

## Short Paper

---

# Dimensionality Reduction for Indexing Time Series Based on the Minimum Distance\*

SANGJUN LEE, DONGSEOP KWON AND SUKHO LEE

*School of Electrical Engineering and Computer Science*

*Seoul National University*

*Seoul 151-742, Korea*

We address the problem of efficient similarity search based on the minimum distance in large time series databases. To support minimum distance queries, most of previous work has to take the preprocessing step of vertical shifting. However, the vertical shifting has an additional overhead in building index. In this paper, we propose a novel dimensionality reduction technique for indexing time series based on the minimum distance. We call our approach the SSV-indexing (Segmented Sum of Variation Indexing). The proposed method can match time series of similar shape without vertical shifting and guarantees no false dismissals. Several experiments are performed on real data (stock price movement) to measure the performance of the SSV-indexing.

**Keywords:** database, similarity search, time series, minimum distance, autocorrelation

## 1. INTRODUCTION

A time series database is a collection of data that are generated in series as time goes on and constitutes a large portion of data stored in computers. Typical examples include stock price movements, exchange rates, weather data, biomedical measurements, etc. Similarity search in time series databases is essential because it helps in predicting and in hypothesis testing in data mining and knowledge discovery. Many techniques have been proposed to support the fast retrieval of similar time sequences based on the Euclidean distance [1, 2, 11]. However, the Euclidean distance as a similarity measure has the following problem: it is sensitive to the absolute offsets of the time sequences, so two time sequences that have similar shapes but with different vertical positions may be classified as dissimilar.

Consider a query time sequence  $Q$  (4 9 4 9 4) and two data time sequences  $A$  (7 5 6 7 6) and  $B$  (14 19 14 19 14) shown in Fig. 1. Note that shifting  $Q$  up by 10 units generates  $B$ . Using the similarity definition of the Euclidean distance,  $A$  is a more similar time sequence of  $Q$  than  $B$ . However,  $B$  is more similar to  $Q$  in shape. From this example, the Euclidean distance is not a good measurement of similarity when the shape is the major consideration.

---

Received October 22, 2001; revised April 30 & August 8, 2002; accepted October 1, 2002.

Communicated by Arbee L. P. Chen.

\* This work was supported in part by BK 21 project and ITRC program.

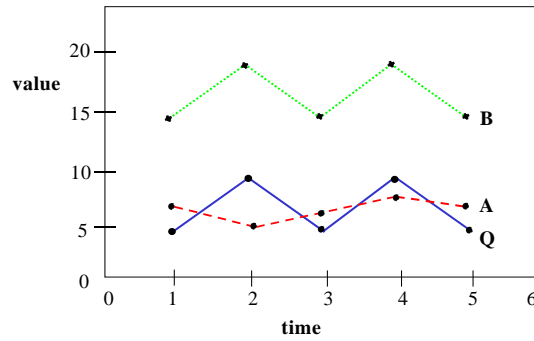


Fig. 1. Shortcoming of the Euclidean distance.

In order to overcome the shortcoming of the Euclidean distance, the minimum distance is often used for time sequence matching. The minimum distance can give a better estimation of similarity in shape between two time sequences irrespective of their vertical positions. The minimum distance is defined as follows.

**Definition 1 [Minimum Distance].** Given an error bound  $\varepsilon$ , two time sequences  $A(a_1, a_2, \dots, a_n)$  and  $B(b_1, b_2, \dots, b_n)$  of equal length  $n$  are said to be *similar in shape* if

$$D_{\text{minimum}}(A, B) = \left( \sum_{i=1}^n |a_i - b_i - m|^k \right)^{1/k} \leq \varepsilon \quad (1)$$

where  $m = \sum_{i=1}^n (a_i - b_i) / n$

The definition of distance depends on the selection of  $k$ . For  $k = 1$  and  $k = 2$ , they are known as the minimum Manhattan distance ( $L_1$  norm) and the minimum Euclidean distance ( $L_2$  norm), respectively. In this paper, the minimum Euclidean distance is used but our proposed method can handle the queries based on the minimum Manhattan distance.

To support minimum distance queries, most previous work has to take the pre-processing step of vertical shifting that normalizes each time sequence by its mean before indexing [3, 18, 22]. This has the effect of shifting the time sequence in the value-axis so that its mean is zero, removing its offset. However, the vertical shifting has the additional overhead of finding mean of the sequence and subtracting the mean from each element of the sequence.

The main issue of similarity search in time series databases is to improve the search performance. Since time sequences are usually long, the similarity calculation can be time consuming. As the size of the time series database increases, the sequential scanning scales poorly. In general, indexing is used to support fast retrieval and matching of similar sequences. Most approaches map sequences of length  $n$  into points in an  $n$ -dimensional space and a spatial access method such as R-tree can be used for fast retrieval of those points. However, a straightforward indexing of time sequences using the spatial access method suffers from performance deterioration due to the curse of dimensionality for the index structure. To solve this problem, several dimensionality reduction

methods have been proposed to map time sequences into a new feature space of a lower dimensionality. Typical dimensionality reduction methods include the Discrete Fourier Transform (DFT) and the Discrete Wavelet Transform (DWT), etc.

