Fast Video Search Algorithm for Large Video Database Using Adjacent Pixel Intensity Difference Quantization Histogram Feature

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

†New Industry Creation Hatchery Center, Tohoku University, Japan
††Department of Electronics, Graduate School of Engineering, Tohoku University, Japan

Summary
In this paper, we present a fast and robust video search algorithm for large video database using the histogram feature which is essentially different from conventional ones. 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 a frame image. Combined with active search [4], a temporal pruning algorithm, fast and robust video search can be achieved. The proposed search algorithm has been evaluated by 6 hours of video to search for given 200 video clips which having a each length of 15 seconds. Experimental results show the proposed algorithm can detect the similar video clip in merely 80ms, and is more accurate and robust against Gaussian noise than conventional fast video search algorithm.
Key words:
adjacent pixel intensity difference quantization (APIDQ), histogram feature, video search, active search

1. Introduction

The price of storage devices, such as hard drives, used in computers for storing videos, has been falling rapidly over the past several years. On the other hand, internet connection speed is becoming faster and faster. Consequently, video content becomes commonplace on the web and the size of video database quickly increases in recent years. Video retrieval has become a hot area of research. 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. It is likely to be increasingly important as the basic video processing technology. 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]-[14] 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][15]. 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.

In section 2, we will first introduce APIDQ histogram feature, and then describe fast video search algorithm we employ in section 3. Experimental results compared to conventional search approach will be discussed in section 4. Finally, conclusions are given in section 5.

2. Adjacent Pixel Intensity Difference Quantization (APIDQ)

The Adjacent Pixel Intensity Difference Quantization (APIDQ) histogram method [5] has been developed for face recognition previously. ure 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).

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

Manuscript received September 5, 2009
Manuscript revised September 20, 2009

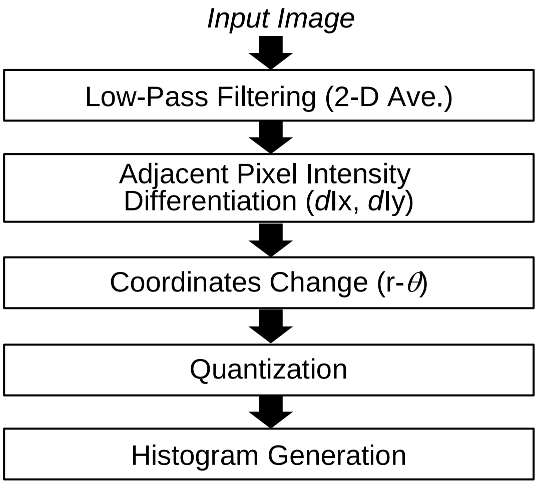
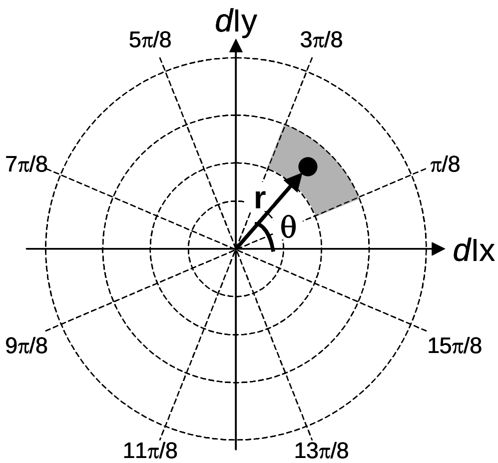


Fig. 1 Processing steps using APIDQ histogram method.



θ: -π/8 π/8 3π/8 5π/8 7π/8 9π/8 11π/8 13π/8 15π/8

r: 1 2 4 7 12 20 30

0							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*
33	*34*	*35*	*36*	*37*	*38*	*39*	*40*
41	*42*	*43*	*44*	*45*	*46*	*47*	*48*
49							

Fig. 2 Processing steps using APIDQ histogram method. Quantization table in *r*-θ plane. The *italic* numbers in the table represent the index numbers of each quantization region. Quantization levels are set at 8 in θ-axis and 8 in *r*-axis (totally 50). Since *d*Ix-*d*Iy vectors are concentrated in small-r (small-*d*Ix, -*d*Iy) regions, non-uniform quantization steps are applied in *r*-axis.

the *d*Ix-*d*Iy 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 the *d*Ix-*d*Iy 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. The quantization table is shown in Figure 2. The *italic* numbers in the table represent the index numbers of each quantization region. Quantization levels are set at 8 in θ-axis and 8 in *r*-axis (totally 50). Since *d*Ix-*d*Iy vectors are concentrated in small-r (small-*d*Ix, -*d*Iy) 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 ORL database [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.

