# Efficient Similarity Search for Time Series Data 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
{freude,subby}@db.snu.ac.kr
shlee@cse.snu.ac.kr

**Abstract.** We address the problem of efficient similarity search based on the minimum distance in large time series databases. Most of previous work is focused on similarity matching and retrieval of time series based on the Euclidean distance. However, as we demonstrate in this paper, the Euclidean distance has limitations as a similarity measurement. It is sensitive to the absolute offsets of time sequences, so two time sequences that have similar shapes but with different vertical positions may be classified as dissimilar. The minimum distance is a more suitable similarity measurement than the Euclidean distance in many applications, where the shape of time series is a major consideration. To support minimum distance queries, most of previous work has the preprocessing step of vertical shifting that normalizes each time sequence by its mean before indexing. In this paper, we propose a novel and fast indexing scheme, called the segmented mean variation indexing(SMV-indexing). Our indexing scheme can match time series of similar shapes without vertical shifting and guarantees no false dismissals. Several experiments are performed on real data(stock price movement) to measure the performance of our indexing scheme. Experiments show that the SMV-indexing is more efficient than the sequential scanning in performance.

## 1 Introduction

Time sequences are of growing importance in many database applications, such as data mining and data warehousing[1,2]. A time sequence is a sequence of real numbers and each number represents a value at a time point. Typical examples include stock price movement, exchange rate, weather data, biomedical measurement, etc. Similarity search in time series databases is essential, because it helps predicting, hypothesis testing in data mining and knowledge discovery[1,2]. Many techniques have been proposed to support the fast retrieval of similar time sequences based on the Euclidean distance[5,6,17]. However, the Euclidean distance as a similarity measurement has the following problem: it is sensitive to the absolute offsets of time sequences, so two time sequences that have similar shapes but with different vertical positions may be classified as dissimilar.

* This work was supported by the Brain Korea 21 Project in 2001

A. Banks Pidduck et al. (Eds.): CAISE 2002, LNCS 2348, pp. 377–391, 2002.

Consider a query time sequence $Q = (4, 9, 4, 9, 4)$ and two data time sequences $A = (7, 5, 6, 7, 6)$ and $B = (14, 19, 14, 19, 14)$ in Figure 1. Note that shifting $Q$ upward for 10 units generates $B$. Using the similarity definition of the Euclidean distance, $A$ is a more similar to $Q$ 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 we would like to retrieve the time sequences of similar shapes irrespective of their vertical positions in time series databases.

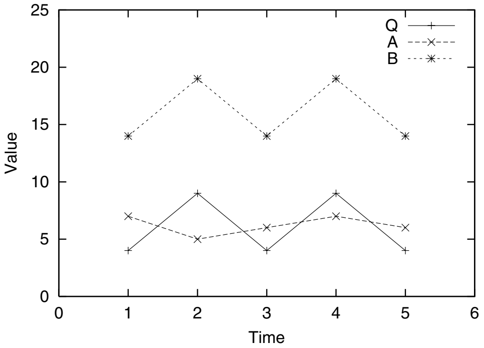


**Fig. 1.** Shortcoming of the Euclidean distance

In order to overcome the shortcoming of the Euclidean distance above mentioned, the minimum distance can be used for time sequence matching[18,19,22]. The minimum distance is the Euclidean distance between two time sequences, neglecting their offsets, that is, the minimum Euclidean distance over various shift factors. This can give a better estimation of similarity in shape between two time sequences irrespective of their vertical positions. To support minimum distance queries, most of previous work has to take the preprocessing step of vertical shifting which normalizes each time sequence by its mean before indexing[7,23,27]. This has the effect of shifting the time sequence in the value-axis such that its mean is zero, removing its offset. However, the vertical shifting has the additional overhead to get the mean of a time sequence and to subtract the mean from each element of the time sequence.

Time sequences are usually long, so the similarity calculation can be time consuming. As the size of time series databases increases, the sequential scanning scales poorly. To reduce the number of similarity calculations, an efficient indexing scheme is required. Indexing, however, on such multidimensional data doesn't seem to be easy because of dimensionality curse in making an index structure using a spatial access method such as R-tree[3] or R*-tree[4].

