World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:3, No:6, 2009

Fast Search for MPEG Video Clips Using Adjacent Pixel Intensity Difference Quantization Histogram Feature

Feifei Lee, Qiu Chen, Koji Kotani, and Tadahiro Ohmi

International Science Index, Electrical and Computer Engineering Vol:3, No:6, 2009 waset.org/Publication/8188

Abstract**—**In this paper, we propose a novel fast search algorithm for short MPEG video clips from video database. This algorithm is based on the adjacent pixel intensity difference quantization (APIDQ) algorithm, which had been reliably applied to human face recognition previously. An APIDQ histogram is utilized as the feature vector of the frame image. Instead of fully decompressed video frames, partially decoded data, namely DC images are utilized. Combined with active search [4], a temporal pruning algorithm, fast and robust video search can be realized. The proposed search algorithm has been evaluated by 6 hours of video to search for given 200 MPEG video clips which each length is 15 seconds. Experimental results show the proposed algorithm can detect the similar video clip in merely 80ms, and Equal Error Rate (ERR) of 3 % is achieved, which is more accurately and robust than conventional fast video search algorithm.

Keywords**—**Fast search, Adjacent pixel intensity difference quantization (APIDQ), DC image, Histogram feature.

I. Introduction

VIDEO retrieval has become an active area of research in recent years because video content becomes commonplace on the web and the size of video database quickly increases due to rapid developments of internet connection and disk storage technology. Video search is an important problem in this area because it has a wide range of applications such as TV commercials detection [1], video copyright enforcement [2], [3], video clustering and so on. In this paper, video search means when a user presents a query video clip to the search engine, the search engine should identify all similar ones, that is to say, accurately locate the position of query video clip if it exists in the video database.

Many video search algorithms [7]-[9] have been proposed, and achieves successes to a certain extent. But such algorithms, however, are computational- power hungry for the exhaustive search of large video database. For large video database, Search speed is an important issue of video search. Base on active search [4], a temporal pruning algorithm, Kashino et al. [1] improved the conventional multimedia search algorithm. Nevertheless, their feature extraction utilizes intensity features of the frame image, so the results may be sensitive to small change of luminance and motion in the frame. In this paper, we utilizes a new feature based on the adjacent pixel intensity difference quantization (APIDQ) algorithm, which had been reliably applied to human face recognition previously [5]. It has the following advantages: computational simplicity, motion-insensitivity and luminance- insensitivity. Because such a feature is compatible with active search algorithm, fast search speed can also be achieved by combining APIDQ and active search.

On the other hand, in many algorithms, the compressed video sequences are usually decoded to separate frames firstly by computational processing steps before video search step. To realize real time application, we utilize DC images which partially decoded from MPEG compressed video [10] by fast processing steps.

In section II, we will first introduce the *Adjacent Pixel Intensity Difference Quantization (APIDQ)* histogram feature which had been successfully applied to human face recognition previously, and then describe fast video search algorithm for short MPEG compressed video clips we employ in section III. Experimental results compared to conventional fast search approach will be discussed in section IV. Finally, conclusions are given in section V.

II. Adjacent Pixel Intensity Difference Quantization (APIDQ)

The Adjacent Pixel Intensity Difference Quantization (APIDQ) histogram method [5] has been developed for face recognition previously. Figure 1 shows the processing steps of APIDQ histogram method. In APIDQ, for each pixel of an input image, the intensity difference of the horizontally adjacent pixels (*d*Ix) and the intensity difference of the vertically adjacent pixels (*d*Iy) are first calculated by using simple subtraction operations shown as formula (1).

$$\begin{aligned} dIx(i,j) &= I(i+1,j) - I(i,j) \\ dIy(i,j) &= I(i,j+1) - I(i,j) \end{aligned} \qquad (1)$$

A calculated (*dIx*, *dIy*) pair represents a single vector in the *dIx*-*dIy* plane. By changing the coordinate system from orthogonal coordinates to polar coordinates, the angle θ and the distance r represent the direction and the amount of intensity variation, respectively. After processing all the pixels in an input image, the dots representing the vectors are distributed in

Feifei Lee, Qiu Chen, and Tadahiro Ohmi are with New Industry Creation Hatchery Center, Tohoku University, Sendai, 980-8579 Japan (phone: +81-22-795-3977; fax: +81-22-795-3986; e-mail: fei@fff.niche.tohoku.ac.jp).