Another important issue of similarity search in time series databases is to choose the similarity measurement. Depending on the applications, several similarity measurements such as  $L_1$  norm and  $L_2$  norm are required for the same sequence database. To support this situation, several indexing methods must be implemented to support multi-modal queries in the same system, which is not easy or efficient. The dimensionality reduction method such as DFT or DWT is efficient only when the given similarity measurement is  $L_2$  norm. However, the indexing method using DFT or DWT cannot process minimum distance queries in  $L_1$  norm.

In this paper we propose a novel dimensionality reduction technique for indexing time series based on the minimum distance. The proposed method is motivated by autocorrelation of sequence. That is, the variation between two adjacent elements of a time sequence is invariant under vertical shifting. This autocorrelation motivated the dimensionality reduction technique introduced in this paper. The proposed method can match time series of similar shapes without vertical shifting and guarantees no false dismissals. In addition, the same index structure can be used to process the minimum distance queries in  $L_1$  norm and  $L_2$  norm.

The remainder of this paper is organized as follows. Section 2 provides a survey of related work our proposed approach is described in section 3. We will present the overall process of minimum distance queries in section 4. Section 5 presents the experimental results. Finally, several concluding remarks are given in section 6.

## 2. RELATED WORK

Various methods have been proposed for fast matching and retrieval of time series. The main goal is to speed up the search process. The most popular methods perform feature extraction as dimensionality reduction of time series data, and then use spatial access methods such as R-tree to index the time series data in the feature space.

An indexing scheme called the *F-index* [1] has been proposed to handle sequences of the same length. The idea is to use the Discrete Fourier Transform (DFT) to transform time sequence data from the time domain to the frequency domain, drop all but the first few frequencies, and then use the remaining ones to index the time sequence using spatial access methods. The results of [1] are extended in [2] and the *ST-index* is proposed for sequence matching for different lengths. The methods proposed in [1, 2] use the Euclidean distance as a similarity measure without considering any transformation. As shown in Fig. 1, it is better to consider the minimum distance as a measurement of similarity between two time sequences in some applications.

In [3], the authors show that the similarity retrieval is invariant to simple shifting and scaling if sequences are normalized before indexing. In [8, 9], the authors present an intuitive similarity measure for time series data. They argue that the similarity model with scaling and shifting is better than the Euclidean distance. However, they do not present any indexing method. In [4], the authors give a method to retrieve similar sequences in the presence of noise, scaling and transformation in time series databases. In [14], the

authors propose a definition of similarity based on scaling and shifting transformations. In [5], the authors present a hierarchical algorithm called the *HierarchyScan*. The idea of this method is to perform correlation between the stored sequences and the template in the transformed domain hierarchically.

In [12, 13], the authors focused on finding time series of similar shapes. They suggest the use of the slope of data sequences as the criterion for similarity based on the individual distance. Two sequences are said to be similar if the slope difference for each pair of corresponding segments is bounded by a predefined error bound.

In [6], the Singular Value Decomposition (SVD) is used for dimensionality reduction technique. The SVD is a global transformation which maps the entire dataset into a much smaller one. The SVD can be used to support ad hoc queries on large datasets of time sequences. The problem with using the SVD is the performance deterioration by update of index. SVD has to examine the entire dataset again to update index.

In [7], the authors proposed a set of linear transformations such as moving average, time warping, and reversing. These transformations can be used as the basis of similarity queries for time series data. The results of [7] are extended in [15] and the authors propose the method for processing queries that express similarity in terms of multiple transformations instead of a single one. In [10, 19], the authors use time warping as distance function and present algorithms for retrieving similar time sequences under this function. However, a time warping distance does not satisfy the triangular inequality and can cause false dismissals.

In [16], the authors first proposed using the Discrete Wavelet Transform (DWT) for dimensionality reduction and compared this method to the DFT. They showed that the Haar wavelet transform performs better than the DFT. However, the performance of the DFT could be improved using the symmetry property of Fourier Transform and the DWT has the limitation that it can only be defined for time sequences with a length of the power of two.

In [17], the *STB-indexing* method has been proposed. In this method, the time sequence is divided into non-overlapping segments of predefined window size. The segment is represented by bin value. If the segment is up-state, the bin value is set to 1 and if the segment is not up-state, the bin value is set to 0. After this transformation, the time sequence is stored in bins. Since this method stores the direction of segments, the meaning of similarity is different from the Euclidean distance. Thus, this method can cause false dismissals in the context of the Euclidean distance.

In [20], the authors proposed the *Landmark Model* for similarity-based pattern queries in time series databases. The *Landmark Model* integrates similarity measurement, data representation, and smoothing techniques in to a single framework. The model is based on the fact that people recognize patterns by identifying important points.

