ADAPTIVE DIVISION OF FEATURE SPACE FOR RAPID DETECTION OF NEAR-DUPLICATE VIDEO SEGMENTS

Ichiro Ide[‡ *], *Shugo Suzuki*[‡ †], *Tomokazu Takahashi*[††], *Hiroshi Murase*[‡]

‡ Nagoya University
Graduate School of Information Science
Furo-cho, Chikusa-ku, Nagoya, 464-8601, Japan
E-mail: {ide,murase}@is.nagoya-u.ac.jp

†† Gifu Shotoku Gakuen University
Faculty of Economics and Information
1-38 Naka Uzura, Gifu, 500-8288, Japan
E-mail: ttakahashi@gifu.shotoku.ac.jp

ABSTRACT

Near-duplicate video detection is becoming a core-technology for analyzing the structure of a large-scale video archive. It, however, is naturally an $O(n^2)$ problem, where n is a value proportional to the total length of an input video stream. We have previously challenged this time-consuming task by reducing the cost required for each of the $O(n^2)$ comparisons. This paper, on the other hand, proposes a method that reduces the number of comparisons by adaptively dividing the feature space according to the distribution of feature points.

Index Terms— Near-duplicate video detection, feature space division, video structuring

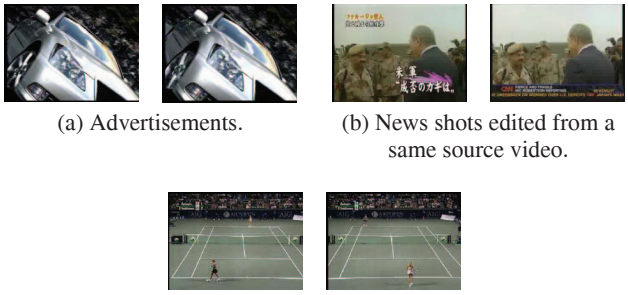
1. INTRODUCTION

Recent advance in digital storage technologies has enabled us to store a huge amount of video data as an online archive. Near-duplicate video segment detection is a task that retrieves all pairs of nearly identical video segments from a given video stream. To analyze the structure of a large-scale video archive, fast near-duplicate video detection could be considered a core-technology. For example, Naturel et al. made use of this technology to align broadcast video streams to Electronic Program Guide data [1], and we made use of it for cross-channel/lingual retrieval of news stories [2]. Several other groups also made use of it for news story tracking [3, 4].

This task is, however, extremely time-consuming since it requires ${}_n\mathrm{C}_2$ times of comparison of video features, where n is the number of video fragments derived from an input video stream. In order to efficiently process this task, we have proposed a two-step method that first roughly compares the ${}_n\mathrm{C}_2$ pairs of video fragments in a low-dimension feature space, and then precisely compares only the pairs detected in the first step [5]. This method drastically reduces the processing time, but it still requires ${}_n\mathrm{C}_2$ times of comparison between video fragments.

To further improve the efficiency of the process, this paper proposes a method that reduces the times of comparisons by hierarchically dividing the feature space and comparing all pairs of feature points only within each space. Simply applying this operation, however, results in overlooking pairs near the boundary of the divided feature spaces. In order to detect such pairs, both feature spaces are extended at the boundary by adding a margin space that contains feature points overlapping both spaces. Since this may result in producing corpulent feature spaces after the division, typically when the feature points are densely distributed near the boundary, we set a condition that stops the hierarchical division process based on the number of feature points in the margin space.

Note that the proposed method and its original work in [5] output all pairs of near-duplicate video segments that satisfy the definition described in 2.1. Since the quality of the detected results depends on the application, we will not discuss the accuracy in this paper.



(a) Advertisements. (b) News shots edited from a same source video.

(c) Sports scenes shot from a same camera.

Fig. 1. Examples of near-duplicate video segments.

2. NEAR-DUPLICATE VIDEO SEGMENT DETECTION

2.1. Definition

A near-duplicate video segment is a video segment that is nearly identical to another video segment in a video stream. As shown in Figure 1, near-duplicate video segments in broadcast video include advertisements, slightly modified news shots edited from a common video source, scenes from a fixed camera in sports programs, etc.