In this paper, we propose a novel and fast indexing scheme for time series, called the segmented mean variation indexing(SMV-indexing). This method is motivated by autocorrelation of time sequence, that is, the variation between two adjacent elements of time sequence is invariant under vertical shifting. This autocorrelation motivated the dimensionality reduction technique introduced in this paper. Our indexing scheme can match time series of similar shapes without vertical shifting and guarantees no false dismissals.

The remainder of this paper is organized as follows. Section 2 provides a survey of related work. We will give similarity measurements used in sequence matching in Section 3, and our proposed approach is described in Section 4. We will present the overall process of minimum distance queries in Section 5. Section 6 presents the experimental results. Finally, several concluding remarks are given in Section 7.

## 2 Related Work

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

An indexing scheme called *F-index*[5] is proposed to handle sequences of the same length. The idea is to use Discrete Fourier Transform(DFT) to transform time sequence data from time domain to frequency domain, drop all but the first few frequencies, and then use the remaining ones to index the time sequence using a spatial access method such as R-tree. The results of [5] are extended in [6] and the *ST-index* is proposed for sequence matching of the different length. The methods proposed in [5,6] use the Euclidean distance as a similarity measurement without considering any transformation. As shown in Figure 1, it is better to consider the minimum distance as a similarity measurement in some applications.

In [7], the authors show that the similarity retrieval will be invariant to simple shifting and scaling if sequences are normalized before indexing. In [14,15], authors present an intuitive similarity measurement 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 [8], authors give a method to retrieve similar sequences in the presence of noise, scaling and transformation in time series databases. In [20], the authors propose a definition of similarity based on scaling and shifting transformations. In [9] authors present a hierarchical algorithm called *HierarchyScan*. The idea of this method is to perform correlation between the stored sequences and the template in the transformed domain hierarchically. In [18,19], authors propose the definition of similarity between sequences based on the slope of segments.

In [12], authors use vantage point trees for indexing and matching sequences of different lengths based on a modified version of the edit distance, considering two sequences similar if a majority of elements match. This technique is extended to matching of multidimensional data sequences such as sequences of images[13].

In [10], Singular Value Decomposition(SVD) is used for dimensionality reduction technique. SVD is a global transformation which maps the entire dataset into a much smaller one. SVD can be used to support ad hoc queries on large dataset of time sequences. The problem of SVD is the performance deterioration by update of an index structure. SVD has to examine the entire dataset again to update an index structure.

In [11], authors propose 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 [11] are extended in [21] and authors propose the method for processing queries that express similarity in terms of multiple transformations instead of a single one. In [16,24], 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 triangular inequality and can cause false dismissals.

In [22], authors propose to use Discrete Wavelet Transform(DWT) for dimensionality reduction and compare this method to DFT. They argue that Haar wavelet transform performs better than DFT. However, the performance of DFT can be improved using the symmetry of Fourier Transforms and DWT has the limitation that it can only be defined for time sequences with a length of the power of two.

In [25], authors propose the *Landmark Model* for similarity-based pattern queries in time series databases. The *Landmark Model* integrates similarity measurement, data representation, and smoothing techniques in a single framework. The model is based on the fact that people recognize patterns by identifying important points.

In [23,27] the authors introduce new dimensionality reduction technique of time sequence by using segmented mean features. In this method, the time sequence is divided into non-overlapping segments. The feature of segment is represented by segment's mean value. Then, the time sequence is indexed in the feature space by a spatial access method. The same concept of independent research is proposed in [26]. They show that the segmented mean features can be used in arbitrary $L_p$ norm. However, the segmented mean features cannot support minimum distance queries, vertical shifting is required in preprocessing raw data to support minimum distance queries.

## 3 Similarity Measurements for Sequence Matching

In this section, we will give two similarity measurements used in sequence matching. The first definition is based on the Euclidean distance between two time sequences.

**Definition 1 (Euclidean Distance).** *Given a threshold $\epsilon$, two time sequences $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$ of equal length $n$ are said to be similar if*

$$L_2^{Euclidean}(A, B) = (\sum_{i=1}^{n} |a_i - b_i|^2)^{1/2} \leq \epsilon$$

A shortcoming of Definition 1 is demonstrated in Figure 1. Two time sequences $B$ and $Q$ are the same in shape, because B can be obtained by shifting $Q$ upward 10 units. However, they can be classified as dissimilar by Definition 1. From this example, the Euclidean distance is not a suitable measurement of similarity when the shape is a major consideration.

