DYNAMIC-SEGMENTATION-BASED FEATURE DIMENSION REDUCTION FOR QUICK AUDIO/VIDEO SEARCHING

Akisato Kimura, Kunio Kashino, Takayuki Kurozumi and Hiroshi Murase

NTT Communication Science Laboratories, NTT Corporation
3-1, Morinosato-Wakamiya, Atsugi-shi, Kanagawa, 243-0198, Japan
E-Mail : {akisato, kunio, kurozumi, murase}@eye.brl.ntt.co.jp

ABSTRACT

We propose a new feature dimension reduction method for multimedia search. The main technique in the method is dynamic segmentation that partitions sequential feature trajectories dynamically. While dynamic segmentation reduces the average dimensionality and accelerates the search, it requires huge amount of calculation. Thus, our method quickly executes suboptimal partitioning of the trajectories by using the discreteness of dimension changes. This guarantees the optimal amount of calculation to derive the suboptimal partitioning under the condition that the dimension monotonously increases as the segment length increases. The experiment shows that our method is over 10 times faster than a straightforward dynamic segmentation method.

1. INTRODUCTION

This paper discusses feature dimension reduction for a quick audio/video search. One of the applications we have specifically in mind is to detect and locate a known audio or video signal (a *reference signal* or a *query*) in a long multimedia signal stream (a *stored signal* or a *database*) based on signal similarity. We call this audio/video search. A major research issue in this approach is speed. Specifically, features for audio/video signals tend to be high-dimensional, which is not necessarily suitable for the various tree-search algorithms developed in the database field [1, 2].

In coping with the high-dimensionality problem, it is natural to think of dimension reduction. Previously, we proposed a quick and accurate search algorithm for multimedia signals based on dimension reduction [3]. The main techniques in the algorithm are piecewise linear representation of sequential feature trajectories (called segment-based PCA) and efficient pruning of the search space (called distance bounding). In the dimension reduction technique, segment-based PCA was carried out by dividing trajectories into equal-length segments and doing KL transform in every segment. However, it is expected that allowing the segments to have variable lengths would improve dimension reduction performance.

Here, we introduce dynamic segmentation. Dynamic segmentation refers to partitioning feature trajectories dynamically so as to minimize the average dimensionality. However, finding optimal partitioning requires a huge amount of calculation (e.g. [4]). Thus, our technique addresses quick suboptimal partitioning of the trajectories by modifying the formulation of dynamic segmentation and using the discreteness of dimension changes. It also achieves theoretical optimality in the amount of calculation to derive the suboptimal partitioning under the condition that the dimension monotonously increases as the segment length increases.

Most of the related works also utilize a suboptimal approach. For example, Keogh *et al.* [5] used a bottom-up merging of segments. Wang *et al.* [6] tested a line fitting approach, namely, finding the longest line segment such that the approximation error does not exceed the given threshold. Keogh *et al.* [7] also reported a conversion of the problem into a wavelet decomposition problem. Although these methods are useful for low-dimensional time-series, such as fluctuation of stock prices and seismic data, it is not straightforward to apply them to high-dimensional trajectories.

This paper is organized as follows: Section 2 overviews the search algorithm and our previous method [3]. Section 3 explains the dimension reduction method, the core part of our new algorithm. Section 4 evaluates the performance of the algorithm using a recording of real TV broadcasting. Finally, Section 5 gives conclusions.

2. SEARCH ALGORITHM

Fig. 1 outlines the search method [3]. This method is based on Time-series Active Search, which we proposed earlier [8].

