movie databases have shown good retrieval performance.

1 Introduction

The rapid development in information technology has enabled users to browse large amount of multimedia data over internet. However, tools for facilitating search and retrieval of these data, especially video data, are still limited.

Recently, there have been several video content-based retrieval systems being developed. In [1] video retrieval is based on multiple low-level global features. Videobook [3] characterizes motion, texture and color using each feature's mutual information. In [2] video indexing is done in the compressed domain using DCT coefficients and by counting numbers of different types of macroblocks used in MPEG. A clustering method is used in [6] for video browsing and annotation. In [15] a system is built for annotation of basketball video. A video engine is also reported in [9].

The results presented in this paper concerns mostly with the temporal partitioning and the use of color, texture, and motion for video search. Our system differs from others reported in the literature in one or more of the following aspects:

- A simple and novel approach to index motion information is presented. The proposed motion histogram is a variation of the well known color histogram technique adopted to motion features.
- Many of the current systems use a few key frames to represent each partitioned video segment. While key framing is a simple and efficient way for characterizing color and texture, it is not sufficient to describe motion information of the entire scene. Further, it also suffers from the problem of how to choose good key frames. In our approach, Every I frame (block DCT coded frame) of the MPEG data is used to extract color and texture features. This is similar to the key-frame approach. But we also use all the P and B frames (motion prediction coded frames) of the MPEG data to extract motion information. The entire video shot is used as the query instead of key frames.
- A Gabor texture feature set which is shown in [11] to have a better performance than other texture features is used.
- Most of the video processing is done in the MPEG compressed domain (see [10] for a complete review of MPEG). The need for storage, retrieval, and network distribution of large amount of video data makes it more practical to process them directly in compressed domain and obtain as much information as possible without or with least amount of decoding.

A demonstration version of the system which illustrates similarity retrieval is available on the web at *http://copland.ece.ucsb.edu/Demo/video/*.

In the next section, we give a brief review of our video partitioning algorithm. Section 3 discusses the three low-level features used in our system. Section 4 shows some experimental results.

2 Automatic Video Partitioning

Video partitioning is the process of segmenting a long video clip in temporal domain into small video shots each of which contain consistent visual information. Often each video shot is a natural scene and partitioning points are camera breaks or editing cuts. While breaks are easy to detect, exact starting and ending points of gradual transitions are more difficult to locate. Here we use an approach similar to [7] for automatic video partitioning in compressed domain. The algorithm consists of three steps in order to detect both abrupt scene cuts and gradual transitions.

0-8186-8183-7/97 $10.00 © 1997 IEEE

Step 1: Camera breaks are detected using macroblock information of P and B frames in MPEG. Motion discontinuity will occur if there is any sudden change between two consecutive frames. This results in a significant drop of forward motion prediction coded macroblocks and can be easily detected by setting a threshold.

Step 2: Gradual transitions such as dissolves, wipes, fade-in, fade-out, and other editing effects are detected by comparing color histogram and intensity differences between consecutive I frames. Normally the interval between two I frames is around 10 - 15. Transitional effects will result in large changes in either color histogram or intensity values. Denote the distance of color histogram between two images i and j as $dH_{i,j}$ (definition will be given in the next section). Define the distance of intensity (luminance component of YC_rC_b color space) between two images as

$$dI_{i,j} = \sum_{x,y} |I_i(x,y) - I_j(x,y)| \qquad (1)$$

where $I(x,y)$ is the luminance (Y component) image. Let V_{dI} and V_{dH} be the variance of $dI_{i,i-1}$ and $dH_{i,i-1}$, respectively, of the entire sequence. We use the following two criteria to determine whether there is any scene change.

1. $|dI_{i,i-1} - dI_{i+1,i}| > wV_{dI}$ & $|dH_{i,i-1} - dH_{i+1,i}| > V_{dH}$
2. $|dH_{i,i-1} - dH_{i+1,i}| > wV_{dH}$ & $|dI_{i,i-1} - dI_{i+1,i}| > V_{dI}$

where w is a ratio factor chosen to be 2 in the experiments. Satisfaction of either condition indicates a scene change and this includes camera breaks as well as gradual transitions. The purpose of using differences of distances is to keep detection localized so that it does not create false positives in a busy scene or miss out a small scene change in an idle scene. If $dI_{i,i-1} > dI_{i+1,i}$, scene change should be between frame $i-1$ and i. Else, it is between frame i and $i+1$. Same argument holds of dH.

