A Fast Content-Based Multimedia Retrieval Technique Using Compressed Data

Borko Furht and Pornvit Saksobhavivat
NSF Multimedia Laboratory
Florida Atlantic University, Boca Raton, Florida 33431

ABSTRACT

In this paper, we present a novel technique that can be used for fast similarity-based indexing and retrieval of both image and video databases in distributed environments. We assume that image or video databases are stored in the compressed form using standard techniques such as JPEG for images, and M-JPEG or MPEG for videos. The existing techniques, proposed in the literature, use computationally intensive features and cost functions for content-based image and video retrieval and indexing. The proposed algorithm uses an innovative approach based on histograms of DC coefficients only, and therefore is computationally less expensive than the other approaches.

In the case of a JPEG-compressed image database, the query process is the following. The user submits a request for search-by-similarity by presenting the desired image. The algorithm calculates the DC coefficients of this image and creates the histogram of DC coefficients. Then, the algorithm compares the DC histogram of the submitted image with the DC histograms of the images stored in the database using a histogram similarity metric. The image database can be local or at a remote server. In our experiments, we compared several histogram similarity metrics: weighted Euclidean distance, square difference, and absolute difference. The algorithm then selects and presents to the user the images with the smallest values of the metric that best match the submitted image.

In the case of a compressed video database, the similarity-based indexing and retrieval is more complex. The manipulation of a video database consists of three main operations: (1) partitioning of the video into clips, (2) key frame extraction, and (3) indexing and retrieval of key frames. The proposed algorithm has been applied in all three steps. First, the DC histograms are implemented for partitioning each video into clips or camera shots. Then, in the next phase the same DC histograms are used to extract key frames and create a database of key frames only. Finally, in the last step, the user submits one or more video frames that he/she is searching for.

We implemented the described algorithm for similarity-based retrieval to both image and video databases. The experimental results, presented in the paper, show that the proposed algorithm can be very efficient for similarity-based search of images and videos in distributed environments, such as Internet, Intranets, or local-area networks.

Keywords: content-based retrieval and indexing, multimedia databases, DC coefficients, histogram of DC coefficients

1. INTRODUCTION

There are two main approaches in indexing and retrieval of images and videos in multimedia databases: (a) keyword-based indexing and (b) content-based indexing. The keyword-based indexing uses keywords or descriptive text, which is stored together with images and videos in the databases. Retrieval is performed by matching the query, given in the form of keywords, with the stored keywords. This approach is not satisfactory, because the text-based description tends to be incomplete, imprecise, and inconsistent in specifying visual information.

To overcome this problem, recent research has been focused on content-based indexing and retrieval techniques [1,2,3,4,5]. This approach allows users to index and retrieve images and videos from databases using visual content (such as prominent regions, color, shape, size, and texture), motion related information (movement of objects, enlarging or shrinking, and global camera operation), and similarity-based. The current techniques, proposed in literature, mostly deal with uncompressed multimedia objects (images and videos). There are several techniques proposed for shot detection and segmentation of compressed video [3,4,5]. These techniques use block comparison metrics, which measure the differences between DCT coefficients of blocks in two frames.

In this paper, we present a technique for content-based image and video indexing and retrieval, which uses histograms of DC coefficients. We assume that images and videos in multimedia databases are stored in compressed form (JPEG for images or MPEG and M-JPEG for videos). We propose a fast retrieval and indexing algorithm that can be very efficiently used for content-based search on the Internet. The fundamental idea of the new algorithm consists of using histograms of DC coefficients only of the stored JPEG images, or I-frames in the case of the compressed MPEG or M-JPEG video. The experiments show that the histogram of DC coefficients is a very distinguishable characteristic of an image and can be effectively used for image or video retrieval and indexing. On the other hand, the calculation of the histogram of DC coefficients and related cost functions turns to be very fast and does not require computationally intensive algorithms.

2. AN ALGORITHM FOR SIMILARITY-BASED RETRIVAL OF IMAGES

The JPEG encoding standard for full-color images is based on DCT transformation. An image is divided into 8x8 blocks, and pixels from each block are transformed from spatial to frequency domain. The transformed 64-point discrete signal is a function of two spatial dimensions, and its components are called spatial frequencies or DCT coefficients. The F(0,0) coefficient is called the DC coefficient, and the remaining 63 coefficients are called AC coefficients.