**Definition 2 (Minimum Distance).** *Given a threshold $\epsilon$, two time sequences $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$ of equal length $n$ are said to be similar in shape if*

$$L_2^{minimum}(A,B) = (\sum_{i=1}^{n} |a_i - b_i - m|^2)^{1/2} \leq \epsilon$$

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

From Definition 2, two time sequences are said to be similar in shape if the minimum distance is less or equal to a threshold $\epsilon$. This definition is a more flexible similarity measurement when we would like to retrieve time sequences of similar shapes from databases irrespective of their vertical positions.

## 4 Proposed Approach

The problem we focus on is the design of fast retrieval of similar time sequences in databases based on the minimum distance. To the best of our knowledge, this is the first work that examines indexing methods for minimum distance queries without vertical shifting for time series of an arbitrary length. We will now introduce the SMV-indexing and show that it guarantees no false dismissals.

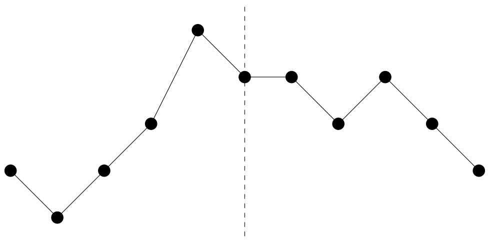
### 4.1 Dimensionality Reduction

Our goal is to extract features that capture the information on the shape of a time sequence, and that will lead to the feature distance definition satisfying the lower bound condition of the minimum distance. Suppose that 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 $m$ 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 time sequence is divided disconnectedly. In case the length of time series is not $m(l-1)+1$, then we add the final element's values at the end of sequence. This has no effect on query results. Next, we extract a simple feature from each segment. We propose to use the mean variation as the feature of each segment. $FA_j$ denotes the feature of the $j-th$ segment of the time sequence $A$. We define the feature vector of the time sequence $A$ as follows.

**Definition 3 (Segmented Mean Variation Feature).** *Given a sequence $A = \{a_1, a_2, \ldots, a_n\}$ and the number of segments $m > 0$, define the feature vector $\overrightarrow{FA}$ of the time sequence $A$ by*

$$\overrightarrow{FA} = (FA_1, FA_2, \ldots, FA_m)$$
$$= \frac{1}{l-1} \cdot (\sum_{i=1}^{l-1} |a_{i+1} - a_i|, \sum_{i=l}^{2(l-1)} |a_{i+1} - a_i|, \ldots, \sum_{i=(m-1)l+(2-m)}^{m(l-1)} |a_{i+1} - a_i|)$$

Figure 2 illustrates the dimensionality reduction technique used in this paper. A time sequence of length 11 is projected into two dimensions. The time sequence is divided into two segments and the mean variation of each segment is obtained.



$$\overrightarrow{FA} = (MeanVariation(5,4,5,6,8,7), MeanVariation(7,7,6,7,6,5))$$
$$= (1.2, 0.8)$$

**Fig. 2.** Dimensionality Reduction Technique used in this paper

The algorithm to extract the feature vector is obviously simple. The mean variation of a segment is invariant under vertical shifting, so the segmented mean variation features can be indexed to support minimum distance queries. The proposed matching scheme is motivated by the following observation.

**Observation 1 (Autocorrelation of Time Sequence)** *The segmented mean variation is invariant under vertical shifting.*

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

$$
\begin{aligned}
MeanVariation(A) &= \frac{1}{(n-1)} \sum_{i=1}^{n-1} |a_{i+1} - a_i| \\
&= \frac{1}{(n-1)} \sum_{i=1}^{n-1} |(a_{i+1} - m_a) - (a_i - m_a)|
\end{aligned}
$$

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

### 4.2 Lower Bounding of the Minimum Distance

In order to guarantee no false dismissals, we must show that the distance between feature vectors is the lower bound of the minimum distance between original sequences. Lower bounding the minimum distance with the feature distance is a condition that guarantees no false dismissals for similarity search in time series databases. It is not hard to show that the feature distance is the lower bound of the minimum distance.