Koji Kotani is with Department of Electronics, Graduate School of Engineering, Tohoku University, Sendai, 980-8579 Japan.

the *dIx-dIy* plane. The distribution of dots (density and shape) represents the features of the input image.

Each intensity variation vector is then quantized in the r-θ plane. Quantization levels are set at 8 in θ-axis and 8 in *r*-axis (totally 50). Since *dIx-dIy* vectors are concentrated in small-r (small-*dIx*, *-dIy*) region, non-uniform quantization steps are applied in *r*-axis. The number of vectors quantized in each quantization region is counted and a histogram is generated.

In the face recognition approach, this histogram becomes the feature vector of the human face. Experimental results show recognition rate of 95.7 % for 40 persons' 400 images of publicly available database of AT&T Laboratories Cambridge [6] containing variations in lighting, posing, and expressions. The total recognition processing time is only 31 msec running on a conventional PC (AMD Athron 1.1GHz), enabling the video-rate face recognition.

The essence of the APIDQ histogram method can be considered that the operation detects and quantizes the direction and the amount of intensity variation in the image block. Hence the APIDQ histogram contains very effective image feature information. We will describe how to apply it as feature vector of frame to solving the fast video search problem in next section.

III. Proposed Fast Search Algorithm

The procedure of proposed fast search algorithm is shown in figure 2. In the preprocessing stage, firstly, DC images sequences are obtained from the query MPEG compressed video clip and the video database respectively by partial decoding method proposed in [10]. The processing is

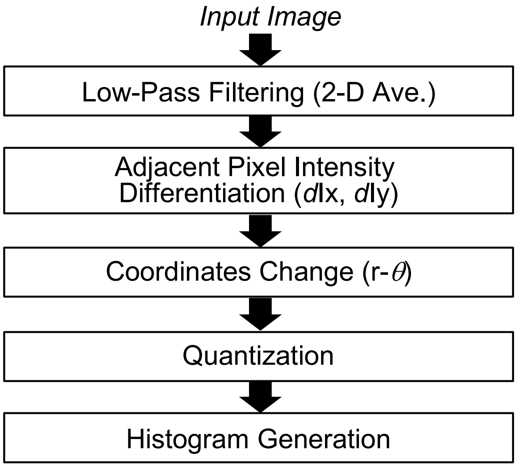
Input Image
Low-Pass Filtering (2-D Ave.)
Adjacent Pixel Intensity Differentiation (dIx, dIy)
Coordinates Change (r-θ)
Quantization
Histogram Generation

Fig.1 Processing steps of APIDQ histogram method.

performed directly on compressed data, so full decompression by computational processing steps are not needed. Furthermore, because the size of DC image will be 1/64 of the original image, posterior processing can only deal with a small fraction of the original video data. Then the feature vectors are calculated from the DC images of the query video clip and the video database by APIDQ histogram method described in section II. The feature vectors are then quantized using VQ algorithm which the bin boundaries are selected so that the same number of feature vectors fall in the bins for each dimension [1], then histograms obtained by counting the number of vectors in each