For color images, represented by YUV or YCbCr format, the DCT transform is performed to all three components. The proposed algorithm is based on DC coefficients that are calculated only for Y (luminance) component. There are two reasons for this decision: (1) human visual system is more sensitive to Y than to two other chrominance components, and (2) the JPEG and MPEG standards typically use higher density for Y than for the other two components.

Histogram of DC Coefficients

The pixels of the original Y component in spatial domain are coded with 8 bits. However, after the DCT transformation, the sizes of DC coefficients of the Y component become 11 bits; the DC coefficients are in the range [-1024 to +1023]. The histogram of DC coefficients can be now created. For illustration purposes, the DC histogram is created for the image "elephant," which consists of 600x800 pixels. The image contains 75x100 microblocks, which gives 7,500 DC coefficients. The histogram of DC coefficients is shown in Figure 1.

The number of histogram bins in this example is 2048, which corresponds to all values of DC coefficients in the range [-1,024 to +1,023]. However, the histogram of DC coefficients can be reduced to a smaller size of histogram bins – 1024, 512, or 256 bins. The histogram with a smaller size of bins requires less computation when histogram similarity metric is calculated.

Histogram Similarity Metrics

Histogram similarity metrics are used to compare DC histograms of a given image with histograms of compressed images from the database. We analyzed three histogram-comparison metrics: (1) Weighted Euclidean Distance, (2) Square Difference, and (3) Absolute Difference. These three metrics are defined next.

Let's denote the j^{th} histogram bin value of a query image as $H_Q(j)$, and the j^{th} histogram bin value of an image in the database as $H_D(j)$. Then, the Weighted Euclidean Distance (WED) metric is defined as

$$WED = \sqrt{\sum_{j=1}^{N} w_j [H_Q(j) - H_D(j)]^2}$$

where:
N is the total number of histogram bins, and
w_j is the weight in bin j defined as