In order to detect near-duplicate video segments with an arbitrary length at an arbitrary position in a video stream, we consider a fixed length video fragment as the minimum unit. As shown in Figure 2, a pair of near-duplicate video fragments is defined as follows:

- Each video fragment is represented as a point in a feature space.
- A pair of near-duplicate video fragments is a pair of points in the feature space that exist within an Euclidean distance of ϵ.

2.2. Previous works

There are some works that aim at detecting near-duplicate video segments fast by an approximate approach such as those by Sekimoto et

*Also affiliated to the National Institute of Informatics, Tokyo, Japan.
†Currently at Toyota Motor Corporation, Aichi, Japan.

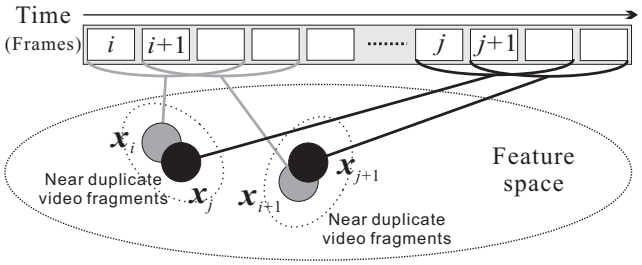


Fig. 2. The detection of near-duplicate video fragments in a feature space. A pair of video fragments are considered near-duplicate when $\|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon$. A sequence of pairs of near-duplicate video fragments compose a pair of near-duplicate video segments.

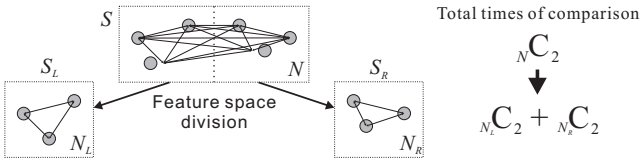


Fig. 3. The reduction of times of comparison by dividing the feature space S into sub-feature spaces S_L and S_R; ${}_{N_L}C_2 + {}_{N_R}C_2 \leq {}_NC_2$ when $N = N_L + N_R$.

al. [6], Yamagishi et al. [7], and Naturel and Gros [1]. On the other hand, we have been proposing a fast detection method that guarantees the detection of all near-duplicate video segments in video data. A rough overview of the framework of our approach is as follows:

i. Choose the bases for feature points representation
Principal component analysis is applied to a sufficiently long video stream that could be considered to represent the nature of general broadcast video. Eigen vectors corresponding to the D largest eigen values are chosen as bases for a D-dimension feature space.

ii. Detect near-duplicate video fragments

(a) Candidate detection in the low-dimension feature space
Video features projected onto the D-dimension feature space are compared as low-dimension vectors, which makes each comparison fast. No pair of near-duplicate video fragments are overlooked at this step as long as ϵ is fixed, due to the nature of Euclidean distance.

(b) Precise detection in the original feature space
Only the candidates detected in Step ii.-(a) are compared as the original high-dimension vectors to check if they are truly near-duplicate or not.

For details of the framework, see [5]. Note that in this paper we focus on reducing the processing time for the rough comparison in the low-dimension feature space (Step ii.-(a)), and thus, the overall detection speed of the framework will not be discussed here.

3. ADAPTIVE DIVISION OF FEATURE SPACE

3.1. Hierarchical division

In order to reduce the times of comparison, we take a hierarchical feature space division approach. As shown in Figure 3, if a feature space is divided in two sub-feature spaces, the total times of feature points comparison is reduced. This division operation could be applied recursively to obtain further reduction. As a result, if a feature space is divided into I sub-feature spaces $S_i (i = 1, ..., I)$ which contain N_i feature points, respectively, the total times of feature points comparison $\sum_{i=1}^{I} {}_{N_i}C_2$ is always smaller than the original times of comparison ${}_NC_2$ where $N = \sum_{i=1}^{I} N_i$.

Next comes the problem on how to divide the feature spaces. To maximize the reduction of the times of feature points comparison at each division operation, it is necessary to divide a feature space so that the divided spaces contain an equal number of feature points.

The following division algorithm is applied recursively to the feature space:

1. Let S be the original feature space with N feature points $\{\mathbf{x}_n | n = 1, ..., N\}$.
2. Select a basis $\mathbf{v}$ and project all the feature points onto it.
3. Sort the projected points $x_{n\mathbf{v}}$ on the basis, and select their median $m_{\mathbf{v}_i}$.
4. Divide the feature points on each side of $m_{\mathbf{v}}$ into two sub-feature spaces S_L and S_R (See Figure 4).