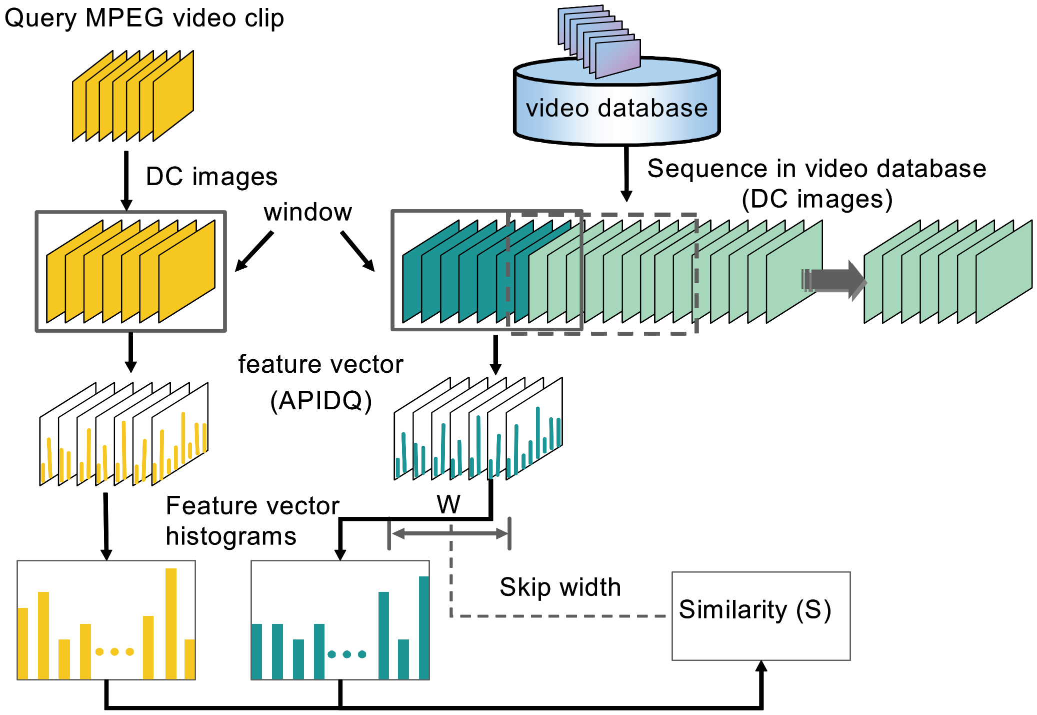

Fig.2 The procedure of proposed fast search algorithm.

quantization bin.

In the search stage, the windows are applied to both the query feature vectors and the feature vectors of video database. In the next step, the number of vectors quantized in the windows of the query video clip and video database are counted and feature vector histograms are created respectively. The similarity between these histograms is then calculated. If the similarity exceeded a threshold value given previously, the query video clip will be detected and located. Otherwise, the window on the video database will be skipped to the next position determined by the similarity in current position and the threshold value. In the last step, the window on the video database is shifted forward in time and the search proceeds.

Here, histogram intersection is used as the similarity measure [4], and is defined as formula (2).

$$\begin{aligned} S_{QD} &= S(H_Q, H_D) \\ &= \frac{1}{N}\sum_{l=1}^{L} \min(h_{Ql}, h_{Dl}) \end{aligned} \tag{2}$$

where h_{Ql}, h_{Dl} are the numbers of feature vectors contained in the *l-th* bin of the histograms for the query and the stored signal, respectively, L is the number of histogram bins, and N is the total number of feature vectors contained in the histogram. The skip width w is shown by formula (3).

$$w = \begin{cases} floor(N(\theta - S_{QD})) + 1 & (S_{QD} < \theta) \\ 1 & otherwise \end{cases} \tag{3}$$

where *floor(x)* means the greatest integral value less than x, and θ is a given threshold.

IV. Experimental Results and Discussions

We performed all of the experiments on a conventional PC @ 3.2GHz (1G memory). The algorithm was implemented in ANSI C. We used 6 hours of video captured from TV program. In our experiments, the video frame rate was 29.97 fps, and the captured frame size was 320*240 as shown in table I and the size of corresponding DC image was 40*30. We captured 6 hours of video twice, one for video database sequence and the other for generating query video clips, and then saved them as MPEG videos. Query video clips were generated by selecting video clips randomly for 200 times from the second video, and also saved as MPEG compressed video clips. Then we can perform search for 200 video clips from 6 hours of video.