$$w_j = H_Q(j) \rightarrow if..H_Q(j) \neq 0$$
$$w_j = 1 \rightarrow otherwise$$

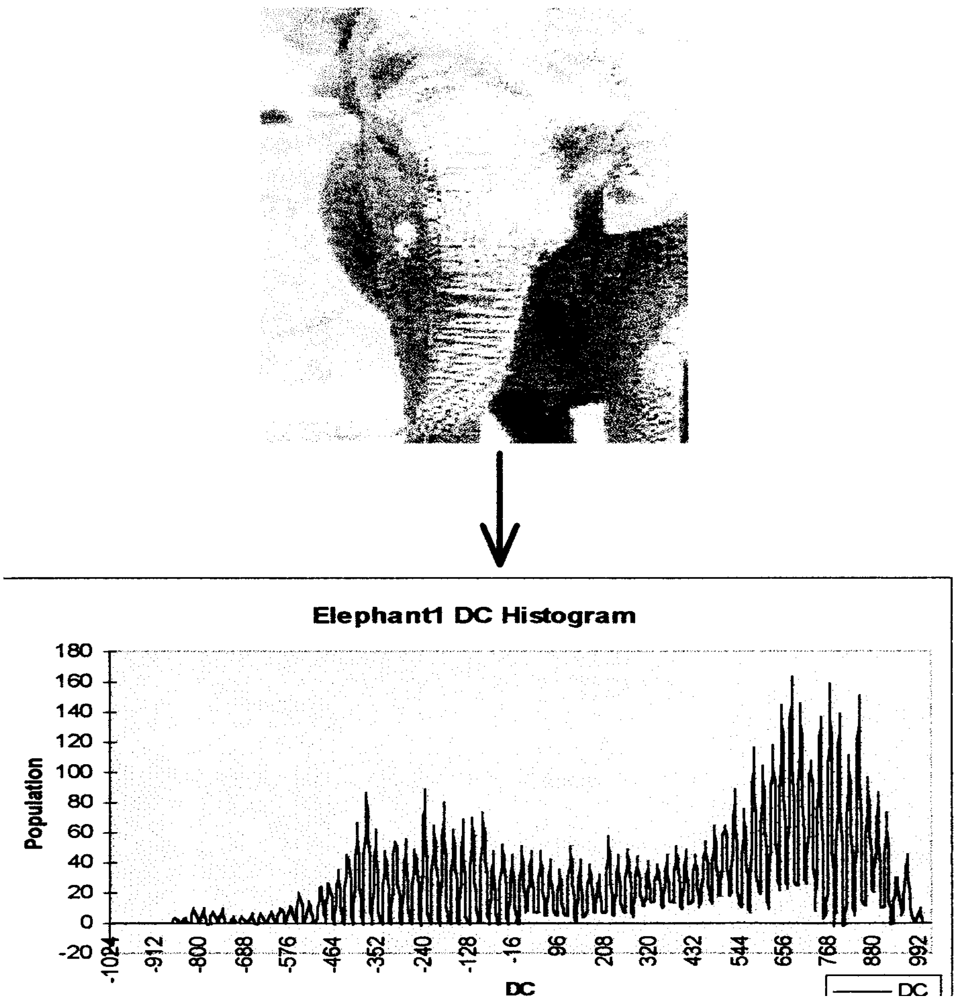

Figure 1. Histogram of DC coefficients for the image “elephant.”

The Square Difference (SD) metric is defined as

$$SD = \sum_{j=1}^{N} [H_Q(j) - H_D(j)]^2$$

and the Absolute Difference (AD) metric as

$$AD = \sum_{j=1}^{N} \left| H_Q(j) - H_D(j) \right|$$

The complexity of all three metrics depends on the number of histogram bins. Our experiments have shown that the metrics based on 512 bins perform quite well and not much worse than with 2,048 bins.

Example of Similarity-Based Retrieval of an Image Database

In the following example, we compared the efficiency of three metrics in retrieving compressed images from an image database. We created an image database, which consists of 200 images. We performed the experiments for different number of histogram bins: 2048, 1024, 512, and 256.

The user submits a request for search by similarity by presenting the desired image to the algorithm. The algorithm calculates the DC coefficients of this image. Then, one of the histogram similarity metric is calculated to compare the DC histogram of the submitted image with the DC histograms of the images stored in the database. Then, the algorithm presents to the user the set of images with the smallest values of histogram similarity metrics. The whole query process takes only a few seconds. For illustration, in Table 1 and Figure 2, results of query-by-similarity are presented. In Figure 2, the algorithm presented best 20 matches of the compressed images based on the absolute difference metric.

Table 1. Results of Retrieving Image “elephant1.jpg” from the Image Database

IMAGE NAME	WED	IMAGE NAME	SD	IMAGE NAME	AD
Elephant1.jpg	0.30	Elephant1.jpg	0	Elephant1.jpg	0
Elephant3.jpg	1650	Elephant3.jpg	9.35	Elephant3.jpg	0.5
Oregeon-sunset.jpg	2425	Icefield1.jpg	28.81	Elephant2.jpg	0.83
Icefield1.jpg	2508	Oregeon-sunset.jpg	28.99	Flower3.jpg	1.04
Icefield2.jpg	2532	Namesis2.jpg	29.87	Goat1.jpg	1.07
Chamber.jpg	2546	Icefield2.jpg	29.96	Flower7.jpg	1.09
Namesis2.jpg	2548	Namesis3.jpg	30.21	Surf1.jpg	1.11
Porcelan.jpg	2568	Namesis4.jpg	31.09	Flower6.jpg	1.12
Woman.jpg	2573	Namesis6.jpg	31.44	Flower4.jpg	1.16
Namesis3.jpg	2583	Lake-goat.jpg	32.32	Sd5.jpg	1.20

The following conclusions can be drawn from these experiments:

- All three metrics gave good results in similarity-based retrieval, but the absolute difference metric seems to be the most reliable.
- Reducing the number of histogram bins from 2,048 to 1,024 was efficient. First, it reduced the number of operations needed for the calculation of similarity metrics. Second, the smaller number of bins reduced the sensitivity of indexing due to quantization noise. However, when the number of bins was further reduced to 512 and 256, the indexing results were deteriorated.