In the preparation stage, the basic features are calculated from the stored signal. For example, sets of normalized short-time power spectra are used as features for audio signals, and sets of average RGB values in subimages are used as video features. The basic features are then quantized by using a vector quantization (VQ) algorithm. Then, the windows are applied to the basic feature sequence, and histograms are created by counting the number of the basic features over the window for each VQ codeword. The histograms are compressed by the dimension reduction method as described later. The compressed features, the final form

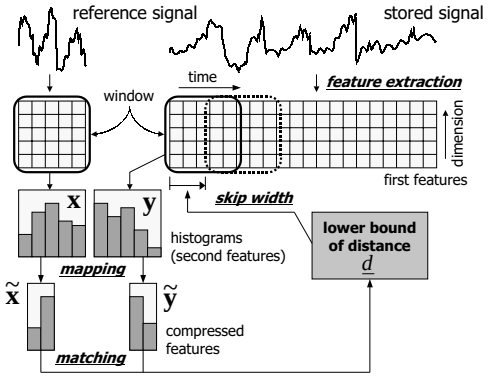

Fig. 1. Overview of the search method

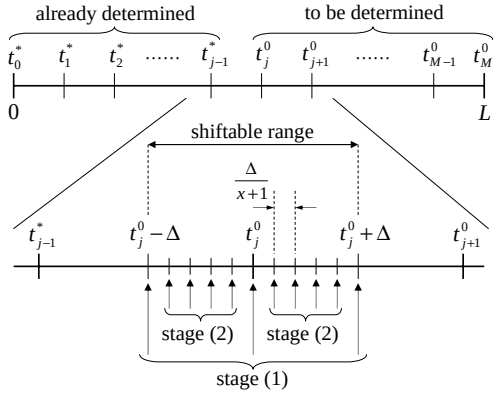

Fig. 2. Overview of Dynamic Segmentation

of the signal features, comprise the mapped histogram and the mapping distance. The mapping distance means the distance between the corresponding histograms before and after the compression.

In the search stage, a compressed feature is created from the reference signal in the same way that one is created from the stored signal. Next, the compressed reference feature and the stored one corresponding to the matching point are matched. In the matching, the distance between these compressed features is calculated. Although the distance measure can be determined in several ways, here we use L_2-*distance* (*Euclid distance*) $d(\cdot,\cdot)$. The distance between compressed features, $\boldsymbol{y}_1, \boldsymbol{y}_2$, gives a lower bound of the distance between the original histograms, $\boldsymbol{x}_1, \boldsymbol{x}_2$:

$$\begin{aligned}\{d_2(\boldsymbol{y}_1,\boldsymbol{y}_2)\}^2 &= \{d_2(g(\boldsymbol{x}_1),g(\boldsymbol{x}_2))\}^2 \\ &\quad +\{d_2(\boldsymbol{x}_1,g(\boldsymbol{x}_1))-d_2(\boldsymbol{x}_2,g(\boldsymbol{x}_2))\}^2 \\ &= \min\{d_2(\boldsymbol{x}_1,\boldsymbol{x}_2)\}^2,\end{aligned}$$

where $g(\cdot)$ is a function for dimension reduction, and the minimum is taken over all $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ given $g(\boldsymbol{x}_1)$, $g(\boldsymbol{x}_2)$, $d_2(\boldsymbol{x}_1, g(\boldsymbol{x}_1))$ and $d_2(\boldsymbol{x}_2, g(\boldsymbol{x}_2))$. If the distance between compressed features falls below a given value (a *search threshold*), then the high-dimensional original histograms are matched. If the distance between the original histograms falls below the search threshold again, the reference signal is determined to be detected. In the last step, the window on the stored signal is shifted forward in time and the search proceeds.

When the distance is calculated for one segment, the feature matching for the following segments can be skipped if the lower bound of the distance, which can be calculated from the current segment distance, is not smaller than the search threshold. The use of the lower bound guarantees that no segment to be detected is missed by the skipping. The skip width w is given by

$$w = \begin{cases} \lfloor \sqrt{2}D(d_2-\theta_1) \rfloor + 1 & (\text{if } d_2 > \theta_1) \\ 1. & (\text{otherwise}) \end{cases}$$

L	the total number of frames contained in the stored signal
M	the number of segments
σ	the given contribution rate
t_j	the frame number on the right boundary of the j-th segment $(j = 1, 2, \cdots, M)$
t_j^0	the initial value of t_j
t_j^*	the value of t_j to be found
Δ	the width of shiftable range of boundary
$c(t_i, t_j, \sigma)$	the minimum number of components on the segment (t_i, t_j) such that the contribution rate exceeds σ

Table 1. Notations

where $\lfloor x \rfloor$ means the greatest integer less than x, $d_2 = d_2(\boldsymbol{y}_1, \boldsymbol{y}_2)$, and θ_1 is the search threshold.