Selection of the basis $\mathbf{v}$ in Step 2., and the termination condition of the recursive process are explained in 3.2.2.

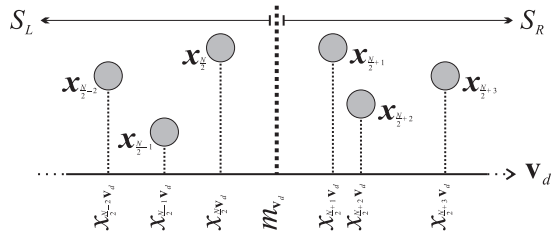


Fig. 4. The division of feature space by projection of feature points on a basis $\mathbf{v}$. This case shows an example when N is an odd number.

3.2. Adaptive division with an overlapped margin space

3.2.1. Feature space division with an overlapped margin space

It may seem that the hierarchical feature space division is the solution to our task. It, however, has a serious problem that it cannot detect pairs of feature points even within the range of ϵ when they are divided by the division boundary.

In order to compensate for this problem, the following operation is added after Step 4. in the algorithm in Sect. 3.1.

5. Set a margin space S_m with a width of $\frac{\epsilon}{2}$ on both sides of the boundary, and extend the sub-feature spaces on both sides so that all the feature points in the margin space belong to both sub-feature spaces; $S_{L'} = S_L + S_{m_R}, S_{R'} = S_R + S_{m_L}$ (See Figure 5). In other words, the margin space $S_m = S_{m_L} + S_{m_R}$ exists as an overlapped space of both sub-feature spaces.

Although the margin space assures the detection of all the pairs that should be detected even after the feature space division, it necessarily increases the number of feature points in each sub-feature space from N_L, N_R to $N_{L'} = N_L + N_{m_R}, N_{R'} = N_R + N_{m_L}$, correspondingly, where N_i indicates the number of feature points in a space S_i. This leads to an increase in the times of comparison, which becomes intolerable when $N_{L'} + N_{R'}$ exceeds N.

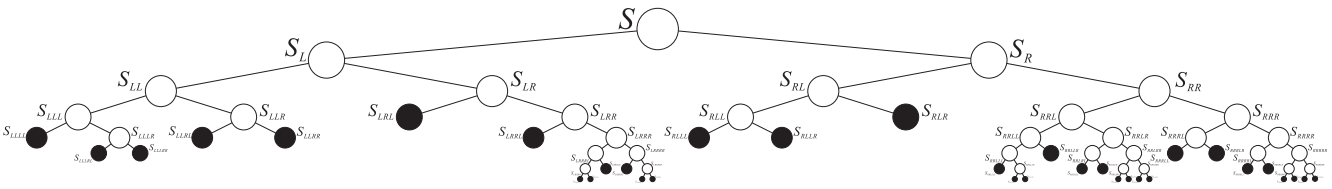


Fig. 6. An example of a division tree of a feature space created by the proposed method. The feature points comparison process is performed only in the black sub-feature spaces. The tree is asynchronous since it reflects the distribution of feature points in the feature space.

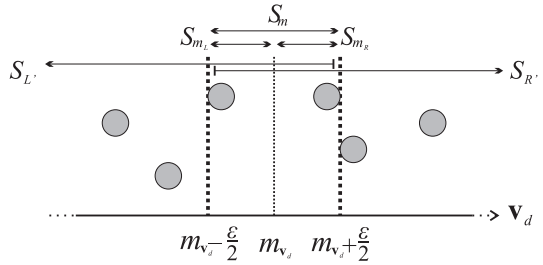


Fig. 5. The margin space set on both sides of the division boundary assures the detection of all pairs of feature points within the distance ϵ even in the sub-feature spaces.

As a consequence, the following function is tested in order to avoid such prohibitive feature space division operations.

$$f(S, \mathbf{v}_d) = {}_N\mathrm{C}_2 - \left({}_{\frac{N}{2}+N_{m_L}}\mathrm{C}_2 + {}_{\frac{N}{2}+N_{m_R}}\mathrm{C}_2 \right) \quad (1)$$

If $f(S, \mathbf{v}_d)$ is negative, division of the feature space by projection to a basis $\mathbf{v}$ should be avoided.