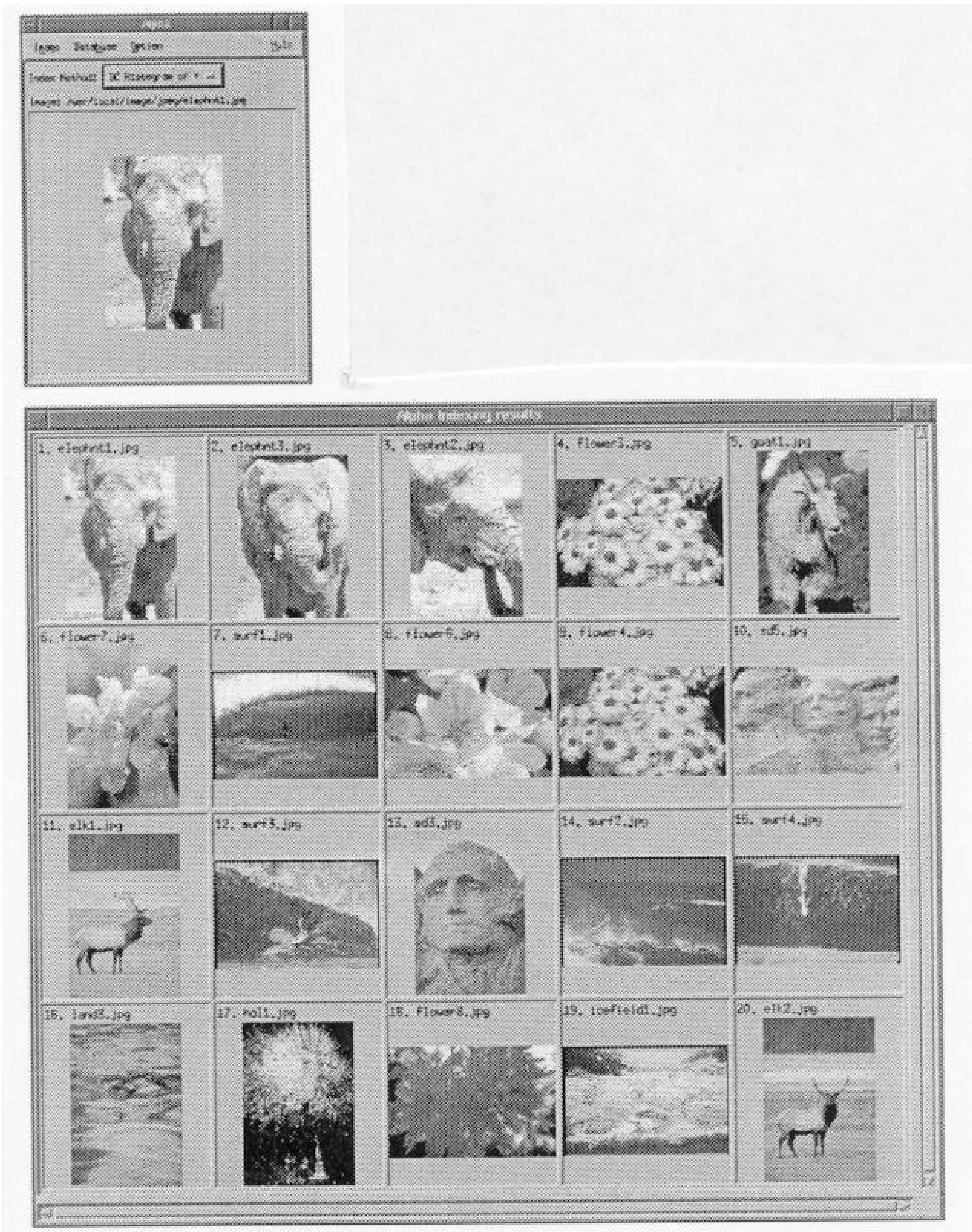

Figure 2. Example of similarity-based retrieval using the
DC histogram and the absolute difference metric.

3. AN ALGORITHM FOR SIMILARITY-BASED RETRIEVAL OF COMPRESSED VIDEOS

In the case of compressed video databases, the procedure is more complex. The manipulation of a video database consists of three main operations:

(1) Partitioning of the video into clips,
(2) Key frame extraction, and
(3) Indexing and retrieval of key frames.