$$
L_2^{Euclidean}(\overrightarrow{FA}) \leq L_2^{minimum}(A)
$$

In practice, however, the segmented mean variation feature vector $\overrightarrow{FA}$ is a poor approximation of the time sequence $A$, since it is essentially a down-sampling of time sequence. Much of the information will be lost, consequently, too many false alarms will occur. It is necessary to find a way to compensate the loss of information so that we could reduce the number of false alarms. More specifically, we find a factor $\alpha > 1$ such that,

$$
\alpha \cdot L_2^{Euclidean}(\overrightarrow{FA}) \leq L_2^{minimum}(A)
$$

Since $F(\cdot) = |\cdot|^2$ is a convex function on $R$, we can use the property of convex functions to find $\alpha$. There is a well-known mathematical result on convex functions. We can use the following theorem from [26,28]

**Theorem 1.** *Suppose that $x_1, x_2, \ldots, x_n \in R$, and $\lambda_1, \lambda_2, \ldots, \lambda_n \in R$ such that $\lambda_i \geq 0$ and $(\sum_{i=1}^{n} \lambda_i) = 1$. If $F()$ is a convex function on $R$, then*

$$
F(\lambda_1 x_1 + \lambda_2 x_2 + \ldots + \lambda_n x_n) \leq \lambda_1 F(x_1) + \lambda_2 F(x_2) + \ldots + \lambda_n F(x_n)
$$

*where $R$ is the set of real numbers.*

*Proof.* The proof is shown in [28]. □

Using the result of Theorem 1 by taking $\lambda_i = \frac{1}{n}$, we have the following corollary.

**Corollary 1** *For any time sequence $A = (a_1, a_2, \ldots, a_n)$, the following holds.*

$$\begin{aligned}
&(n-1) \cdot |MeanVariation(A)|^2 \\
&\quad \le \sum_{i=1}^{n-1} |a_{i+1} - a_i|^2 \\
&\quad \le 2 \cdot (\sum_{i=1}^{n-1} |a_i - m_a|^2 + \sum_{i=2}^{n} |a_i - m_a|^2) \\
&\quad \le 2^2 \cdot \sum_{i=1}^{n} |a_i - m_a|^2 \\
&where \quad m_a = \sum_{i=1}^{n} (a_i)/n
\end{aligned}$$

*or, equivalently, for each segment $S_j$, $1 \le j \le m$, we have*

$$\begin{aligned}
&(l-1) \cdot |MeanVariation(S_j)|^2 \\
&\quad \le \sum_{i=(j-1)l+(2-j)}^{j(l-1)} |a_{i+1} - a_i|^2 \\
&\quad \le 2 \cdot (\sum_{i=(j-1)l+(2-j)}^{j(l-1)} |a_i - m_a|^2 + \sum_{i=(j-1)l+(3-j)}^{j(l-1)+1} |a_i - m_a|^2) \\
&where \quad m_a = \sum_{i=1}^{n} (a_i)/n
\end{aligned}$$

Now, we present our main theorem as follows.

**Theorem 2.** *For any time sequence $A = (a_1, a_2, \ldots, a_n)$, the following holds.*

$$\begin{aligned}
\sqrt{l-1} \cdot L_2^{Euclidean}(\overrightarrow{FA}) &\le 2 \cdot (\sum_{i=1}^{n} |a_i - m_a|^2)^{1/2} \\
&= 2 \cdot L_2^{minimum}(A) \\
where \quad m_a = \sum_{i=1}^{n} (a_i)/n &
\end{aligned}$$

*Proof.* Based on the definitions of $L_2$ and $\overrightarrow{FA}$,

$$(l-1)\cdot L_2^{Euclidean}(\overrightarrow{FA})^2 = (l-1)\cdot\sum_{j=1}^{m}|MeanVariation(S_j)|^2$$

By Corollary 1, we can get the following inequality.

$$\begin{aligned}
&(l-1)\cdot\sum_{j=1}^{m}|MeanVariation(S_j)|^2\\
&\quad\le 2\cdot\{\sum_{i=1}^{l-1}|a_i-m_a|^2+\sum_{i=2}^{l}|a_i-m_a|^2\}\\
&\quad+2\cdot\{\sum_{i=l}^{2(l-1)}|a_i-m_a|^2+\sum_{i=l+1}^{2(l-1)+1}|a_i-m_a|^2\}\\
&\quad+\ldots\\
&\quad+2\cdot\{\sum_{i=(j-1)l+(2-j)}^{j(l-1)}|a_i-m_a|^2+\sum_{i=(j-1)l+(3-j)}^{j(l-1)+1}|a_i-m_a|^2\}\\
&\quad\le 2^2\cdot\sum_{i=1}^{n}|a_i-m_a|^2\\
&\quad= 2^2\cdot L_2^{minimum}(A)^2\\
&where\quad m_a=\sum_{i=1}^{n}(a_i)/n
\end{aligned}$$

