VIDEO STREAM RETRIEVAL BASED ON TEMPORAL FEATURE OF FRAME DIFFERENCE

Mikito Toguro *Kenji Suzuki* *Pitoyo Hartono* *Shuji Hashimoto*

Dept. of Applied Physics, Waseda University
55N-4F-10A, 3-4-1 Okubo, Shinjuku-ku, Tokyo, Japan
{mit, kenji, hartono, shuji}@shalab.phys.waseda.ac.jp

ABSTRACT

In recent years, we can easily access an enormous amount of digital video stream in standardized video format such as MPEG and etc. However, it is not so easy to find a desired video stream from video database in reasonable short time. Some efficient searching methods in terms of computational cost are definitely required for video stream retrieval.

In this paper, we propose a novel method for video stream retrieval using a video stream as search-key. The objective is to find a part of the video stream stored in the database that is similar to the given video stream key. This method extracts some characteristics in frame transition in video streams, and utilizes the characteristic for the similarity matching. Some experimental results are also shown to verify the efficiency of the proposed method.

Keywords: video stream retrieval, frame transition, feature matrix, similarity matching

1. INTRODUCTION

Many works have been done on multimedia retrieval from database. At present several popular image searching methods use textual index as a searching key, like "Google Image Search"[1] and others[2][3]. But it will be difficult to apply the textual searching for a particular movement pattern in the stream that could not be easily indexed. The techniques of using the amount of the features of a picture are proposed [4][5][6][7]. In [8], a video retrieval system is proposed, in which each image stream in the database is divided into meaningful shots which are labeled with textual descriptions, audio dialogs and cinematic attributes. Chen and Chua proposed a method that considers similarity matching at shot and sequence levels [9]. Kashino et al. proposed a quick method based on adjacent frames similarity [10] with active search[11] which is an effective pruning algorithm.

The conventional usage of video stream retrieval is for the content searching. It is useful for finding contents similar to a query video stream from video database, for example, for finding some illegal contents on the internet, finding a particular spot commercial in TV broadcasting, or verifying the broadcast of commercial messages during TV programs. In such cases, content-based video stream retrieval is definitely required for efficient search.

One of the difficulties in video stream searching is caused by the data size. For example, in the case of the video stream with 320×240 frame size, 30 frames per second and quantized in 24 bit, the data size of the video stream for 10 seconds becomes approximately 3.8×10^{14} bits, which is not manageable for conventional retrieval algorithms in realistic search time.

In order to reduce the search space, some sophisticated methods to generate significant feature index have been proposed. Zhang et al. proposed a method[12] that parses temporally segment and abstracts a video source based on low-level image analyses, then the retrieval and browsing of video are based on key-frames selected during abstraction and spatial-temporal variations of visual features, as well as some shot-level semantics derived from camera operation and motion analysis. In addition to these methods, features that are unique to the video such as cuts, fades, dissolves and wipes are also used for the characteristic vectors [13]. The temporal information is often useful in characterizing a video stream; hence it could be added as one component of the feature vector. One example of the temporal characteristic of the video stream is the average brightness of the frame that is introduced successfully [14]. Another approach is object base retrieval in which the main attributes attached to key or dominant objects are motion, shape etc., as well as other image features such as color and texture [15].

We propose a novel method for video stream retrieval using a video stream as search-key. Most of the video streams are modeled with *n*-th (frame length) order Markov chain. In this sense, conventional characteristic of the still image, like color histogram of each image represents the distribution of zero-order Markov chain. In the pre-sent work, we focus on a particular temporal feature of video stream, which is the transition of frame difference.

As frame difference is useful for finding motion vectors or optical flows, the transition of frame difference represents a particular temporal pattern of video stream.

The sequence of frame difference can be used for the feature vector of each frame. The proposed method thus utilizes the features with the distribution of first order Markov chain. The advantage of the proposed method is in the usage of index data which size is significantly small and also useful for fast similarity matching, thus realizing fast retrieval, and also robust to the degradation of the original video stream. In the following sections, we first describe the feature extraction method, and then show some experimental results using artificially generated random image stream, television program, and also commercially available benchmark test video data. Discussion and conclusions are given in the final section.

2. FEATURE VECTOR AND SEARCHING METHOD

2.1. Temporal feature of frame difference

The feature vector represents temporal characteristics of the transition of frame difference. With regard to each frame in the video stream, we compute the total brightness difference $c(t)$ between the current frame $I(t)$ and the previous frame $I(t\text{-}1)$ as follows:

$$c(t) = \sum_{y=0}^{RGB} \sum_{x=0}^{h} \sum^{w} \left(I_{(RGB,x,y)}(t) - I_{(RGB,x,y)}(t-1) \right) \qquad (1)$$

where h and w represent the height and width of the image frame.