The first two steps are typically performed off-line during the feature extraction phase, while the last step is performed in real time. The proposed algorithm, based on DC histograms, can be applied in all three steps. First, the DC histogram is implemented to partition each video into clips or camera shots. Then, in the next phase the same DC histogram is used to extract key frames and create a database of key frames only. Finally, in the last step, the user submits one or more video frames that he/she is searching for. The algorithm is capable of searching through the video database (key frames only) and retrieve the most similar frames or clips.

Video Partitioning

The histogram of DC coefficients can successfully be used in partitioning video by detecting camera breaks. First, let's consider M-JPEG compressed video, where all frames are I-frames. In this case, we use DC histograms to compare subsequent frames and detect camera breaks. To minimize the computational complexity, the range of DC coefficients is reduced to [-256,+255] by using the following formula:

$$F(0,0) = \frac{1}{32} \sum_{x=0}^{7} \sum_{y=0}^{7} f(x, y)$$

where:
F(0,0) is a DC coefficient, and
f(x,y) is a pixel value of y-component in a 8x8 block.

To test the similarity of histograms of subsequent frames from the same clip, we performed several experiments with standard video clips "Football," Miss America," and "Susie." Results, presented in Figure 3a-c, show two DC histograms for each clip, the histogram of frame 0 and frame 8. In all three cases the histograms of these two frames are almost identical. Then, we compared DC histograms of different clips. Figure 4 compares the DC histograms of frame 0 for these three clips. It shows that the histograms of these three frames are significantly different.

In order to detect camera breaks, we define the normalized square difference metric (NSD):

$$NSD_i = \sum_{j=1}^{N} \frac{[H_i(j) - H_{i-1}(j)]^2}{H_i(j)^2}$$

where:
NSD_i is the normalized square difference metric for frame *i*, and
$H_i(j)$ are DC histogram values for the i_{th} frame, and j is one of possible histogram levels.

If the overall difference is greater than a given threshold T_1, a camera break is declared.