In [18, 22], the authors introduced a dimensionality reduction technique for a time sequence by using segmented mean features. In this method, the time sequence is divided into non-overlapping segments. The feature used to represent a segment is its mean value. Then, the time sequence is indexed in the feature space by spatial access methods. The same concept of independent research is proposed in [21]. However, the segmented mean features cannot support minimum distance queries, so vertical shifting is required in preprocessing raw data to process minimum distance queries.

### 3. PROPOSED APPROACH

The problem we focus on is the design of fast searching and retrieval of similar time sequences in databases based on the minimum distance. We will now introduce the segmented sum of variation indexing (SSV-indexing) and show that it guarantees no false dismissals.

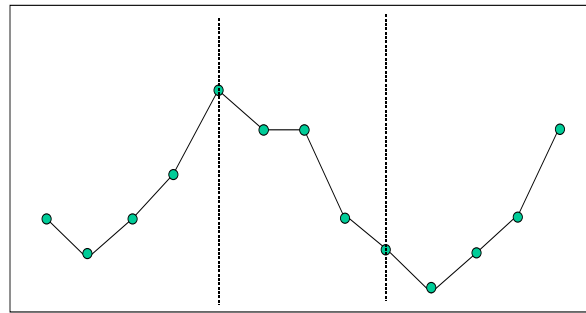
#### 3.1 Dimensionality Reduction

Our goal is to extract features that capture information about the shape of a time sequence, and that will lead to a feature distance definition satisfying the lower bound condition of the minimum distance. Suppose we have a set of time sequences of length  $n$ . The idea of our proposed feature extraction method consists of two steps. First, we divide each time sequence into  $s$  segments of equal length  $l$ . Note that the start point and end point of adjacent segments are the same because the variation will be missed if the sequence is divided disconnectedly. Next, we extract a simple feature from each segment. We propose using the sum of variation as the feature of a segment in a time sequence. Let  $FA_j$  denote the feature of the  $j$ -th segment of a time sequence  $A$ . We define a feature vector of a time sequence  $A$  as follows.

**Definition 2 [Segmented Sum of Variation Features].** Given a sequence  $A(a_1, a_2, \dots, a_n)$  and the number of segments  $s > 0$ , define the feature vector  $FA$  of  $A$  by

$$FA = \langle FA_1, FA_2, \dots, FA_s \rangle \\ = \langle \sum_{i=1}^{l-1} |a_{i+1} - a_i|, \sum_{i=l}^{2(l-1)} |a_{i+1} - a_i|, \dots, \sum_{i=(s-1)l+(2-s)}^{s(l-1)} |a_{i+1} - a_i| \rangle \quad (2)$$

Fig. 2 illustrates our dimensionality reduction technique. A time sequence of length 13 is projected into three dimensions. The time sequence is divided into three segments and the sum of variation of each segment is obtained. The algorithm to extract the feature vector is very simple. The sum of variation is invariant under vertical shifting, so the segmented sum of variation features can be indexed to support minimum distance queries. The proposed method is motivated by the following observation.



Time series A(5,4,5,6,8,7,7,5,4,3,4,5,7)

$$FA = (\text{sum of variation } (5,4,5,6,8), \text{ sum of variation } (8,7,7,5,4), \text{ sum of variation } (4,3,4,5,7)) \\ = (5,4,5)$$

Fig. 2. Dimensionality reduction technique used in this paper.

**Observation 1.** The segmented sum of variation is invariant under vertical shifting.

Verifying its correctness is obvious. The following equality shows its correctness.

$$FA_s = \sum_{i=(s-1)l+2-s}^{s(l-1)} |a_{i+1} - a_i| = \sum_{i=(s-1)l+2-s}^{s(l-1)} |(a_{i+1} - m_a) - (a_i - m_a)| \quad (3)$$

where  $m_a = \sum_{i=1}^n a_i / n$

### 3.2 Lower Bounding of the Minimum Distance<sup>1</sup>

In order to guarantee no false dismissals, we must construct a distance measurement  $D_{feature}(FA, FB)$  defined in the feature space, which has the following property.

$$D_{feature}(FA, FB) \leq D_{minimum}(A, B) \quad (4)$$

The distance between feature vectors  $D_{feature}(FA, FB)$  for the minimum Euclidean distance is:

$$D_{feature}(FA, FB) = \frac{(\sum_{i=1}^s |FA_i - FB_i|^2)^{1/2}}{2 \times \sqrt{l-1}} \quad (5)$$

We must show that the distance between feature vectors is the lower bound of the minimum distance between the original sequences. Before going into the main theorem, we will present some lemmas used in the main theorem.