We then obtain $c^*(t)$ that is linearly quantized into C_{max} classes. The sequence $\{c^*(t)\}=\{c^*(t), c^*(t+1),..., c^*(t+\tau)\}$ is calculated from video stream for the fixed time interval $t \sim t+\tau$. By counting the number of transition within the certain time interval, the adjacency matrix $M(t)$ ($C_{max} \times C_{max}$) is computed from the transition from $c^*(t)$ to $c^*(t+\tau)$. This matrix represents the directed graph of the transition.

$$M(t) = \begin{pmatrix} m_{00} & \cdots & m_{0n} \\ \vdots & m_{ij} & \vdots \\ m_{n0} & \cdots & m_{nn} \end{pmatrix} \qquad (C_{max} = n) \qquad (2)$$

Where m_{ij} denotes the number of the transition from class i to class j within the fixed time span τ. An example of the operation is illustrated in Fig.1. When the sequence $\{c^*(t)\}=\{0,1,2,1,2,1,0\}$ ($C_{max}=3$) is given, the adjacency matrix is obtained as follows:

$$M(t) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix} \qquad (3)$$

We then compute the vector notation $V(t)$ of the adjacency matrix that is used for the feature vector of each frame.

$$V(t) = (m_{00}, m_{0n}, m_{10}, \ldots\ldots, m_{ij}, \ldots, m_{n0}, \ldots, m_{nn}) \qquad (4)$$

2.2. Searching mechanism

In the searching process, the feature vector is used for similarity matching with a given video stream as query that is regarded as key clip. The similarity score $s(t)$ is computed by inner product of feature vectors between key clip and reference video stream. $s(t)$ normalized by $|V(t)||V_k|$ is described as follows:

$$S(t) = V(t) \bullet V_k \Big/ |V(t)||V_k| \qquad (5)$$

$V(t)$ represents the feature vector of a frame at time t in video stream. V_k represents the feature vector of key clip. Let L the total number of frames in video stream, and let l the number of frames in the key clip. The searching system computes the similarity by shifting from t=0 to t=T where $T = L\text{-}l$. The matching result gives the ordered video stream at time t with the highest similarity. This method aims at utilizing a temporal feature of video sequence without caring about particular image features of each frame. This method is expected to provide the robustness in terms of the quality of video stream, i.e. different bit rate or noisy video stream.

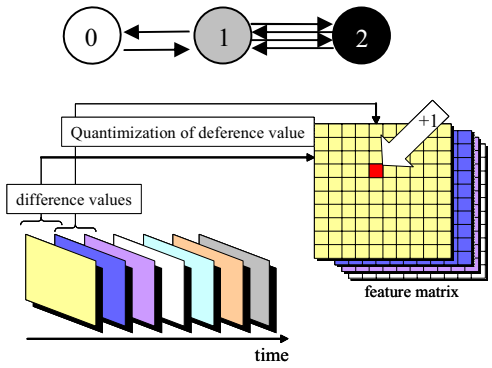

Fig. 1: Overview of the proposed feature vector extraction

3. EXPERIMENTS

3.1. Experiment with random image sequence

In order to verify that the proposed method effectively performs, we did an experiment with an artificial video stream. In this experiment, an artificial video stream is generated by 10,000 images of which the brightness of each pixel is set to random value. For the searching, 30 frames from t=5000 in the video stream are used for key clip. The quantization ratio C_{max} is empirically set to 10.

Fig.2 shows the similarity score $s(t)$ along the time t. The bold arrow shows the position of key clip. It can be seen that the similarity is relatively small except for the part of key clip. Because the video stream does not have

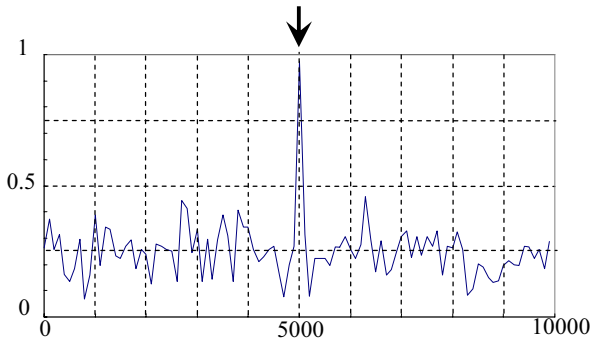

Fig. 2: Experimental result with an artificial video stream consisting of random images.