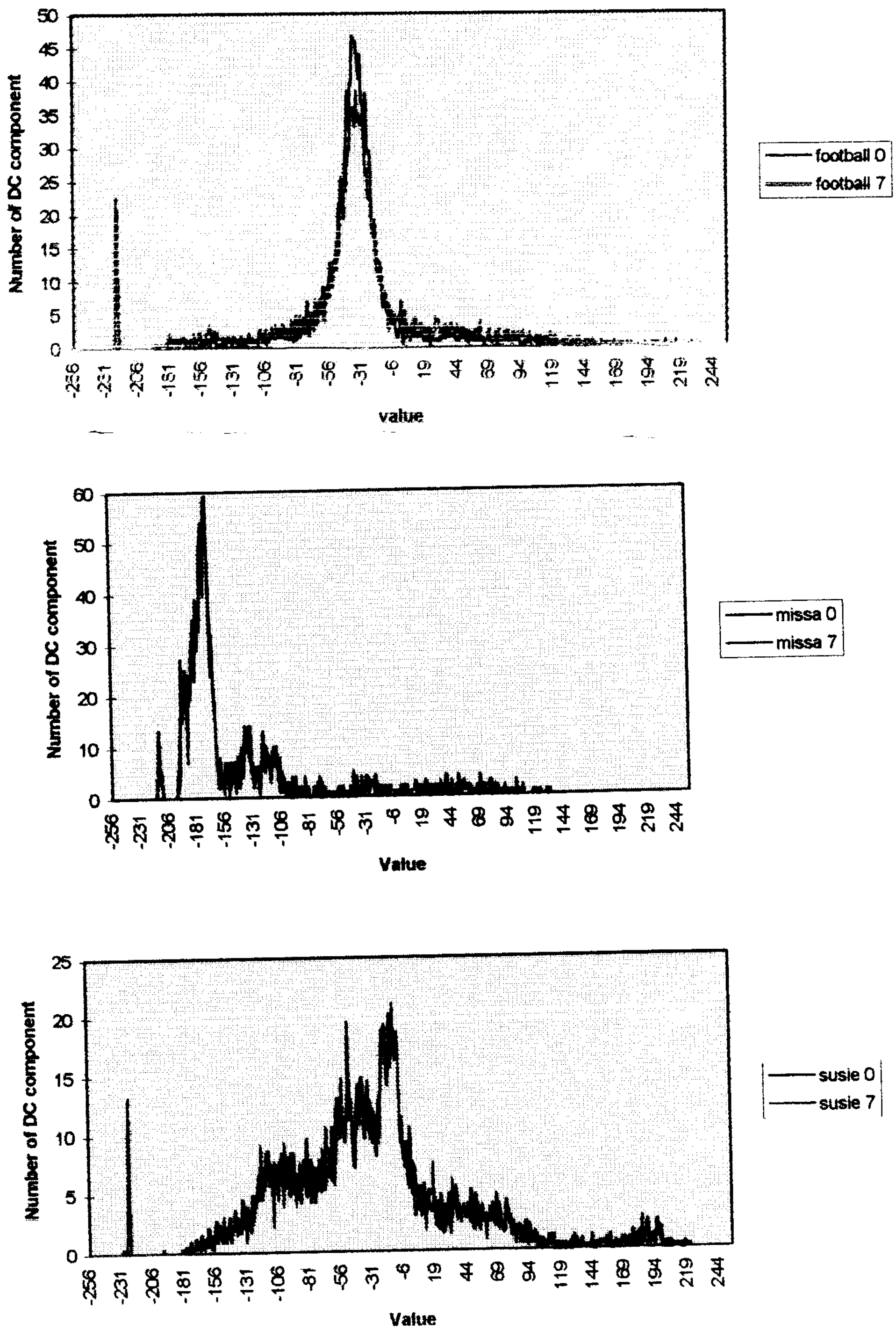

Figure 3. Histogram of DC coefficients of frames 0 and 7 for video clips:
(a) “Football,” (b) “Miss America,” and (c) “Susie.”

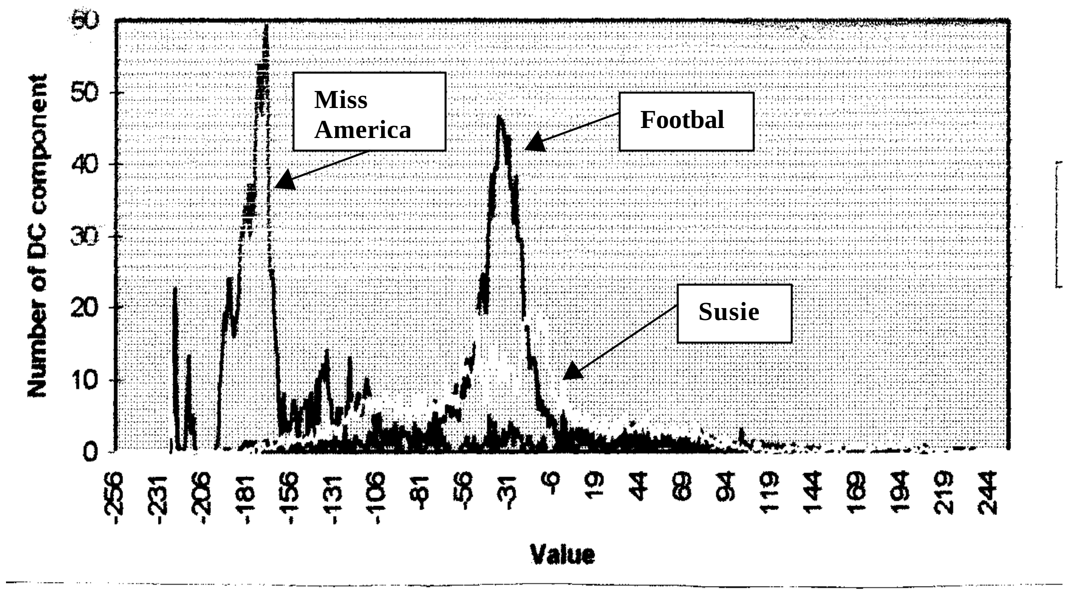

Figure 4. DC histogram comparison of frames 0 for three video clips.

To test the proposed technique, we aplied it to a composed video consisted of three clips, each containing 8 frames. The results of the video partitioning experiment are presented in Figure 5. The algorithm was able to correctly detect both camera breaks. The threshold, used in the experiment, was T_1=20.