By taking square root of both sides, we prove the theorem. □

Owing to Theorem 2, we can efficiently handle minimum distance queries with the segmented mean variation features. We know that $L_2^{minimum}(A,B)\le\epsilon$ represents $L_2^{Euclidean}(\overrightarrow{FA},\overrightarrow{FB})\le 2\cdot\epsilon/\sqrt{l-1}$ by Theorem 2. Consequently, minimum distance queries can be correctly converted to simple Euclidean distance queries in the feature space without false dismissals. We can reduce the search space by a factor of $2/\sqrt{l-1}$.

## 5 Query Processing

In this section, we will first present the sequential scanning which is the simplest searching algorithm for time sequence matching. Then, we will present the overall process of the SMV-indexing scheme.

### 5.1 Sequential Scanning

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

**Algorithm 1:** Sequential Scanning

```
begin
    Input: query time sequence $Q$, error bound $\epsilon$
    Output: data time sequences within error bound $\epsilon$
    Result $\leftarrow$ NULL;
    for all $D_i \in$ databases do
        if ComputeMinimumDistance($Q$, $D_i$)$\leq \epsilon$ then
            Result $\leftarrow D_i \bigcup$ Result;
        else
            Reject $D_i$;
        end
    end
    return Result;
end
```

### 5.2 Segmented Mean Variation Indexing (SMV-Indexing)

We present the overall process of our time series indexing scheme. Before a query is performed, we shall do some preprocessing to extract feature vectors from time sequences, and then to build an index. After the index is built, the similarity search can be performed to select candidate sequences from databases.

#### Pre-processing

•**Step 1. Feature Extraction :** Each time sequence is divided into $m$ segments. Then, the feature vector is extracted from the time sequence using the dimensionality reduction technique mentioned in Section 4.1 for all time sequences in databases.

•**Step 2. Index Construction :** We build a multidimensional index structure such as R-tree using the feature vectors extracted from time sequences.

**Index Searching** After an index structure has been built, we can perform the similarity search against a given query time sequence. The searching 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 results of candidate selection because the feature distance is the lower bound of the minimum distance. The actual minimum distance between query time sequence and candidate sequences are computed and only those within the error bound are reported as the query results. The implementation of the SMV-indexing is described in Algorithm 2.

**Algorithm 2:** SMV-indexing

```
begin
    Input: query time sequence $Q$, error bound $\epsilon$
    Output: data time sequences within error bound $\epsilon$
    Result ← NULL;
    Candidate ← NULL;
    // project the query time sequence $Q$ into the index space,
    $\overrightarrow{FQ}$ ← FeatureExtraction($Q$);
    // candidate selection using a Spatial Access Method,
    Candidate ← IndexSearching($\overrightarrow{FQ}$,SAM,$\frac{2\cdot\epsilon}{\sqrt{l-1}}$ );
    // postprocessing to remove false alarms,
    for all $C_i \in$ Candidate do
        if ComputeMinimumDistance($Q$, $C_i$)$\leq \epsilon$ then
            Result ← $C_i \bigcup$ Result;
        else
            Reject $C_i$;
        end
    end
    return Result;
end
```

## 6 Performance Evaluation

We implemented both the SMV-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 disk page is set to 4KB. For a spatial access method, we used the Katayama's R*-tree source codes[1].

The real time sequences were obtained from Seoul Stock Market, Korea[2]. The stock database consisted of 2000 stocks and for each stock its daily closing prices for 128 days. We have run 25 random queries over real dataset to find similar time sequences based on the minimum distance. The query time sequences were randomly selected from the stock database.

First, we compared the SMV-indexing with the sequential scanning in terms of the number of actual computations required by varying error bound. 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 follows.

$$search\ space\ ratio = \frac{the\ number\ of\ candidate\ sequences}{the\ number\ of\ time\ sequences\ in\ databases}$$

Figure 3 shows the experimental result. Figure 3 shows that there is a significant gain in reducing search space by the SMV-indexing.