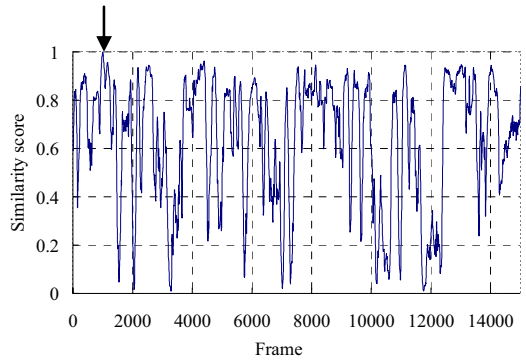

Fig.3 : Experimental result with a NIST video stream

any correlation between frames, similarity score marks the highest one at the position of key clip.

3.2. Experiment with a real video sequence

NIST Digital Video data [16], which are often used as benchmark test data, are used in this experiment. As for the preprocessing, the feature vectors are computed from the video stream encoded with MPEG format. The key clip consists of 120 frames from t=1,000, which is a part of the "ahf1.mpg" in NIST video collection volume 1.

The similarity score along the time is shown in Fig. 3. The bold arrow shows the position of the key clip. Even the highest similarity score marks at the key clip, many other peaks with higher scores can also be found at other frames. The feature vector contains the direction of transition of frame difference, whereas it does not have the order of the transition. A number of video streams with the higher similarity scores are likely to come out.

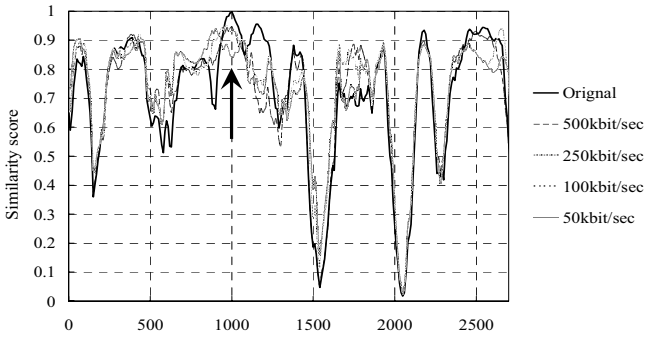

Fig. 4: Similarity values with different bit rate

3.3. Experiment with different encoding rate

In order to verify the robustness, we applied our searching method for video streams with different encoding rates.

We created four video streams with different bit rate; 500kbit/s, 250kbit/s, 100kbit/s and 50kbit/s, which are re-encoded from the original MPEG video stream of NIST Digital Video data encoded with 1.3 M bit/s. The key clip consists of 120 frames from t=1,000, which is extracted from the original video stream. Fig.4 shows the comparison of similarity values among four video streams with different bit rates. The bold arrow represents the position of the key clip. The position of peaks does not change by the difference of bit rate, whereas the average of the similarity score is relatively small at the video stream with higher bit rate. This result shows that searching method is not strongly dependant on the bit rate.

In the conventional audio and video compression, not all of the data is preserved. For example, in lossy compression algorithm based on DCT, the quantization is weighted low elements in the frequency domain. Low frequency elements contribute to rough quantization, while high frequency elements contribute to fine quantization.

As the proposed retrieval algorithm utilizes the feature vector based on the mean value of brightness in the frame difference, these features corresponds to the lowest frequency element. Therefore, the proposed retrieval system shows robust to the conventional video compression.

Table 1 shows the statistics of the similarity scores for different bit rate streams.. The maximum of the similarity score pointed to the key clip in all the bit rates. We can

Table 1: Mean value and variance of similarity score

	Original	500kbit/sec	250kbit/sec	100kbit/sec	50kbit/sec
Mean value of Similarity Score	0.691524	0.701615	0.730465	0.726119	0.730097
Variance of Similarity Score	0.049671	0.043107	0.034765	0.030817	0.029955
* Maximum of Similarity Score	0.999116	0.95354	0.949215	0.938766	0.938766

Table 2: Processing time

Total processing time	37 min.
(MPEG decoding)	33 min.
(feature extraction)	4 min. 36 sec.
Size of index data	23,676 [Byte]
Searching time	2 [sec]

observe that there are significant gaps between the maximums and mean values, which mean that false positive can be distinguished from the desired stream. The proposed algorithm also achieved the stable mean and variance of similarity score regardless to the bit rate of the video stream. Moreover, the similarity score does not change significantly with regard to the alternation of the bit rate.