**Lemma 1** Given two 2-point segments  $A(a_i, a_{i+1})$  and  $B(b_i, b_{i+1})$ , the variations of  $A$  and  $B$  are  $FA = |a_{i+1} - a_i|$  and  $FB = |b_{i+1} - b_i|$ . Then the following inequality holds:

$$\| |a_{i+1} - a_i| - |b_{i+1} - b_i| \|^2 \leq 2(|a_i - b_i - m|^2 + |a_{i+1} - b_{i+1} - m|^2) \quad (6)$$

where  $m = \sum_{i=1}^n (a_i - b_i) / n$

**Proof:** By Observation 1, we know that

$$\| |a_{i+1} - a_i| - |b_{i+1} - b_i| \|^2 = \| (a_{i+1} - m_a) - (a_i - m_a) - (b_{i+1} - m_b) + (b_i - m_b) \|^2 \quad (7)$$

where  $m_a = \frac{1}{n} \sum_{i=1}^n a_i$  and  $m_b = \frac{1}{n} \sum_{i=1}^n b_i$

<sup>1</sup> We will only show the lower bound of the minimum Euclidean distance ( $L_2$  norm) in this paper. The distance feature  $D_{feature}(FA, FB)$  for the minimum Manhattan distance ( $L_1$  norm) is given by

$$D_{feature}(FA, FB) = \frac{\sum_{i=1}^s |FA_i - FB_i|}{2}$$

It can be easily proved that Eq. (4) is satisfied in the case of the minimum Manhattan distance in the same way, so we omit it.

Then, we have

$$\begin{aligned}
 & \left\| (a_{i+1} - m_a) - (a_i - m_a) - (b_{i+1} - m_b) - (b_i - m_b) \right\|^2 \\
 & \leq \left| ((a_{i+1} - m_a) - (a_i - m_a)) - ((b_{i+1} - m_b) - (b_i - m_b)) \right|^2 \\
 & = \left| ((a_{i+1} - m_a) - (b_{i+1} - m_b)) - ((a_i - m_a) - (b_i - m_b)) \right|^2 \\
 & \leq \left( |(a_{i+1} - m_a) - (b_{i+1} - m_b)| + |(a_i - m_a) - (b_i - m_b)| \right)^2 \\
 & \leq 2(|a_i - b_i - (m_a - m_b)|^2 + |a_{i+1} - b_{i+1} - (m_a - m_b)|^2)
 \end{aligned} \tag{8}$$

thus  $\|a_{i+1} - a_i - |b_{i+1} - b_i|\|^2 \leq 2(|a_i - b_i - m|^2 + |a_{i+1} - b_{i+1} - m|^2)$

where  $m = \sum_{i=1}^n (a_i - b_i) / n$  □

**Lemma 2** Given two  $l$ -point segments  $A(a_1, a_2, \dots, a_l)$  and  $B(b_1, b_2, \dots, b_l)$ , the sum of variation of  $A$  is  $FA_1 = \sum_{i=1}^{l-1} |a_{i+1} - a_i|$  and that of  $B$  is  $FB_1 = \sum_{i=1}^{l-1} |b_{i+1} - b_i|$ . Then, the following inequality holds:

$$\begin{aligned}
 |FA_1 - FB_1|^2 & \leq 2(l-1) \left( \sum_{i=1}^{l-1} |a_i - b_i - m|^2 + \sum_{i=2}^l |a_i - b_i - m|^2 \right) \\
 \text{where } m & = \sum_{i=1}^n (a_i - b_i) / n
 \end{aligned} \tag{10}$$

**Proof:** we know that

$$\begin{aligned}
 & |FA_1 - FB_1|^2 \\
 & = \left( |a_2 - a_1| + \dots + |a_l - a_{l-1}| - (|b_2 - b_1| + \dots + |b_l - b_{l-1}|) \right)^2 \\
 & = \left( |a_2 - a_1| - |b_2 - b_1| + \dots + (|a_l - a_{l-1}| - |b_l - b_{l-1}|) \right)^2
 \end{aligned} \tag{11}$$

and the following inequality holds by Cauchy-Schwartz's inequality.

$$\begin{aligned}
 & \left( |a_2 - a_1| - |b_2 - b_1| + \dots + (|a_l - a_{l-1}| - |b_l - b_{l-1}|) \right)^2 \\
 & \leq (l-1) \left( \|a_2 - a_1 - |b_2 - b_1|\|^2 + \dots + \|a_l - a_{l-1} - |b_l - b_{l-1}|\|^2 \right)
 \end{aligned} \tag{12}$$

According to Lemma 1, we can get the following inequality.