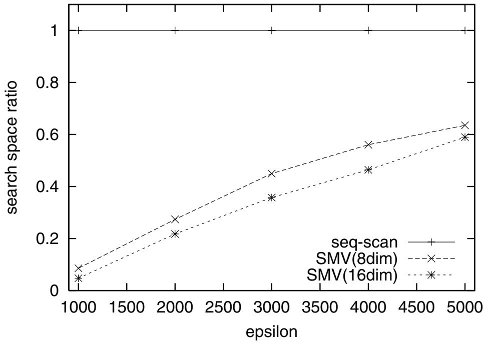


**Fig. 3.** Search Space Ratio: SMV-indexing vs. Sequential Scanning

Good filtering effect dose not always result in good performance since the spatial access method can be affected by the dimensionality of feature vectors. To verify the performance of the SMV-indexing, we compared the SMV-indexing with the sequential scanning with respect to the average execution time. Figure 4 shows the experimental result. The result shows that the SMV-indexing is more efficient than the sequential scanning in performance. However, as shown in

[1] The Katayama's R*-tree source codes can be retrieved from `http://research.nii.ac.jp/~katayama/homepage/research/English.html`

[2] This data can be retrieved from `http://www.kse.or.kr/kor/stat/stat_data.htm`

Figure 4, the average execution time increases as the given error bound increases, eventually, the performance of the SMV-indexing becomes worse than that of the sequential scanning. 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.

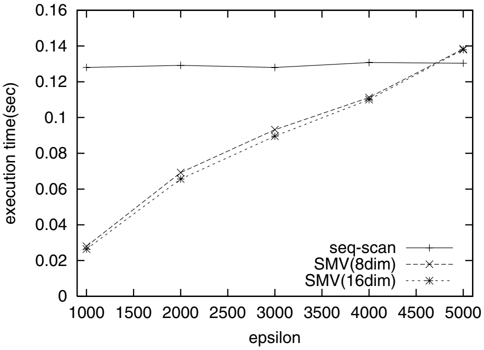


**Fig. 4.** Execution time: SMV-indexing vs. Sequential Scanning

## 7 Conclusion

We focused on fast similarity search in large time series databases, when the similarity measurement is the minimum distance. The Euclidean distance is the most popular similarity measurement for sequence matching. However, the Euclidean distance has the potential drawback that it cannot support the shape queries based on the minimum distance. To support minimum distance queries, most of 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 time series indexing scheme that supports minimum distance queries without vertical shifting. Our indexing scheme is motivated by autocorrelation of time sequence, that is, the mean variation of time sequence is invariant under vertical shifting.

The major contributions of this work are: 1) introducing the similarity matching scheme that allows indexing of time series based on the minimum distance without vertical shifting, 2) introducing lower bound of the minimum distance to filter out dissimilar time sequences without false dismissals.

Our approach can be extended to the matching time series of similar shapes in arbitrary $L_p$ norm. We are currently working in this direction for fast retrieving and matching time series of similar shapes in arbitrary $L_p$ norm.