We utilized a 1-hour video sequence by selecting randomly from the second video to determine boundary threshold which were used to implement scalar VQ process (SQ).

To suit the search task, quantization levels of APIDQ are set at 8 in θ-axis and 2 in r-axis (totally 9) in the feature extraction stage. Thus, the number of histogram bins is total 512. Similarity calculation between the feature vector histograms will be quite faster compared with conventional algorithm which number of histogram bins is 4096.

TABLE I
Parameters of video dataset

Video content	News, drama, sports etc.
Video length	Query video clips: 15s * 200 Video database sequence: 6 hours
Frame rate	29.97 fps
Frame number	Query video clip: 450 Video database sequence: 647,352
Video format	MPEG1
Frame size	320*240

TABLE II
Approximate Computational Cost Table(CPU time)

Stage	Full search	Conventional	Proposed algorithm
Feature Extraction	639sec	639sec	650sec
VQ processing	55ms	55ms	40ms
Search	22sec	560ms	80ms

A. Image Features of Conventional Algorithm

In conventional algorithm [1], they use small scaled images as video features. An image feature vector is defined as formula (4).

$$g(k) = (g_1(k), \cdots, g_j(k) \cdots, g_W(k)) \tag{4}$$

where k is the frame number, j is the division number of the subimages, and W is the number of subimages. The $g_j(k)$ is the normalized intensity and is defined as formula (5).

$$g_j(k) = \frac{\bar{x}_j(k) - \min_i \bar{x}_i(k)}{\max_i \bar{x}_i(k) - \min_i \bar{x}_i(k)} \tag{5}$$

where $\bar{x}_j(k)$ is the average intensity in the j-th subsection.

B. Experimental Results

We compared our algorithm with the algorithm which does not utilize active search (full search), and conventional fast search algorithm described in section IV-A. Table 2 gives the approximate computational cost of the algorithms. As descried above, the number of histogram bins is total 512 in our proposed algorithm, which is 8 times smaller than that of conventional algorithm. From Table II, we can see the search time costs only 80ms, which is 275 times faster than full search, and also 7 times faster than the conventional fast search algorithm.

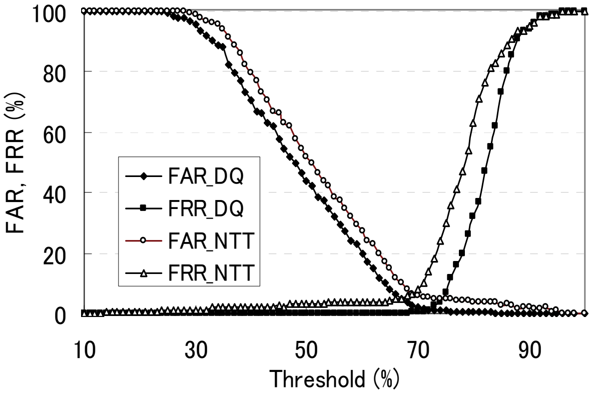

Fig.3 Comparison of FAR and FRR (Using fully decompressed video frames).

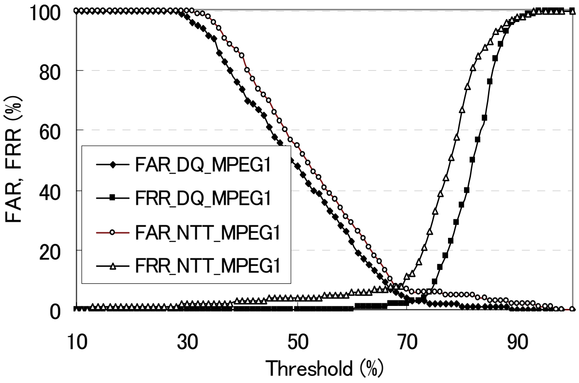

Fig.4 Comparison of FAR and FRR (DC images).