Step 3: Results in step 1 and 2 are combined. In step 2, scene changes are detected but exact locations for partitioning can not be obtained. We reapply the method in step 1 by looking at P and B frames between the detected two I frames. This time two consecutive frames with maximum change is picked instead of using the threshold. Step 2 will miss out in special situations where neither intensity nor color histogram changes much during camera breaks. Thus, we use the results from step 1 for sudden scene cuts and the results from step 2 for gradual transitions.

For processing in the MPEG domain, we use the DC images proposed in [12] in step 2 for color histogram and intensity calculations. The DC images of I frames can be easily obtained by the DC values of the DCT coefficients.

3 Feature Representation

As mentioned earlier, three global visual features, color histogram, Gabor texture features, and motion histogram are used to represent each partitioned video segment. The first two features are extracted from every I frames of the video shot and averaged over the entire shot. Motion features are extracted from all the P and B frames of the video shots. Color and motion features are described in the following, and for texture feature details we refer to [11].

3.1 Color Histogram

Color histogram [8] has been widely used as a color feature for image retrieval [4][13][14]. We have adopted the quadratic distance method in [5] in our system. Generalized Lloyd algorithm (GLA) is used to quantize the RGB color space to N number of bins. $N=256$ is used in our experiments. The entire database is used to train the codebook. Each bin contains colors of similar values and is represented by its centroid $c_i = (r_i, g_i, b_i)$, $i = 0,..., N-1$.

For each pixel in the image, its color is classified into one of the N bins and the number of colors in each frame for each bin is counted. These numbers are summed up for every I frame of the shot and normalized. This forms an N-dimensional color feature vector $P = [p_0\ p_1 ...\ p_{N-1}]^T$, $\sum_i p_i = 1$.

The quadratic distance measure for two color feature vectors P and Q is defined as:

$$D(P,Q) = [(P-Q)^T A (P-Q)]^{1/2} \qquad (2)$$

where A is an NxN matrix describing the similarities between any two bins. Elements of A, a_{ij} are chosen such that the more similar the two colors c_i and c_j, the larger the a_{ij}. A simple Euclidean distance is used to measure the similarity between two colors:

$$d_{ij} = \|c_i - c_j\| \qquad (3)$$

Let $d_{max} = max_{i,j}(d_{ij})$, then

$$a_{ij} = 1 - d_{ij}/d_{max} \qquad (4)$$

3.2 Motion Histogram

A novel approach to motion representation by extending the color histogram method is presented in the following. The codebook design, feature vector extraction and distance measure methods are similar to the color histogram discussed in section 3.1 except that they are calculated for 2D motion vectors instead of 3D colors. Preliminary experiments show that this method is effective in classification of motion information.

The method does not assume any specific model of objects or any specific type of scenes. It is not very sensitive to the inaccuracy of motion estimation for a particular frame because the histogram is extracted from the average statistics of the entire shot. A 2D motion estimation model is used because 3D motion estimation can be unreliable for complicated scenes where the background is also moving. This method also avoids separating camera motion from object motion for the same reason mentioned before. Instead, it integrates the object, background, and camera motions all together and classifies scenes.

As a fast implementation in MPEG, we classify motion vectors of each macroblock (MB) and count the number of each bin in the codebook to form a normalized feature vector. I frames are skipped because they are intra-coded and no motion information is available. P frames can have forward motion prediction and B frames can have both forward and backward motion prediction. MPEG defines motion vector (MV) as the displacement from the Target (current frame) MB to the Prediction (reference frame) MB. This is normalized by the distance between the two frames. Macroblocks for which motion vectors are unknown, such as the intra-coded macroblocks, are discarded. Also bin 0 of the codebook is reserved for 0 motion vector only.

3.3 Feature Matching

Let dC_{ij}, dT_{ij}, dM_{ij} denote the distances between two video shots i and j of color, texture and motion. These distances are normalized by the mean distance values for each feature. The overall distance measure can be obtained by

$$dO_{ij} = w_C dC_{ij} + w_T DT_{ij} + w_M dM_{ij} \qquad (5)$$

where w_C, w_T, and w_M are user specified weights for each feature. The best match for the query is the one with the smallest overall distance.

4 Experimental Results

We tested our system on movie and football game video databases. The football game video is chosen for its well defined shots, dynamic motion scenes, and complicated editing effects. MPEG data are at 29.97 Hz, 2 Mbps, and 352x240 resolution. Figure 1 shows a retrieval example. It is a snapshot from the demonstration program that is available on the web.

The 72 minute football video is partitioned into about 1100 segments. Figure 1(a) shows a query sequence consisting of 290 frames. Figure 1(b) shows the user interface and the retrievals. In this example, the three features have equal weights for similarity matching. The top five retrievals shown all have similar plays.