3. Proposed Fast Video Search Algorithm

3.1 Overview

The procedure of proposed fast search algorithm is shown in Figure 3. Firstly, the feature vectors are calculated from the query video clip by APIDQ histogram method described in section 2. Next, the extracted feature vectors are quantized using a simple VQ algorithm [1] which will be described in section 3.2, then a histogram obtained by counting the number of vectors in each quantization bins. On the other hand, the feature vectors are also extracted from the sequence of video database using the same APIDQ histogram method, and then the search procedure is implemented. Firstly, the windows with same size (the length of query video sequence) 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 window of video database is counted and feature vector histogram is created. The similarity between the

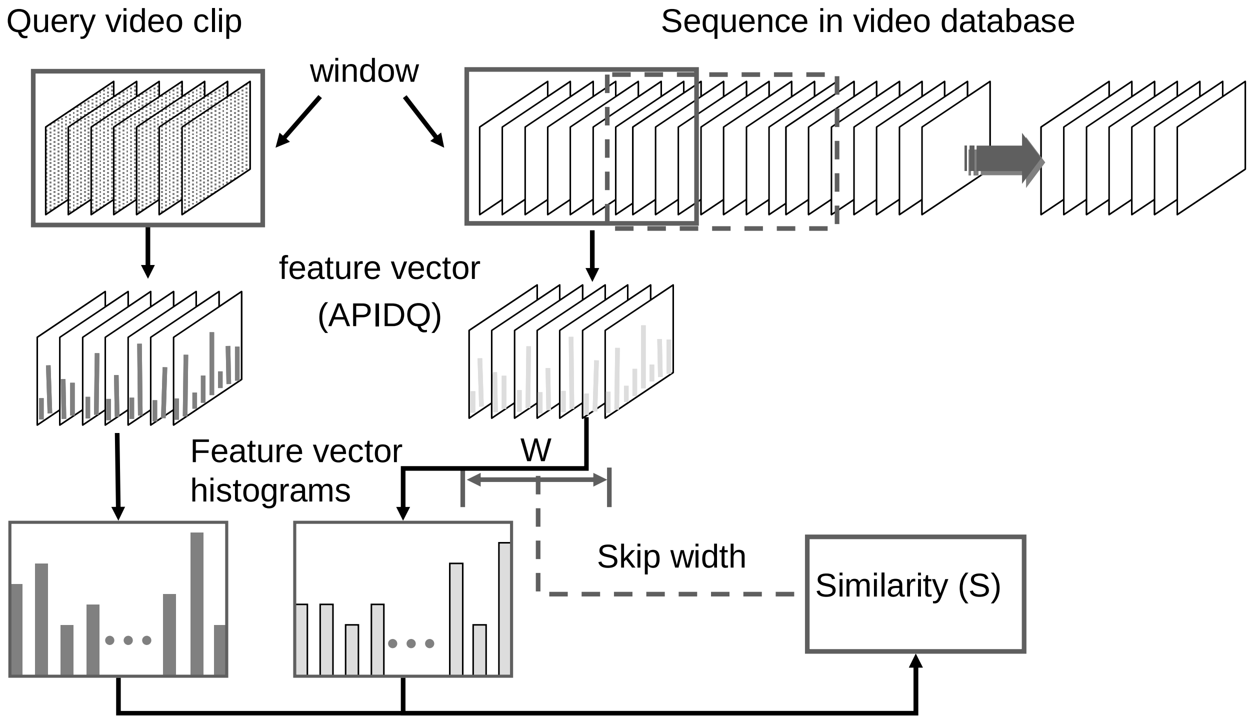


Fig. 3 The procedure of proposed fast search algorithm.

histogram of query video clip and that obtained from video database is then calculated. If the similarity exceeded a given threshold, the location in the sequence of video database for the query video clip can be considered to be determined. Otherwise, the window on the video database will be shifted to the next position determined by the similarity at current position and the threshold value. In this way, the window on the video database is shifted forward in the time sequence and the search proceeds.

3.2 Quantization method

There are many vector quantization (VQ) methods which can classify the feature vectors into a certain number of types, such as LBG algorithm [16] etc. In this paper, considering the computational simplicity, we use the method which the bin boundaries are selected so that the same number of feature vectors fall in the bins for each dimension [17]. The method is descried as follows.