3.2.2. Selection of the projection basis and the termination condition

The projection basis $\mathbf{v}$ is selected so that it satisfies the following condition:

$$\mathbf{v} = \arg \max_{\mathbf{v}_d (d=1,\ldots,D)} f(S, \mathbf{v}_d), \quad (2)$$

where D is the dimension of the feature space. This strategy is taken since the larger $f(S, \mathbf{v}_d)$ may be, the more effective the division will be. Note that the method allows selecting the same basis over and over as long as it could be considered as the most effective one.

If the testing function $f(S, \mathbf{v})$ even for the selected basis $\mathbf{v}$ is negative, further division is terminated there. As a result, an asynchronous feature space division tree as shown in Figure 6 is obtained, which roughly reflects the feature points distribution.

4. EXPERIMENTS

In order to evaluate the effectiveness of the proposed feature space division method (abbreviated as 'FSD' in the graphs), we applied it to actual broadcast video data and observed the reduction of the times of feature points comparison compared to that in the original feature space.

Table 1. The dataset used in the experiments.

Video genre	Total length	Contents
General	24 hours	NHK all programs × 1 day
News	3 hours	NHK 'News 7' × 6 days
Sports	3 hours	1 game of a football match
Comedy	3 hours	3 comedy shows
Format	MPEG-1 (320×240 pixels)	
Frame rate	30 Frames per second	

Table 2. The feature representation of video fragments (feature points).

Parameter	Value
Length of a fragment	5 seconds (150 frames)
Feature vector size (D)	20 dimensions[1]

The dataset used for the experiments, and the feature representation of video fragments were as shown in Tables 1 and 2, respectively. The bases used for the feature space division were those from the 20-dimension compressed video feature space.

4.1. Evaluation by general broadcast video

In order to evaluate the general effectiveness of the proposed method, we first applied it to general video data including various genres of television shows. Figure 7 shows that the proposed method drastically reduces the times of comparison, where the effectiveness increases in proportion to the video length.

Figure 8 shows the overhead for the feature space division operation against the processing time for the comparison process. Although the ratio of the overhead to the total processing time increases, we can deduce from the results in Fig. 7 that it could be ignored considering the drastic reduction of the times of feature points comparison achieved. This could be explained by the fact that the division operation requires an average of $O(n \log n)$ times of comparisons for the sorting of feature points projected on the bases, which increases much slower with n than the $O(n^2)$ comparison process.

4.2. Evaluation by specific genres of broadcast video

Since the proposed method divides a feature space adaptive to the distribution of feature points, it could easily be imagined that its effectiveness may differ among different genres of video data. Figure 9

[1] See [5] for details of the video feature representation and compression for near-duplicate video segments detection.

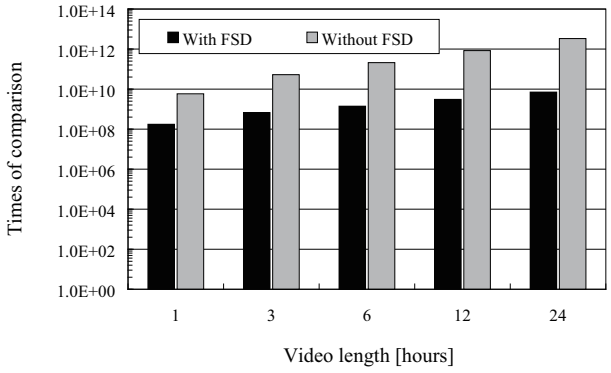


Fig. 7. Reduction of times of feature point comparison in log-scale (General broadcast video).

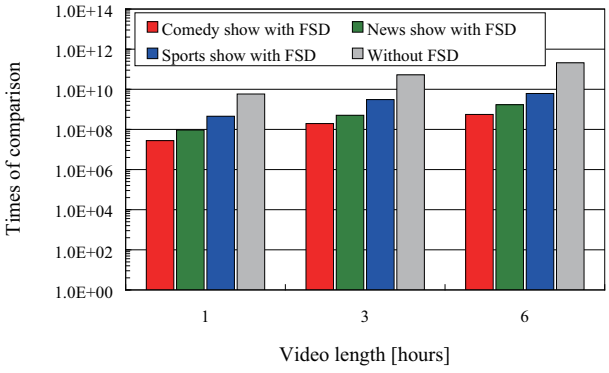


Fig. 9. Reduction of times of feature point comparison in log-scale (By different genres of broadcast video).