3. DIMENSION REDUCTION METHOD

3.1. Outline

Here we note that histogram sequence is "continuous" by nature in the histogram space, and introduce the piecewise linear representation of the sequential histogram trajectory with variable length of segments.

Firstly, a trajectory is divided into a certain number (e.g. 1000) of equal-length segments. Next, dynamic segmentation is performed. It determines the segment boundaries, given a shiftable range of boundaries, so as to minimize the average dimensionality per frame. Then, KL transform is performed for each segment, and functions for the dimension reduction are determined with the minimal number of components such that the contribution rate exceeds a certain predetermined value. Lastly, histograms from the stored signal are transformed by their corresponding function.

3.2. Formulation of Dynamic Segmentation

Refer to Table 1 for mathematical notations. Dynamic segmentation can be formulated by using the framework of Dynamic Programming (DP). DP enables us to obtain the optimal boundaries. However in this case, DP is not practical

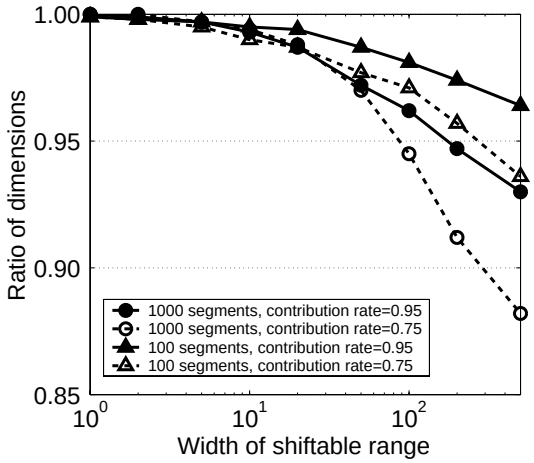

Fig. 3. Dimension reduction and width of shiftable range

because it requires a huge amount of calculation, yet less than naive dynamic segmentation. The total number of calculations is evaluated as follows:

$$(M-2)(2\Delta+1)^2 + 2(2\Delta+1) = \mathcal{O}(M\Delta^2).$$

Therefore, we formulate dynamic segmentation in another way so as to get boundaries with less calculation. The position of the boundary to be found, t_j^*, is obtained by using the following forward recursion.

$$t_j^* = \arg \min_{t_j : t_j^0 - \Delta \le t_j \le t_j^0 + \Delta} \left\{ \frac{c(t_{j-1}^*, t_j, \sigma)}{t_j - t_{j-1}^*} + \frac{c(t_j, t_{j+1}^0, \sigma)}{t_{j+1}^0 - t_j} \right\},$$

where

$$t_0^* = 0,\ t_M^* = L,$$
$$t_j^0 = \left\lceil j\frac{L}{M} \right\rceil \ (j = 0, 1, \cdots, M-1),\ t_M^0 = L,$$

and $\lceil x \rceil$ means the smallest integer greater than x (See Fig. 2). This formulation means that the positions of boundaries are determined in order of time. In this formulation, the total number of calculations is evaluated as

$$2(M-1)(2\Delta+1) = \mathcal{O}(M\Delta).$$

We measured the relationship between the degree of dimension reduction and the width of the shiftable range. In the experiment, we used histograms with 256 dimensions created from a video recording of a TV broadcast. The results are shown in Fig. 3, where the horizontal axis is the width of the shiftable range of the boundary and the vertical axis expresses the ratio of the average number of dimensions of the mapped histograms to the one without dynamic segmentation. The contribution and segment setting are shown in the inset. The data indicate that the average dimensionality of the features monotonically decreases as the width of the shiftable range increases. For example, the average number of dimensions decreases to 88.2% when the contribution rate is 0.75 and the number of segments is 1000.

x	half of the number of calculations in the stage 2
$f(x)$	the total number of calculations given x
K	the number of portions where the change of dimensionality occurs

Table 2. Notations

3.3. Speed-up Method

The formulation in 3.2 gives suboptimal segmentation of feature trajectory in the sense of dimensionality. However, this formulation still requires large amount of computations, yet less than DP. Here, we note that the change of dimensionality is continuous but discrete. Therefore, the optimal positions of segment boundaries must exist at the changing points of dimensions or at the edges of the shiftable range.