When the probability that a vector belongs to a certain bin ($b_1, b_2, \cdots, b_V$) is denoted to $P(B_1 = b_1, B_2 = b_2, \cdots, B_V = b_V)$ or simpler form $P(b_1, b_2, \cdots, b_V)$, the entropy of the histogram E can be defined as:

$$E = -\sum_{b_1}\sum_{b_2}\cdots\sum_{b_V} P(b_1, b_2, \cdots, b_V) \times \log P(b_1, b_2, \cdots, b_V) \qquad (2)$$

If $P(b_1, b_2, \cdots, b_V)$ comes to be equal probability towards to $b_1, b_2, \cdots, b_V$, the entropy of the histogram E will obtain maximum value according to the maximum entropy principle (MEP). The recognition performance can be expected to be best because the average information content will be maximum at this point. Furthermore, if we assume that the bins of vector are independent of each other, we can get the following formula.

$$P(b_1, b_2, \cdots, b_V) = P(B_1 = b_1)P(B_2 = b_2)\cdots P(B_V = b_V) \qquad (3)$$

So we can see the average information content of the histogram will be maximized when $P(B_j = b_j)$ is constant (equal probability) towards to all of b_j (here $j = 1, 2, \cdots, V$).

In learning procedure, the strategy is that firstly we sub-sample the vectors, and then decide the boundary of bins so as to make the same amount of feature vectors to belong to different bins.

It is necessary to choose a suitable number of bins. If the number of bins increases, the computational amount will increase rapidly as well as the resolution of the histogram may become so sensitive that noise-corrupted vectors may significantly distort the histogram.

The total number of histogram bins L can be defined as

$$L = b^N \tag{4}$$

where the number of bins for each element is b and the feature dimension is *N*.

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 bins for each element *b* is 2. In this experiment, *N* is set to be 9, so the number of histogram bins *L* is total 512.

3.3 Similarity measure

Histogram intersection is used as the similarity measure between histograms obtained from query and saved videos [4], and is defined as

$$\begin{aligned} S_{QD} &= S(H_Q, H_D) \\ &= \frac{1}{N}\sum_{l=1}^{L}\min(h_{Ql}, h_{Dl}) \end{aligned} \tag{5}$$

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. As you can imagine, similarity will be the maximum, 1, if H_Q equals H_D and decreases with the increase in the difference between them.

3.4 Skip width

The skip width *w* is shown by formula (6).

$$w = \begin{cases} floor(N(\theta - S_{QD})) + 1 & (S_{QD} < \theta) \\ 1 & otherwise \end{cases} \tag{6}$$

where *floor(x)* means the greatest integral value less than *x*, and θ is a given threshold. *N* is the number of frames of the corresponding matching window.

4. 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 the experiment, the video frame rate was 14.97 fps, and image size was 80*60 as shown in table I. We captured 6 hours of video twice, one for video database sequence and the other for query video clips. Query video clips were generated by selecting video clips randomly for 200 times from the second video. Then we can perform search for 200 video clips from 6 hours of video. The threshold θ is 0.7 determined by FAR and FRR curve obtained by pre-executed experiments, which will be discussed later.

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	14.98 fps
Frame number	Query video clips: 15s * 200 Video database sequence: 6 hours
Image format	PPM
Capture size	80*60

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

As described above, quantization levels of APIDQ are set at 8 in θ-axis and 2 in *r*-axis (*b* = 2) in the feature extraction stage. Thus, the number of histogram bins *L* is total 512. Similarity calculation between the feature vector histograms will be quite faster compared with conventional algorithm [1] in which number of histogram bins is 4096.

4.1. Image Features of Conventional Algorithm

In conventional algorithm [1], small scaled images are used as video features. An image feature vector is defined as formula (7).

$$g(k) = (g_1(k), \cdots, g_j(k) \cdots, g_W(k)) \tag{7}$$

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

$$g_j(k) = \frac{\bar{x}_j(k) - \min_i \bar{x}_i(k)}{\max_i \bar{x}_i(k) - \min_i \bar{x}_i(k)} \tag{8}$$

4.2. Experimental Results

Figure 4 shows an example of result in 200 times search. The ellipse marks the correct location of detected query video clip. Results of search accuracy are shown in Figure 5. Search accuracy is shown as a function of window size.