## References

1. Rakesh Agrawal, Tomasz Imielinski and Arun N. Swami: Database Mining: A Performance Perspective. IEEE TKDE, Special issue on Learning and Discovery in Knowledge-Based Databases 5-6(1993) 914-925 377
2. Usama M. Fayyad, Gregory Piatetsky-Shapiroa and Padhraic Smyth: Knowledge Discovery and Data Mining: Towards a Unifying Framework. In Proc. of International Conference on Knowledge Discovery and Data Mining(1996) 82-88 377
3. A. Guttman : R-trees: A Dynamic Index Structure for Spatial Searching. In Proc. of SIGMOD Conference on Management of Data(1984) 47-57 378
4. N. Beckmann, H. P. Kriegel, R. Schneider and B. Seeger: The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. In Proc. of SIGMOD Conference on Management of Data(1990) 322-331 378
5. Rakesh Agrawal, Christos Faloutsos and Arun N. Swami: Efficient Similarity Search In Sequence Databases. In Proc. of International Conference on Foundations of Data Organization and Algorithms(1993) 69-84 377, 379
6. Christos Faloutsos, M. Ranganathan and Yannis. Manolopoulos: Fast Subsequence Matching in Time-Series Databases. In Proc. of SIGMOD Conference on Management of Data(1994) 419-429 377, 379
7. Dina Q. Goldin, Paris C. Kanellakis: On Similarity Queries for Time-Series Data: Constraint Specification and Implementation. In Proc. of International Conference on Principles and Practice of Constraint Programming(1995) 137-153 378, 379
8. Rakesh Agrawal, King-Ip Lin, Harpreet S. Sawhney and Kyuseok Shim: Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases. In Proc. of International Conference on Very Large Data Bases(1995) 490-501 379
9. Chung-Sheng Li, Philip S. Yu and Vittorio Castelli: HierarchyScan: A Hierarchical Similarity Search Algorithm for Databases of Long Sequences. In Proc. of International Conference on Data Engineering(1996) 546-553 379
10. Flip Korn, H. V. Jagadish and Christos Faloutsos: Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences. In Proc. of SIGMOD Conference on Management of Data(1997) 289-300 379
11. Davood Rafiei, Alberto O. Mendelzon: Similarity-Based Queries for Time Series Data. In Proc. of SIGMOD Conference on Management of Data(1997) 13-25 380
12. Tolga Bozkaya, Nasser Yazdani and Z. Meral Ozsoyoglu: Matching and Indexing Sequences of Different Lengths. In Proc. of International Conference on Information and Knowledge Management(1997) 128-135 379
13. Nasser Yazdani and Z. Meral Ozsoyoglu: Sequence Matching of Images. In Proc. of International Conference on Scientific and Statistical Database Management(1996) 53-62 379
14. Gautam Das, Dimitrios Gunopulos and Heikki Mannila: Finding Similar Time Series. In Proc. of European Conference on Principles of Data Mining and Knowledge Discovery(1997) 88-100 379
15. Bela Bollobas, Gautam Das, Dimitrios Gunopulos and Heikki Mannila: Time-Series Similarity Problems and Well-Separated Geometric Sets. In Proc. of Symposium on Computational Geometry(1997) 454-456 379
16. Byoung-Kee Yi, H. V. Jagadish and Christos Faloutsos: Efficient Retrieval of Similar Time Sequences Under Time Warping. In Proc. of International Conference on Data Engineering(1998) 201-208 380

17. Davood Rafiei and Alberto O. Mendelzon: Efficient Retrieval of Similar Time Sequences Using DFT. In Proc. of International Conference on Foundations of Data Organization and Algorithms(1998) 377
18. Sze Kin Lam, Man Hon Wong: A Fast Projection Algorithm for Sequence Data Searching. Data and Knowledge Engineering 28-3(1998) 321-339 378, 379
19. Kelvin Kam Wing Chu, Sze Kin Lam and Man Hon Wong: An Efficient Hash-Based Algorithm for Sequence Data Searching. The Computer Journal 41-6(1998) 402-415 378, 379
20. Kelvin Kam Wing Chu, Man Hon Wong: Fast Time-Series Searching with Scaling and Shifting. In Proc. of Symposium on Principles of Database Systems(1999) 237-248 379
21. Davood Rafiei: On Similarity-Based Queries for Time-Series Data. In Proc. of International Conference on Data Engineering(1999) 410-417 380
22. Kin-pong Chan, Ada Wai-chee Fu: Efficient Time Series Matching by Wavelets. In Proc. of International Conference on Data Engineering(1999) 126-133 378, 380
23. Eamonn J. Keogh, Michael J. Pazzani: A Simple Dimensionality Reduction Technique for Fast Similarity Search in Large Time Series Databases. In Proc. of Pacific-Asia Conference on Knowledge Discovery and Data Mining(2000) 122-133 378, 380
24. Sanghyun Park, Wesley W. Chu, Jeehee Yoon and Chihcheng Hsu: Efficient Searches for Similar Subsequences of Different Lengths in Sequence Databases. In Proc. of International Conference on Data Engineering(2000) 23-32 380
25. Chang-Shing Perng, Haixun Wang, Sylvia R. Zhang and D. Stott Parker: Landmarks: a New Model for Similarity-based Pattern Querying in Time Series Databases. In Proc. of International Conference on Data Engineering(2000) 33-42 380
26. Byoung-Kee Yi, Christos Faloutsos: Fast Time Sequence Indexing for Arbitrary Lp Norms. In Proc. of International Conference on Very Large Data Bases(2000) 385-394 380, 383
27. Eamonn J. Keogh, Kaushik Chakrabarti, Sharad Mehrotra and Michael J. Pazzani: Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. In Proc. of SIGMOD Conference on Management of Data(2001) 151-162 378, 380
28. M. H. Protter and C. B. Morrey: A First Course in Real Analysis. Springer-Verlag(1977) 383