From the above discussion, we propose a speed-up method for dynamic segmentation based on a coarse-to-fine approach. First, dimensions of the segments are calculated at the initial position and at the edges of the shiftable range (stage 1). Next, dimensions are calculated roughly in the shiftable range of segment boundary (stage 2). The number of calculation is estimated from the result of the stage 1. Then, dimensions are calculated in detail only in the portion where the change of dimensionality occurs (stage 3).

The problem here is how to determine the number of calculation in the stage 2. We can theoretically determine the optimal number of calculations. To do this, we set one assumption: the dimension monotonously increases as the segment length increases.

We again define some notations as in Table 2. Suppose that calculations are performed at the even intervals in the stage 2. Then, $f(x)$ is given as follows:

$$f(x) = 2\left\{(2x+3) + K\frac{\Delta}{x+1}\right\},$$

where the first term refers to the number of calculations in the stages 1 and 2, and the second term refers to that in the stage 3. K is given as follows:

$$K \begin{cases} = & C_{LR} - C_{LL} \quad \text{if } C_{LR} \le C_{RR}, C_{LL} < C_{RL} \\ \le & 2\min(C_{RC}, C_{LR}) - (C_{LL} + C_{RR}) \\ \ge & (C_{LC} - C_{LL}) + |C_{LC} - C_{RR}| \\ & \quad \text{if } C_{LR} > C_{RR}, C_{LL} < C_{RL}, C_{LC} \le C_{RC} \\ \le & 2\min(C_{LC}, C_{RL}) - (C_{LL} + C_{RR}) \\ \ge & (C_{RC} - C_{RR}) + |C_{RC} - C_{LL}| \\ & \quad \text{if } C_{LR} > C_{RR}, C_{LL} < C_{RL}, C_{LC} > C_{RC} \\ = & 0 \quad \text{Otherwise} \end{cases}$$

where

$$C_{LL} = c(t_{j-1}^*, t_j^0 - \Delta, \sigma), \quad C_{RL} = c(t_j^0 - \Delta, t_{j+1}^0, \sigma)$$
$$C_{LC} = c(t_{j-1}^*, t_j^0, \sigma), \quad C_{RC} = c(t_j^0, t_{j+1}^0, \sigma)$$
$$C_{LR} = c(t_{j-1}^*, t_j^0 + \Delta, \sigma), \quad C_{RR} = c(t_j^0 + \Delta, t_{j+1}^0, \sigma).$$

Therefore, the average number of the dimensionality changes, $\overline{K}$, is evaluated as

$$\overline{K} =$$

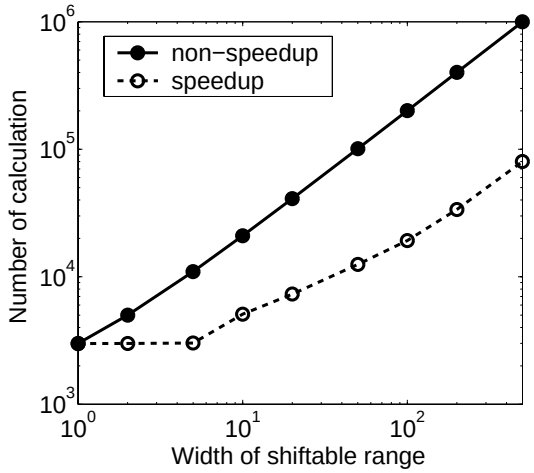

Fig. 4. Performance of the proposed method (1000 segments, contribution=0.75)

$$\begin{cases} C_{LR} - C_{LL} \quad \text{if } C_{LR} \le C_{RR}, C_{LL} < C_{RL} \\ (C_{LC} - C_{LL}) + \min(C_{RC}, C_{LR}) - \min(C_{LC}, C_{RR}) \\ \qquad \text{if } C_{LR} > C_{RR}, C_{LL} < C_{RL}, C_{LC} \le C_{RC} \\ (C_{RC} - C_{RR}) + \min(C_{LC}, C_{RL}) - \min(C_{RC}, C_{LL}) \\ \qquad \text{if } C_{LR} > C_{RR}, C_{LL} < C_{RL}, C_{LC} > C_{RC} \\ 0. \quad \text{Otherwise} \end{cases}$$