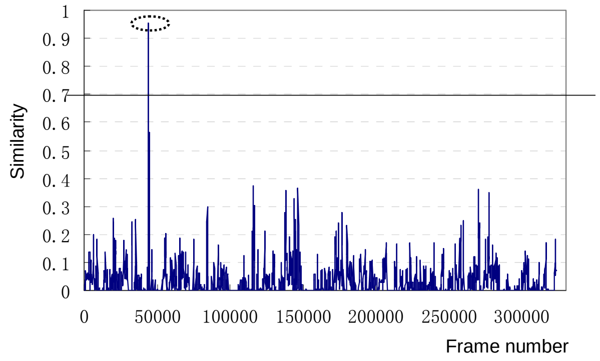


Fig. 4 The example of search result.

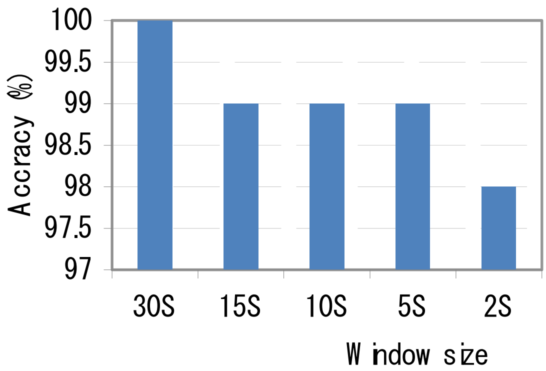


Fig. 5 Window size vs. search accuracy. Search accuracy is shown as a function of window size. The perfect accuracy of 100% is obtained when window size is given 30sec. But even if window size decreases to 2sec, the accuracy still remains 98%.

The perfect accuracy of 100% is obtained when window size is set at 30sec. But even if window size decreases to 2sec, the accuracy still remains at 98%.

TABLE II
APPROXIMATE COMPUTATIONAL COST (CPU TIME).

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

We also compared our algorithm with the algorithm which does not utilize active search (full search), and conventional fast search algorithm [1] described in section 4.1. Table II gives the approximate computational cost of

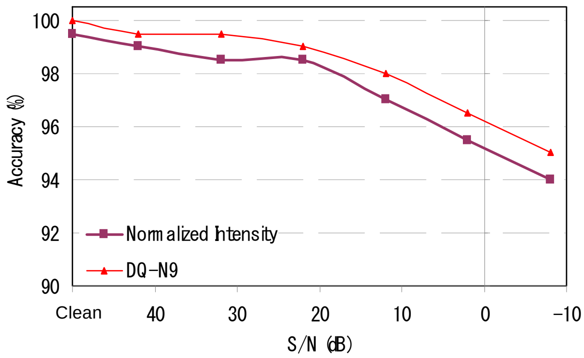


Fig. 6 Gaussian noise vs search accuracy. Although the search accuracy decreases with increase the amount of Gaussian noises, it can be said that proposed algorithm is more robust for video search task than the conventional approach.

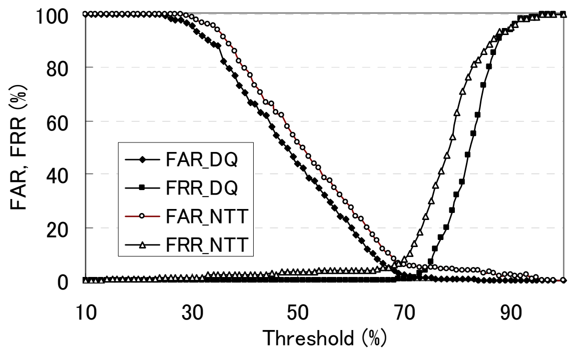


Fig. 7 Comparison of FAR and FRR. Compared with the value of Equal Error Rate (ERR) of about 6.5% obtained by conventional fast search approach (FAR_NTT, FRR_NTT), 1% is achieved at the threshold of about 0.7 by using proposed algorithm (FAR_DQ, FRR_DQ).

the algorithms. As descried above, the number of histogram bins is total 512 in our proposed algorithm, 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.

We also investigated the robustness of image features used in respective algorithms by adding Gaussian noise to the query video clips. Figure 6 shows how the search accuracy changes with the amount of noises. The curves with trigonal mark and foursquare mark stand for proposed algorithm and conventional algorithm, respectively. Our proposed algorithm achieves higher search accuracy than conventional algorithm. Although the search accuracy decreases with increase the amount of Gaussian noises, it can be said that proposed algorithm is more robust for video search task than the conventional approach.

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 7 shows FAR and FRR plots for search experiment using another video dataset consisting of 6 hours of video and 200 video clips. 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). This can explain why we select 0.7 as the global threshold for video search.