$$\begin{aligned}
 & (l-1) \left( \|a_2 - a_1 - |b_2 - b_1|\|^2 + \dots + \|a_l - a_{l-1} - |b_l - b_{l-1}|\|^2 \right) \\
 & \leq (l-1) \cdot 2 \cdot \left\{ \begin{aligned} & (|a_2 - b_2 - m|^2 + |a_1 - b_1 - m|^2) + \\ & (|a_3 - b_3 - m|^2 + |a_2 - b_2 - m|^2) + \dots + \\ & (|a_{l-1} - b_{l-1} - m|^2 + |a_{l-2} - b_{l-2} - m|^2) + \\ & (|a_l - b_l - m|^2 + |a_{l-1} - b_{l-1} - m|^2) \end{aligned} \right\} \\
 & = 2(l-1) \left( \sum_{i=1}^{l-1} |a_i - b_i - m|^2 + \sum_{i=2}^l |a_i - b_i - m|^2 \right)
 \end{aligned} \tag{13}$$

$$\begin{aligned} \text{thus } |FA_1 - FB_1|^2 &\leq 2(l-1) \left( \sum_{i=1}^{l-1} |a_i - b_i - m|^2 + \sum_{i=2}^l |a_i - b_i - m|^2 \right) \\ \text{where } m &= \sum_{i=1}^n (a_i - b_i) / n \quad \square \end{aligned} \quad (14)$$

**Theorem 1**  $D_{feature}(FA, FB)$  is the lower bound of  $D_{minimum}(A, B)$  and it guarantees no false dismissals.

$$D_{feature}(FA, FB) \leq D_{minimum}(A, B) \quad (15)$$

**Proof:** Based on Lemma 2, we can get the following inequality.

$$\begin{aligned} \sum_{i=1}^s |FA_i - FB_i|^2 &= |FA_1 - FB_1|^2 + |FA_2 - FB_2|^2 + \dots + |FA_s - FB_s|^2 \\ &\leq 2(l-1) \left\{ \begin{array}{l} \sum_{i=1}^{l-1} |a_i - b_i - m|^2 + \sum_{i=2}^l |a_i - b_i - m|^2 + \\ \sum_{i=1}^{2(l-1)} |a_i - b_i - m|^2 + \sum_{i=l+1}^{2(l-1)+1} |a_i - b_i - m|^2 + \\ \dots \\ \sum_{i=(s-1)l+(2-s)}^{s(l-1)} |a_i - b_i - m|^2 + \sum_{i=(s-1)l+(3-s)}^{s(l-1)+1} |a_i - b_i - m|^2 \end{array} \right\} \\ &= 2(l-1) \left( \sum_{i=1}^{n-1} |a_i - b_i - m|^2 + \sum_{i=2}^n |a_i - b_i - m|^2 \right) \\ &\leq 2^2(l-1) \sum_{i=1}^n |a_i - b_i - m|^2 \end{aligned} \quad (16)$$

From the above result, we can prove the Theorem 1.

$$\begin{aligned} D_{feature}(FA, FB) &= \frac{(\sum_{i=1}^s |FA_i - FB_i|^2)^{1/2}}{2 \times \sqrt{l-1}} \\ &\leq (\sum_{i=1}^n |a_i - b_i - m|^2)^{1/2} = D_{minimum}(A, B) \\ \text{where } m &= \sum_{i=1}^n (a_i - b_i) / n \quad \square \end{aligned} \quad (17)$$

Lower bound on the minimum distance with the feature distance is a condition that guarantees no false dismissals for similarity search in time series databases.

#### 4. QUERY PROCESSING

In this section we first present a sequential scanning which is the simplest search algorithm for sequence matching based on the minimum distance. Then, we present the overall process of the segmented sum of variation indexing scheme.



## 4.1 Sequential Scanning

Sequential scanning searches an entire database. It computes the minimum distance between query sequence and all data sequences in a database. If the minimum distance is within a given error bound, the data sequence is classified as similar to the query sequence. Otherwise, it is classified as dissimilar and rejected. Sequential scanning guarantees that no qualified data sequence is wrongly rejected. However, sequential scanning is slow because it has to access all data sequences in the database. As the size of a time series database increases, sequential scanning scales poorly. The implementation of the sequential scanning is described in Algorithm 1.

### Algorithm 1 Sequential Scanning

**Input:** query time sequence  $Q$ , error bound  $\epsilon$

**Output:** matching data time sequences within error bound  $\epsilon$

**Begin**

    result  $\leftarrow$  NULL

//  $S_i \in$  time series database

**for** ( $i = 1; i < \text{size of database}; i++$ )

**if** ( $\text{ComputeMinimumDistance}(S_i, Q) \leq \epsilon$ )

            result  $\leftarrow S_i \cup \text{result}$

**else** reject  $S_i$

**return** result

**End**

## 4.2 Segmented Sum of Variation Indexing (SSV-indexing)

We present the overall process of the SSV-indexing. Before a query is evaluated, some preprocessing is needed to extract feature vectors from time sequences, and to build an index. After the index is built, the similarity search can be performed to select candidate sequences from the database. The transformation for a single sequence of length  $n$  takes  $O(n)$  time, and thus the entire index for  $K$  time sequences can be built in  $O(Kn)$ .

### 4.2.1 Preprocessing

#### Step 1. Feature Extraction as Dimensionality Reduction

Each sequence is divided into  $s$  segments. Then, the feature vector is extracted from the sequence using the method described in section 3.1 for all time sequences in the database.