The limitations of global low-level visual features are apparent as they can only provide very preliminary video classification. Our current research is focused on spatio-temporal segmentation for localized object feature representation. For example, identifying individual players in a football sequence will facilitate queries such as "show all passes that resulted in a touch down". We are also developing algorithms for interactive learning to improve search results.

Acknowledgments: This research was supported in part by grants from NSF IRI-94-11330 and NASA NAGW #3951.

5 REFERENCES

[1] E. Ardizzone, M.L. Cascia, "Multifeature image and video content-based storage and retrieval", *Proc. of SPIE*, vol.2916, pp265-276, 1996.

[2] V. Kobla, D. Doermann, and K. Lin, "Archiving, indexing, and retrieval of video in the compressed domain", *Proc. of SPIE*, vol.2916, pp78-89, 1996.

[3] G. Iyengar and A.B.Lippman, "Videobook: an experiment in characterization of Video", *Proc. of IEEE Intl. Conf. on Image Processing*, vol.3, pp 855-858, 1996.

[4] W.Y. Ma, Y. Deng, and B.S. Manjunath, "Tools for texture / color based search of images," *Proc. of SPIE*, vol. 3106, 1997.

[5] J. Hafner et.al., "Efficient color histogram indexing for quadratic form distance functions", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no.7, July 1995.

[6] D. Zhong, H.J. Zhang and S-F. Chang, "Clustering methods for video browsing and annotation", *Proc. SPIE*, vol.2670, pp239-40, 1996.

[7] Q. Wei, H. Zhang, Y. Zhong, "Robust approach to video segmenation using compressed data", *Proc. of SPIE*, vol.3022, pp448-457, 1997.

[8] M.J. Swain and D.H. Ballard, "Color indexing", *Int. Journ. of Computer Vision*, vol.7, no.1, pp11-32, 1991.

[9] A. Hampapur, et. al., "Virage video engine", *Proc. of SPIE*, vol. 3022, pp 188-200, 1997.

[10] B. G. Haskell, A. Puri and A.N. Netravali, *Digital Video: an Introduction to MPEG-2*, Chapman & Holl, 1997.

[11] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp 837-842, Aug. 1996.

[12] B. L. Yeo and B. Liu, "On the extraction of DC sequence from MPEG compressed video", *Proc. of IEEE Int. Conf. on Image Processing*, vol.2, pp 260-263, 1995.

[13] J. R. Smith and S. F. Chang, "Local color and texture extraction and spatial query", *Proc. of IEEE Int. Conf. on Image Processing*, vol.3, pp 1011-1014, 1996.

[14] H. Zhang, Y. Gong, C.Y. Low, S.W. Smoliar, "Image retrieval based on color fetures: an evaluation study", *Proc. of SPIE*, vol.2606, pp212-220, 1995.

[15] D.D. Saur, Y.-P. Tan, S.R. Kulkarni, P.J. Ramadge, "Automated analysis and annotation of basketball video", *Proc. of SPIE*, vol.3022, pp176-187, 1997.