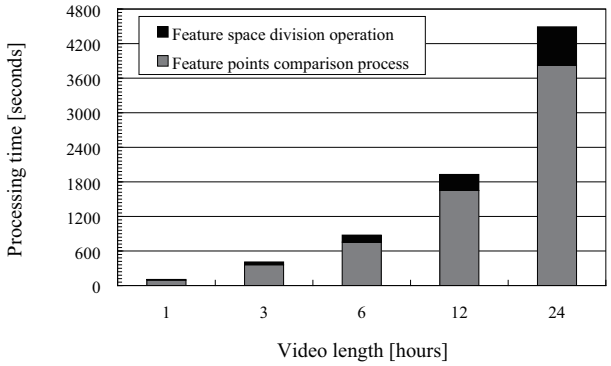


Fig. 8. Overhead of the feature space division operation against the feature points comparison process (General broadcast video).

shows the times of comparison for three different genres of broadcast video; news, sports (football), and comedy.

The difference between the genres could be explained by the general tendencies that a sports show tends to be composed of similar scenes shot from fixed cameras, where a comedy show tends to be composed of various scenes. News shows are in between the two genres since it is composed of a fixed studio scene and various scenes from news sites. Such differences in the distribution of feature points affect the effectiveness of the proposed method since where the feature points are dense, the feature division terminates in an earlier stage compared to where they are sparse.

5. CONCLUSION

This paper introduced and evaluated a hierarchical feature space division method aimed to reduce the $O(n^2)$ times of comparison between feature points needed to detect all near-duplicate video segments in a video stream. Results of the experiments showed the effectiveness of the proposed method, together with its tendencies depending on the genres of video data.

Future works include the consideration of better division boundaries that minimizes the number of feature points in a margin space, and the overall evaluation of the near-duplicate detection framework.

6. ACKNOWLEDGEMENTS

Parts of this work were supported by the Grants-in-Aid for Scientific Research from the Japanese Ministry of Education, Culture, Sports, Science and Technology. The proposed method was implemented using the MIST library, distributed at http://mist.suenaga.m.is.nagoya-u.ac.jp/trac-en/.

7. REFERENCES

[1] Xavier Naturel and Patrick Gros, "A fast shot matching strategy for detecting duplicate sequences in a television stream," in *Proc. 2nd Int. Workshop on Computer Vision meets Databases*, June 2005, pp. 21–27.

[2] Akira Ogawa, Tomokazu Takahashi, Ichiro Ide, and Hiroshi Murase, "Cross-lingual retrieval of identical news events using image information," in *Advances in Multimedia Modeling, 14th Int. Multimedia Modeling Conf., MMM2008 Procs.*, Shin'ichi Satoh, Frank Nack, and Minoru Etoh, Eds. Jan. 2008, vol. 4903 of *Lecture Notes in Computer Science*, pp. 287–296, Springer-Verlag.

[3] Pinar Duygulu, Jia-Yu Pan, and David A. Forsyth, "Towards auto-documentary: Tracking the evolution of news stories," in *Proc. 12th ACM Int. Conf. on Multimedia*, Oct. 2004, pp. 820–827.

[4] Yun Zhai and Mubarak Shah, "Tracking news stories across different sources," in *Proc. 13th ACM Int. Conf. on Multimedia*, Nov. 2005, pp. 2–10.

[5] Ichiro Ide, Kazuhiro Noda, Tomokazu Takahashi, and Hiroshi Murase, "Genre-adaptive near-duplicate video segment detection," in *Proc. 2007 IEEE Int. Conf. on Multimedia and Expo*, July 2007, pp. 484–487.

[6] Nobuhiro Sekimoto, Takuichi Nishimura, Hironobu Takahashi, and Ryuichi Oka, "Continuous retrieval of video using segmentation-free query," in *Proc. IAPR 15th Int. Conf. on Pattern Recognition*, Sept. 2000, pp. 375–378.

[7] Fuminori Yamagishi, Shin'ichi Satoh, Takashi Hamada, and Masao Sakauchi, "Identical video segment detection for large-scale broadcast video archives," in *Proc. 3rd Int. Workshop on Content-Based Multimedia Indexing*, Sept. 2003, pp. 135–141.