#### Step 2. Index Construction

We build a multidimensional index structure such as an R-tree using the feature vectors extracted from time sequences. Note that the optimal dimensionality of feature vectors can be obtained by experiments. The increase in dimensionality results in better index selectivity, which gives fewer false alarms. This reduction in false alarms is reflected in the postprocessing time. However, the time to search the index structure increases as the dimensionality increases because the fanout gets smaller and the tree gets taller. There is a trade-off between the tree search time and the postprocessing time.

### 4.2.2 Index searching

After an index structure has been built, we can perform the similarity search against a given query sequence. The search algorithm consists of two main parts. The first is for candidate selection and the other is postprocessing to remove false alarms. Some non-qualifying time sequences may be included in the result of candidate selection because the  $D_{feature}(FA, FB)$  is the lower bound of  $D_{minimum}(A, B)$ . The actual minimum distance between the query sequence and candidate sequences is computed and only those within the error bound are selected as query results. The implementation of the segmented sum of variation indexing is described in Algorithm 2.

#### Algorithm 2 Segmented Sum of Variation Indexing

**Input:** query time sequence  $Q$ , error bound  $\epsilon$

**Output:** data time sequences within error bound  $\epsilon$

**Begin**

    result  $\leftarrow$  NULL

    candidate  $\leftarrow$  NULL

// project the query time sequence  $Q$  into the index space

    FQ  $\leftarrow$  FeatureExtraction( $Q$ )

// Candidate selection using a Spatial Access Method

    candidate  $\leftarrow$  candidate  $\cup$  FeatureSpaceSearching(FQ, SAM,  $\epsilon$ )

// Postprocessing to remove false alarms

//  $C_i \in$  candidate

**for** ( $i = 1$ ;  $i <$  size of candidate;  $i++$ )

**if**(ComputeMinimumDistance( $C_i$ ,  $Q$ )  $\leq \epsilon$ )

            result  $\leftarrow$   $C_i \cup$  result

**else** reject  $C_i$

**return** result

**End**

## 5. PERFORMANCE EVALUATION

In this section, we will present the results of some experiments to analyze the performance of the SSV-indexing. To verify the effectiveness of the proposed SSV-indexing, we compared the SSV-indexing with sequential scanning with respect to the search space ratio and the elapsed time to process minimum distance queries. The experimental settings are described first and the results of experiments are given next.

### 5.1 Experimental Settings

We implemented both the SSV-indexing and the sequential scanning in C++ on a Linux machine (Redhat 7.1, kernel version 2.4.2) with dual Pentium III 500MHz CPUs, 512MB of memory and 40GB HDD. The size of the disk page is set to 4KB. For a spatial access method, we used the Katayama's R\*-tree source codes<sup>2</sup>. The real sequence data

<sup>2</sup> The Katayama's R\*-tree source codes can be retrieved from <http://research.nii.ac.jp/~katayama/homepage/research/srtree/English.html>.

were obtained from Seoul Stock Market, Korea from January 1998 to March 2000<sup>3</sup>. The stock data were based on their daily closing prices. The stock price database consists of 2000 stocks of length 128. We ran 25 random queries over real dataset to find similar time sequences based on the minimum distance. The query sequences were randomly selected from the database.

## 5.2 Experimental Results

First, we compared the SSV-indexing with the sequential scanning in terms of the actual number of computations required for the  $L_1$  norm and  $L_2$  norm. We evaluated the search space ratio to test the filtering power of removing irrelevant time sequences in the process of index searching. The search space ratio is defined as

$$\text{search space ratio} = \frac{\text{number of candidate sequences}}{\text{number of sequences in a database}}$$

Note that the value of the search space ratio depends on the feature vectors and is independent of implementation factors such as spatial access method, software and hardware platforms. Fig. 3. shows the search space ratios as a function of error bound using the  $L_1$  norm and  $L_2$  norm. The values of error bounds are related to the selectivity of result. The values were chosen such that the average selectivity of results be up to about 1% for the  $L_1$  norm and up to about 2% for the  $L_2$  norm. The dimensionality of feature vectors for the SSV-indexing is set to 8 in this experiment. The experimental result shows that there is a significant gain in reducing the search space by the SSV-indexing.

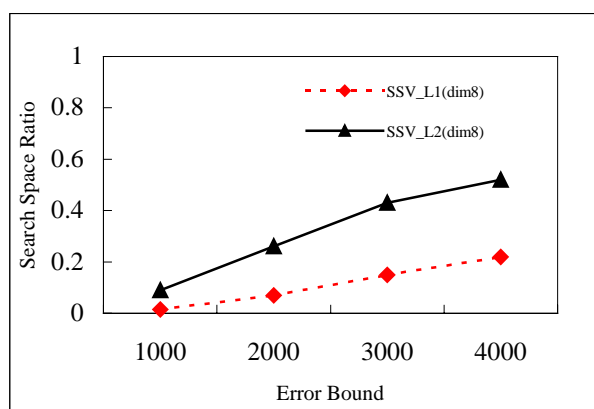


Fig. 3. Search Space Ratio as a function of error bound using  $L_1$  norm and  $L_2$  norm.