The $f(x)$ takes the minimum, $4\sqrt{2K\Delta}+1 = \mathcal{O}(\sqrt{K\Delta})$, when $x = \sqrt{\frac{1}{2}K\Delta} - 1$. Then, the total number of calculations is evaluated as

$$M\left(4\sqrt{2\overline{K}\Delta}+1\right) = \mathcal{O}\left(M\sqrt{\overline{K}\Delta}\right).$$

4. EXPERIMENTS

We carried out experiments to ascertain the performance of the speed-up method.

In the experiments, we used a video recording of a 24-hour TV broadcast as a stored signal. The frame rate was 29.97 Hz, and image size was 320×240 pixels. The tests were carried out on a PC. In the feature extraction, each frame was first divided into 6 (2×3) areas, and then the average of RGB-values was calculated for each area. Therefore, the number of dimensions of the original feature vector was 18. Those feature vectors were calculated on every frame, and then quantized. The codebook size for the feature vectors was 256. Then, histograms of the feature vectors were created. The histogram feature dimension was therefore 256 before compression. Those parameter values were empirically chosen.

Figure 4 shows the number of calculations in searching for optimal boundaries of the segments, where the horizontal axis is the width of the shiftable range and the vertical axis is the number of calculation. The proposed method is over 10 times faster than the one that does not employ speed-up technique when the number of segments is 1000, the contribution rate is 0.75, and the width of the shiftable range is 500.

5. CONCLUSIONS

We have proposed a quick and efficient dimension reduction method for multimedia search based on dynamic segmentation. A speed-up technique based on a coarse-to-fine approach achieves quick finding of suboptimal segmentation based on theoretical determination of the optimal number of calculation. In our experiment, the algorithm was over 10 times faster than a straightforward dynamic segmentation method in terms of the number of calculation when the number of segments is 1000, the contribution rate is 0.75, and the width of shiftable range is 500. Due to space limitations, we discussed a video example in the experiments section. The application to audio signal will be reported in a separate paper. Future work will include further investigation of the optimal decision of the segments.

Acknowledgment: The authors thank Dr. Ken-ichiro Ishii, Dr. Noboru Sugamura and Mr. Yoshinao Shiraki for their help and encouragement.

6. REFERENCES

[1] S. Berchtold et al.: "The X-tree : An Index Structure for High-Dimensional Data", *Proc. of VLDB96*, pp. 28-39, 1996.

[2] N. Katayama et al.: "The SR-tree : An Index Structure for High-Dimensional Nearest Neighbor Queries", *Proc. of ACM SIGMOD Conference 97*, pp. 369-380, 1997.

[3] A. Kimura et al.: "A Quick Search Method for Multimedia Signals Using Feature Compression Based on Piecewise Linear Maps", *Proc. of ICASSP2002*, Vol. 4, pp. 3656-3659, 2002.

[4] T. Pavlidis: "Waveform segmentation through functional approximation", *IEEE Trans. on Computers*, Vol. C-22, No. 7, pp. 689-697, 1973.

[5] E. Keogh et al.: "A Probabilistic Approach for Fast Pattern Matching in Time-series Databases", *Proc. of KDD97*, pp. 24-30, 1997.

[6] C. Wang et al.: "Supporting Content-based Searches on Time Series via Approximation", *Proc. of SSDBM2000*, pp. 69-81, 2000.

[7] E. Keogh et al.: "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases", *Proc. of ACM SIGMOD Conference 2001*, pp. 151-162, 2001.

[8] K. Kashino et al.: "Time-Series Active Search for Quick Retrieval of Audio and Video", *Proc. of ICASSP99*, Vol. 6, pp. 2993-2996, 1999.