5. Conclusions and future works

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 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, proposed algorithm is more robust against Gaussian noise for video search task than the conventional approach.

To realize real time application, we will investigate how to perform processing directly on compressed video like MPEG without full decompression as our future works.

6. 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.

7. 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, no.12, pp. 2735-2744, 2000 (In Japanese).

[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, pp. 59-74, 2003.

[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, pp. 194-201, 2002.

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

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

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

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

[8] R. Lienhart, C. Kuhmunch, W. Effelsberg, "On the detection and recognition of television commercials", In Proceedings of. IEEE Conference on Multimedia Computing and Systems, pp. 509-5161, 1997.

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

[10] A. K. Jain, A. Vailaya, and X. Wei, "Query by video clip", Multimedia Systems, vol.7, no.5, pp.369-384, 1999.

[11] S. Kim and R. Park, "An efficient algorithm for video sequence matching using the modified hausdorff distance and the directed divergence", IEEE Transactions on Circuits and Systems for Video Technology, vol.12, no.7, pp.592-596, 2002.

[12] P. Sand and S. Teller, "Video matching", ACM Transactions on Graphics, vol.23, no.3, pp.592-599, 2004.

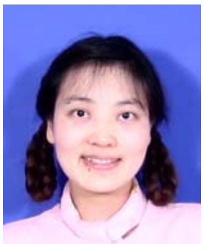
[13] S. Guimaraes and R. Kelly, "Counting of video clip repetitions using a modified bmh algorithm", Preliminary results, In Proceeding of IEEE International Conference on Multimedia Expo, pp.1065-1068, 2006.

[14] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "Segmenting, modeling, and matching video clips containing multiple moving objects", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.29, no.3, pp.477-491, 2007.

[15] F. Lee, K. Kotani, Q. Chen, T. Ohmi, "Fast search for large video database using adjacent pixel intensity difference quantization histogram", In Proceeding of the 10th IASTED International Conference on Signal and Image Processing (SIP 2008), 623-181, pp.114-117, Kailua-Kona, HI, USA, Aug. 2008.

[16] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design", IEEE Trans. Commun., vol.COM-28, pp.84-95, 1980.

[17] K., Kashino, G. A. Smith, and H. Murase, "A quick search algorithm for acoustic signals using histogram features – Time-series active search", IEICE Trans., J82-D-II, no.9, pp. 1365-1373, 1999 (In Japanese).



Feifei Lee received Ph.D. degree in electronic engineering from Tohoku University, Japan, in 2007. She is currently an assistant professor at New Industry Creation Hatchery Center (NICHe), Tohoku University. Her research interests include pattern recognition and multimedia processing. Dr. F. Lee is a member of IEEE.

Koji Kotani received the B.S., M.S. and Ph.D. degrees all in electronic engineering from Tohoku University, Japan, in 1988, 1990 and 1993, respectively. He joined the Department of Electronic Engineering, Tohoku University as a research associate in 1993. From 1997 to 1998, he belonged to the VLSI Design and Education Center (VDEC) of the University of Tokyo. He is currently an associate professor at Department of Electronic Engineering, Tohoku University. He is engaged in the research and development of high performance devices/circuits as well as intelligent electronic systems. Dr. K. Kotani is a member of IEEE and a member of the Institute of Electronics, Information and Communication Engineers of Japan.

Qiu Chen received Ph.D. degree in electronic engineering from Tohoku University, Japan, in 2004. During 2004-2007, he was a COE fellow of Tohoku University 21st century COE program. He is currently an assistant professor at New Industry Creation Hatchery Center (NICHe), Tohoku University. His research interests include pattern recognition, computer vision, information retrieval and their applications. Dr. Q. Chen is a member of IEEE.

Tadahiro Ohmi served as a research associate in the Department of Electronics, Tokyo Institute of Technology, from 1966 until 1972.Then, he moved to Research Institute of Electrical Communication, Tohoku University and became an associate professor in 1976. In 1985 he became a professor at Department of Electronics, Faculty of Engineering, Tohoku University. Since 1998, he has been a professor at New Industry Creation Hatchery Center (NICHe), Tohoku University.

His research field covers whole Si-based semiconductor and flat panel display technologies in terms of material, process, device, circuit, and system technologies. He is known as an originator of "Ultra1clean Technology," which introduced ultraclean and scientific way of thinking into semiconductor manufacturing industry and became indispensable technology today. Dr. T. Ohmi is a fellow of IEEE and a fellow of the Institute of Electronics, Information and Communication Engineers of Japan.