Good filtering does not always result in good performance since the spatial access method can be affected by the dimensionality of feature (i.e., dimensionality curse). Although the pruning power improves as the dimensionality of feature vector increases,

<sup>3</sup> This data can be retrieved from [http://www.kse.or.kr/kor/stat/stat\\_data.htm](http://www.kse.or.kr/kor/stat/stat_data.htm).

most indexing structures show a rapid degradation with increasing dimensionality which leads to accessing the entire database for each query. To verify the dimensionality curse, we compared SSV-indexing of 8 dimensions and 16 dimensions with sequential scanning with respect to the search space ratio and the average execution time by varying the error bound in the  $L_2$  norm. Figs. 4 and 5 show that SSV-indexing of 16 dimensions is more efficient than SSV-indexing of 8 dimensions in reducing the search space. However, the performance of SSV-indexing of 16 dimensions is not better than the SSV-indexing of 8 dimensions. The average response time increases as the error bound increases. This is because the number of retrieved time sequences for a larger error bound increases more rapidly than the number of relevant time sequences. Consequently, the postprocessing time to remove false alarms increases with a larger error bound.

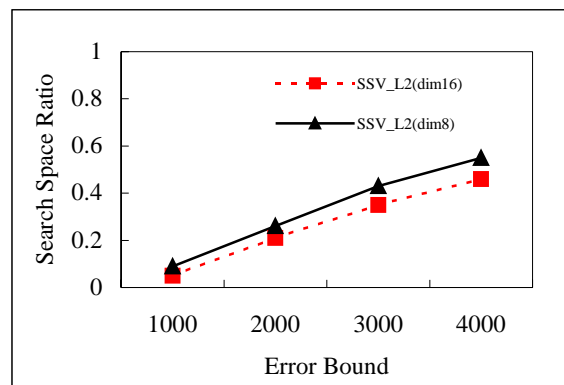


Fig. 4. Search Space Ratio: SSV-indexing vs. sequential scanning in  $L_2$  norm.

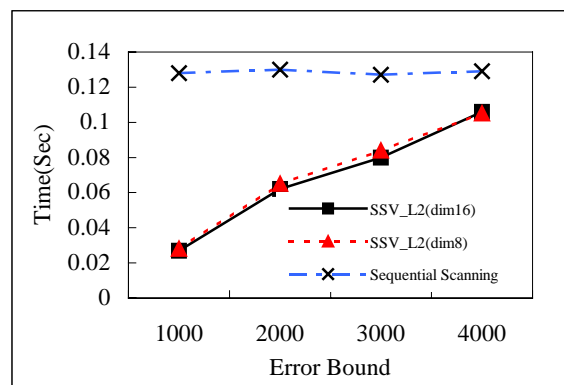


Fig. 5. Average Response Time: SSV-indexing vs. sequential scanning in  $L_2$  norm.

## 6. CONCLUSIONS

This paper focused on fast similarity search in time series databases, when the similarity measurement is the minimum distance. To support minimum distance queries, most previous work has to take the preprocessing step of vertical shifting that normalizes each time sequence by its mean before indexing. In this paper, we have proposed a novel dimensionality reduction technique for time series indexing that supports minimum distance queries without vertical shifting. Our dimensionality reduction technique is motivated by autocorrelation of a time sequence, that is, the sum of variation in a time sequence is invariant under vertical shifting. Using the autocorrelation of a time sequence, we can handle minimum distance queries without vertical shifting.

The major contributions of this work are: 1) introducing an efficient dimensionality reduction technique for time series indexing that allows fast matching of time series based on the minimum distance without vertical shifting; 2) presenting the lower bound of the minimum distance to filter out dissimilar time sequences without false dismissals; 3) showing that the same index structure can be used to process the minimum distance queries in the  $L_1$  norm and  $L_2$  norm.

We have performed the experiments on real stock data, and examined the pruning power and performance of our proposed method compared with sequential scanning. The experiments show that the SSV-indexing is more efficient than sequential scanning.