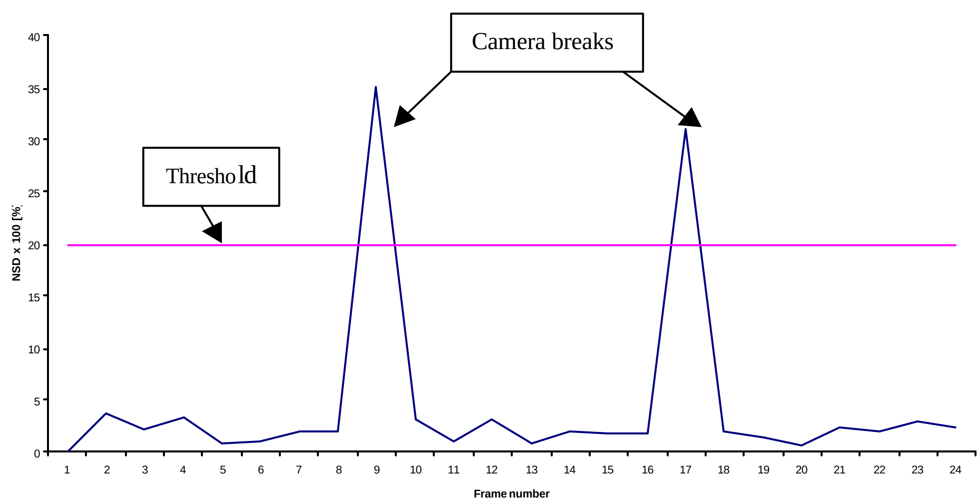

Figure 5. DC histogram comparison technique applied to video partitioning.

For a video database compressed using the MPEG technique, the video partitioning uses a two-pass approach [3]. In the first pass, the proposed technique, based on DC histograms, is applied to I-frames only. For example, for a MPEG sequence {IBBPBBPBB}{IBBPBBPBPP}, etc., the algorithm will detect the camera breaks occurred between I-frames. In the second pass, a technique based on motion vectors [7] is applied to detect the camera break within those sequences which are detected in the first pass.

Key Frame Extraction

In the next step, the key frames are extracted from the video segments identified in the first step. The DC histogram comparison technique is used in this step as well. However, the similarity metric is now defined as the accumulated difference between the current frame and the previous key frame

$$NSD_i = \sum_{j=1}^{N} \frac{[H_i(j) - H_{KF}(j)]^2}{H_i(j)^2}$$

where:
$H_{KF}(j)$ is the j^{th} histogram bin value of the DC histogram of the previous key frame.

The first frame in a video clip is always declared as the first key frame. Then, the other frames are compared to this frame. When the difference becomes greater than the threshold T_2, the current frame is declared as the next key frame. The following frames are then compared to this key frame. Figure 6 illustrates the procedure for extracting key frames. Note that the threshold T_2=10, used for the key frame extraction, is smaller than the threshold T_1, used for video partitioning. The described process is applied to I-frames only.