For practical applications of video search system, not a simple accuracy rate but a *False Acceptation Rate* (FAR) and a *False Rejection Rate* (FRR) are more important. Figure 3 shows FAR and FRR plots for search experiment by using the video dataset consisting of 6 hours of video and 200 video clips (captured video frames without compression). Compared with the value of *Equal Error Rate* (ERR) of about 6.5% obtained by conventional fast search approach (FAR_NTT, FRR_NTT), 1% is achieved at the threshold of about 0.7 by using proposed algorithm (FAR_DQ, FRR_DQ). Figure 4 shows FAR and FRR plots by using the MPEG compressed videos described in table I. In this case, ERR of conventional fast search approach (FAR_NTT_MPEG1, FRR_NTT_MPEG1) increases to 8%, and ERR of proposed algorithm (FAR_DQ_MPEG1, FRR_DQ_MPEG1) increases to 3%. Compared with the results obtained by utilizing captured video frames without compression, the value of ERR changes little although the image quality has been changed largely between frame images without compression and DC images. Furthermore, proposed fast search algorithm is more accurately and robust for video search task than the conventional approach.

V. Conclusion

By using a new feature based on the adjacent pixel intensity difference quantization (APIDQ) algorithm, we present a fast and robust video search algorithm for video clips from large video database. The proposed search algorithm has been evaluated by 6 hours of video to search for 200 MPEG compressed video clips. Experimental results show that search time costs only 80ms, which is 275 times faster than full search, and also 7 times faster than the conventional fast search algorithm. Furthermore, Equal Error Rate (ERR) of 3 % is achieved by proposed algorithm, which is more accurately and robust than conventional fast video search algorithm.

Acknowledgement

This research was partially supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan, Grant-in-Aid for Young Scientists (B), No.20700074, 2008-2010.

References

[1] K. Kashino, T. Kurozumi, and H. Murase, “Quick AND/OR search for multimedia signals based on histogram features”, *IEICE Trans., J83-D-II*, vol.12, 2000, pp. 2735-2744.

[2] S.S. Cheung and A. Zakhor, “Efficient video similarity measurement with video signature”, *IEEE Trans. on Circuits and System for Video Technology,* vol.13, no.1, 2003, pp. 59-74.

[3] A. Hampapur, K. Hyun, and R. Bolle, “Comparison of sequence matching techniques for video copy detection”, *SPIE. Storage and Retrieval for Media Databases 2002, 4676*, San Jose, CA, USA, 2002, pp. 194-201.

[4] V.V. Vinod, H. Murase, “Focused color intersection with efficient searching for object extraction”, *Pattern Recognition,* vol. 30, no.10, 1997, pp. 1787-1797.

[5] K. Kotani, F.F. Lee, Q. Chen, and T. Ohmi, “Face recognition based on the adjacent pixel intensity difference quantization histogram method”, *Proc. 2003 Int. Symposium on Intelligent Signal Processing and Communication Systems, D7-4*, Japan, 2003, pp. 877-880.

[6] AT&T Laboratories Cambridge, The Database of Faces, at http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[7] L. Agnihotre, N. Dimitrova, T. McGee, S. Jeannin, S. Schaffer, J. Nesvadba, “Evolvable visual commercial detector”, *IEEE. International Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. 79-84.

[8] R. Lienhart, C. Kuhmunch, W. Effelsberg, “On the detection and recognition of television commercials”, *In Proc. IEEE Conf. on Multimedia Computing and Systems*, 1997, pp. 509-516.

[9] R. Mohan, “Video sequence matching”, *In Proc. of the International Conference on Audio, Speech and Signal Processing*, vol.6, 1998, pp. 3679-3700.

[10] B. Yeo and B. Liu, “Rapid scene analysis on compressed videos”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol.5, no.6, 1995, pp. 533-544.