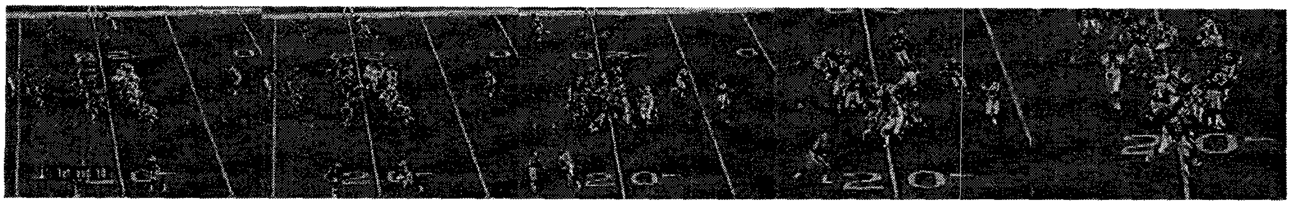
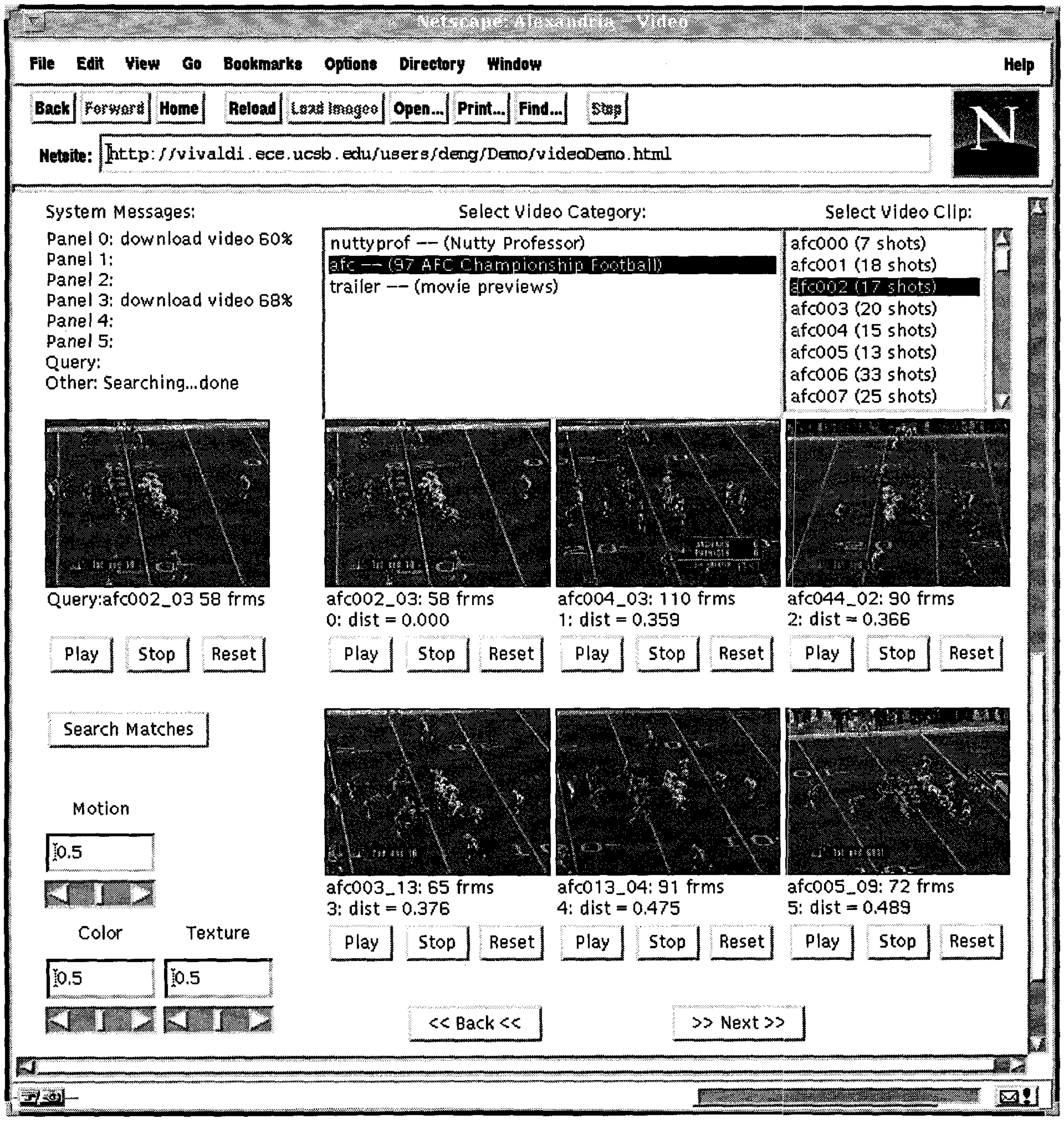


Figure 1(a). Sample frames of the query video shot. Interval between two frames in the filmstrip is 30 frames. This is a typical football game scene. Features present include green color, grass texture, random motion of small objects (players) as well camera panning and zooming.



Netsite: http://vivaldi.ece.ucsb.edu/users/deng/Demo/videoDemo.html

System Messages:
Panel 0: download video 60%
Panel 1:
Panel 2:
Panel 3: download video 68%
Panel 4:
Panel 5:
Query:
Other: Searching...done

Select Video Category:
nuttyprof -- (Nutty Professor)
afc -- (97 AFC Championship Football)
trailer -- (movie previews)

Select Video Clip:
afc000 (7 shots)
afc001 (18 shots)
afc002 (17 shots)
afc003 (20 shots)
afc004 (15 shots)
afc005 (13 shots)
afc006 (33 shots)
afc007 (25 shots)

Query:afc002_03 58 frms

afc002_03: 58 frms
0: dist = 0.000

afc004_03: 110 frms
1: dist = 0.359

afc044_02: 90 frms
2: dist = 0.366

afc003_13: 65 frms
3: dist = 0.376

afc013_04: 91 frms
4: dist = 0.475

afc005_09: 72 frms
5: dist = 0.489

Motion
0.5

Color
0.5

Texture
0.5

Figure 1(b). A retrieval example with user interface.