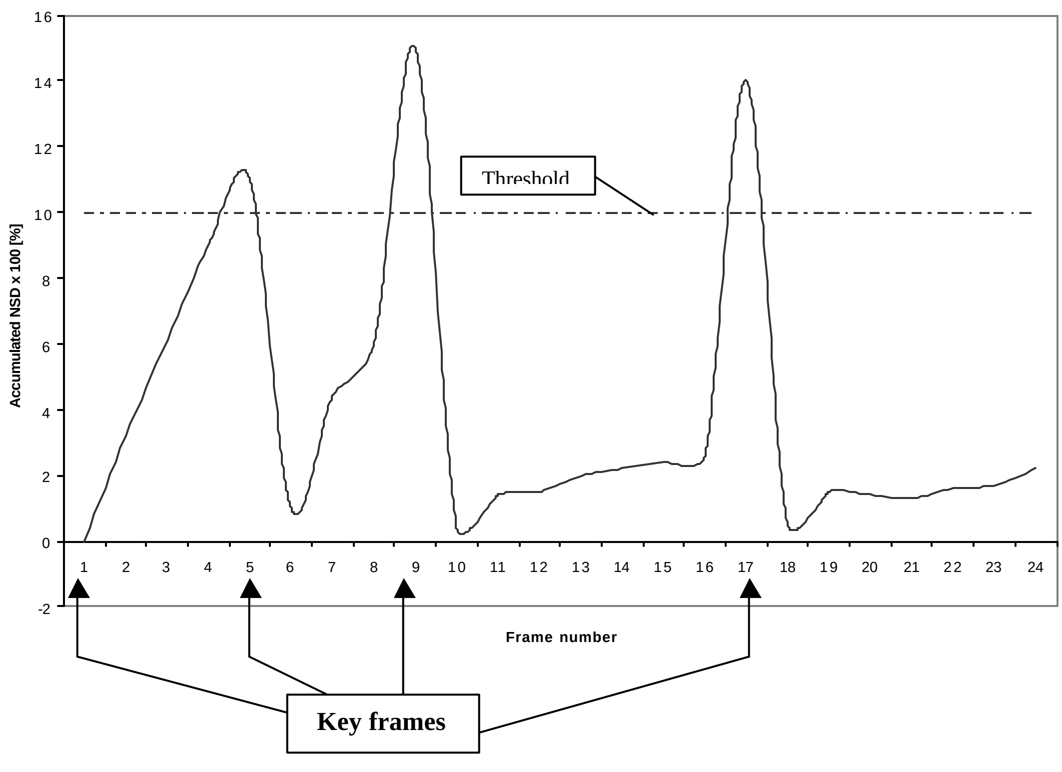

Figure 6. DC histogram comparison technique for extraction of key frames.

In the example in Figure 6, the video clip comprised of three sequences: “Football,” “Miss America,” and “Susie,” each consisting of 8 frames. The algorithm has extracted four key frames.

Indexing and Retrieval of Key Frames

Finally, in the last step, the DC histogram technique is applied to similarity-based search of extracted key frames. The set of key frames, extracted in the previous step, comprises a key-frame database, and the search is now performed on key frames only. This step is equivalent to the similarity-based retrieval of image databases, described in Section 2.

In our experiment, we created a database of key frames and applied the proposed algorithm for the retrieval of frames, which are similar to the given frame. The results are shown in Figure 7.

Figure 7. Example of similarity-based retrieval of key frames using DC histograms.

4. CONCLUSION

We presented an algorithm for similarity-based indexing and retrieval of image and video databases. The proposed algorithm is based on DC histograms of compressed images and video frames. We analyzed several histogram similarity metrics in order to select the most efficient one. The algorithm has been tested on a small compressed image database as well as on several video sequences. In summary, the proposed algorithm can be very efficient for similarity-based retrieval of images and videos in distributed environments, such as Internet, Intranets, or local-area networks.

REFERENCES

1. H.J. Zhang, S.Y. Tan, S.W. Smoliar, and Y. Gong, "Automatic Parsing and Indexing of News Video," *Multimedia Systems,* Vol. 2, No. 6, pp. 255-264, 1995.
2. E. Ardizzone and M. La Cascia, "Automatic Video Database Indexing and Retrieval," *Journal of Multimedia Tools and Applications*, Vol. 4, pp. 29-56, 1997.
3. B. Furht, S.W. Smoliar, and H.J. Zhang, "Video and Image Processing in Multimedia Systems," *Kluwer Academic Publishers*, Norwell, MA, 1995.
4. F. Arman et al., "Content-Based Browsing of Video Sequences," *Proc. of ACM Multimedia* '94, San Francisco, CA, October 1994.
5. H.J. Zhang, et al., "Video Parsing Using Compressed Data," *Proc. SPIE'94 Symposium on Image and Video Processing*, San Jose, CA, pp. 142-149, February 1994.
6. B.-L. Yeo and B. Liu, "A Unified Approach to Temporal Segmentation of Motion JPEG and MPEG Compressed Video," *Proc. IEEE International Conference on Multimedia Computing and Networking*, Washington D.C., pp. 81-88, May 1995.
7. A. Akutsu et. al., "Video Indexing Using Motion Vectors," *Proc. of SPIE'92 Symposium on Communications and Image Processing*, Boston, MA, pp. 1522-1530, November 1992.