3.4. Processing time

The computational cost for each processing is shown Table 2. Although conversion from MPEG to PNM format is required for the searching program, 88% of processing time is concerning with MPEG decoding. On the other hand, the required time for the processing of feature extraction is about 3min, and for searching a desired video clip in 8min. 45sec. video stream is about 2 seconds. The required time for both feature extraction and searching is reasonable for practical use.

4. CONCLUSION

We proposed a novel searching method of content-based video stream retrieval, focusing on temporal feature of frame difference. We also implemented the proposed searching engine with the www-based video stream retrieval system. Although the feature vector has merely limited to small, searching performs well for video stream retrieval. Experimental results with both the artificial and real video stream prove that the proposed method is capable of retrieving a desired video clip in a reasonable time. Moreover, meaningful contents can also be extracted.

The parameter C_{max} is a critical factor in balancing the retrieval accuracy and speed. Although this is given experientially now, it considers this that a future examination is required.

We utilized the difference image for the feature vector in the present work, whereas using motion vectors is our further consideration. The classification of dynamic features from motion prediction vector or the change of bit rate of MPEG video stream will be also included in the future issues.

Acknowledgement

This work was supported in part by The 21st Century Center of Excellence Program, "The innovative research on symbiosis technologies for human and robots in the elderly dominated society," Waseda University.

5. REFERENCES

[1] Google Image Search, http://www.google.com/imghp

[2] L. A. Rowe, J.S. Boreczky, C. A. Eads, “Indexes for User Access to Large Video Databases,” Storage and Retrieval for Image and Video Databases II, The International Society for Optical Engineering, Vol 2185, pp.150-161, 1994.

[3] C.Federighi L.A. Rowe, ”A Distributed Hierarchical Storage Manager for a Video-on-Demand System,” Storage and Retrieval for Image and Video Databases II, The International Society for Optical Engineering, Vol.2185, pp 185-197,1994.

[4] M.Flickner, H.Sawhney, W.Niblack, J.Ashley, Qi.Huang, B.Dom, M.Gorkani, J.Hafner, D.Lee, D.Petkovic, D.Steele, P.Yanker, “Query by Image and Video Content: The QBIC System,” IEEE Computer, vol.28 , no.9, pp.23-32, 1995.

[5] T.Okamura, et al, “Construction of a Flower Image Database with Feature and Index-based Searching Mechanism,” 5th International Workshop on Image Analysis for Multimedia Interactive Services, 2004.

[6] A. Hamrapur, A. Gupta, B. Horowitz, C.F. Shu, C. Fuller, J. Bach, M. Gorkani, R. Jain, “Virage Video Engine,” SPIE Proceedings on Storage and Retrieval for Image and Video Databases V, pp.188-97,1997.

[7] S.Chang, et al, ”An automated content based video search system using visual cues,” Proc. of the fifth ACM Intl. Conf. .on Multimedia, pp.313-324, 1997.

[8] Tat-Seng Chua, Liqun. Ruan. “A video retrieval and sequencing system,” ACM Trans. of Information Systems. 13(4), pp. 373-407, 1995.

[9] L.Chen, Tat-Sheng Chua, ”A match and tiling approach to content-based video retrieval,” Proc. of IEEE International Conference on Multimedia and Expo 2001, pp.301 – 304, 2001.

[10] K.Kashino, T.Kurozumi, H.Murase, “A quick search method for audio and video signals based on histogram pruning,” IEEE Trans. on Multimedia, Volume: 5 , Issue: 3 , pp.348-357, 2003.

[11] A.Kimura, K.Kashino, T.Kurozumi, H.Murase, “Very quick audio searching: introducing global pruning to the Time-Series Active Search,” Proc. of ICASSP 2001, Vol.3, pp.1429 – 1432, 2001.

[12]S.Takagi, et al, "Implementation of Video Searching System Using Videoprint," Tech. Rep. IEICE, CS2001-104, pp. 19-24, 2001.

[13] H.J. Zhang, C.Y. Low, S.W. Smoliar and J.H. Wu, “Video parsing, retrieval and browsing: an integrated and content-based solution,” Proc. ACM Multimedia’95, pp.15-24, 1995.

[14] R. Zabih, et al, “A feature-based algorithm for detecting and classifying scene breaks,” Proc. ACM Multimedia '95, pp. 189-200, 1995.

[15] H.J. Zhang, et al, “Video parsing, retrieval and browsing: an integrated and content-based solution,” Proc. ACM Multimedia'95, pp.15-24, 1995.

[16] [online] "NIST Digital Video Collection Vol. 1”, Available at http://www.nist.gov/srd/nistsd26.htm