## REFERENCES

1. R. Agrawal, C. Faloutsos, and A. N. Swami, "Efficient similarity search in sequence databases," in *Proceedings of International Conference of Foundations of Data Organization (FODO)*, 1993, pp. 69-84.
2. C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," in *Proceedings of ACM SIGMOD Conference*, 1994, pp. 419-429.
3. D. Q. Goldin and P. C. Kanellakis, "On similarity queries for time-series data: constraint specification and implementation," in *Proceedings of International Conference on Principles and Practice of Constraint*, 1995, pp. 137-153.
4. R. Agrawal, K.-I. Lin, H. S. Sawhney, and K. Shim, "Fast similarity search in the presence of noise, scaling, and translation in time-series databases," in *Proceedings of International Conference on Very Large Data Bases (VLDB)*, 1995, pp. 490-501.
5. C.-S. Li, P. S. Yu, and V. Castelli, "HierarchyScan: A hierarchical similarity search algorithm for databases of long sequences," in *Proceedings of International Conference on Data Engineering (ICDE)*, 1996, pp. 546-553.
6. F. Korn, H. V. Jagadish, and C. Faloutsos, "Efficiently supporting Ad hoc queries in large datasets of time sequences," in *Proceedings of ACM SIGMOD Conference*, 1997, pp. 289-300.
7. D. Rafiei and A. O. Mendelzon, "Similarity-based queries for time series data," in *Proceedings of ACM SIGMOD Conference*, 1997, pp. 13-25.
8. G. Das, D. Gunopulos, and H. Mannila, "Finding similar time series," in *Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, 1997, pp. 88-100.

9. B. Bollobás, G. Das, D. Gunopulos, and H. Mannila, "Time-series similarity problems and well-separated geometric sets," in *Proceedings of Symposium on Computational Geometry*, 1997, pp. 454-456.
10. B.-K. Yi, H. V. Jagadish, and C. Faloutsos, "Efficient retrieval of similar time sequences under time warping," in *Proceedings of International Conference on Data Engineering (ICDE)*, 1998, pp. 201-208.
11. D. Rafiei and A. O. Mendelzon, "Efficient retrieval of similar time sequences using DFT," in *Proceedings of International Conference of Foundations of Data Organization (FODO)*, 1998, pp. 249-257.
12. S. K. Lam and M. H. Wong, "A fast projection algorithm for sequence data searching," *Data & Knowledge Engineering (DKE)*, Vol. 28, 1998, pp. 321-339.
13. K. K. W. Chu, S. K. Lam, and M. H. Wong, "An efficient hash-based algorithm for sequence data searching," *The Computer Journal*, Vol. 41, 1998, pp. 402-415.
14. K. K. W. Chu and M. H. Wong, "Fast time-series searching with scaling and shifting," in *Proceedings of Symposium on Principles of Database Systems (PODS)*, 1999, pp. 237-248.
15. D. Rafiei, "On similarity-based queries for time series data," in *Proceedings of International Conference on Data Engineering (ICDE)*, 1999, pp. 410-417.
16. K.-P. Chan and A. W.-C. Fu, "Efficient time series matching by wavelets," in *Proceedings of International Conference on Data Engineering (ICDE)*, 1999, pp. 126-133.
17. E. J. Keogh and M. J. Pazzani, "An indexing scheme for fast similarity search in large time series databases," in *Proceedings of International Conference on Scientific and Statistical Database Management (SSDBM)*, 1999, pp. 56-67.
18. E. J. Keogh and M. J. Pazzani, "A simple dimensionality reduction technique for fast similarity search in large time series databases," in *Proceedings Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2000, pp. 122-133.
19. S. Park, W. W. Chu, J. Yoon, and C. Hsu, "Efficient searches for similar subsequences of different lengths in sequence databases," in *Proceedings of International Conference on Data Engineering (ICDE)*, 2000, pp. 23-32.
20. C.-S. Perng, H. Wang, S. R. Zhang, and D. S. Parker, "Landmarks: a new model for similarity-based pattern querying in time series databases," in *Proceedings of International Conference on Data Engineering (ICDE)*, 2000, pp. 33-42.
21. B.-K. Yi and C. Faloutsos, "Fast time sequence indexing for arbitrary  $L_p$  norms," in *Proceedings of International Conference on Very Large Data Bases (VLDB)*, 2000, pp. 385-394.
22. E. J. Keogh, K. Chakrabarti, S. Mehrotra, and M. J. Pazzani, "Locally adaptive dimensionality reduction for indexing large time series databases," in *Proceedings of ACM SIGMOD Conference*, 2001, pp. 151-162.

**Sangjun Lee** is a Ph.D. candidate in the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea. He received his M.S. and B.S. degrees in the Department of Computer Engineering from Seoul National University, Seoul, Korea, in 1996 and 1998, respectively. His current research interests include high dimensional index structures, mobile data managements, and time series databases.

**Dongseop Kwon** is a Ph.D. candidate in the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea. He received his M.S. and B.S. degrees in the Department of Computer Engineering from Seoul National University, Seoul, Korea, in 1998 and 2000, respectively. His current research interests include XML, high dimensional index structures, mobile data managements, and time series databases.

**Sukho Lee** received his B.A. degree in Political Science and Diplomacy from Yonsei University, Seoul, Korea, in 1964 and his M.S. and Ph.D. in Computer Sciences from the University of Texas at Austin in 1975 and 1979, respectively. He is currently a professor of the School of Computer Science and Engineering, Seoul National University, Seoul, Korea, where he has been leading the Database Research Laboratory. He has served as the president of Korea Information Science Society. His current research interests include database management systems, spatial database systems, and multimedia database systems.