LETTER

# Physical Database Design for Efficient Time-Series Similarity Search

Sang-Wook KIM[†a)], Jinho KIM[††], *Nonmembers, and* Sanghyun PARK[†††], *Member*

**SUMMARY** *Similarity search* in time-series databases finds such data sequences whose changing patterns are similar to that of a query sequence. For efficient processing, it normally employs a multi-dimensional index. In order to alleviate the well-known *dimensionality curse*, the previous methods for similarity search apply the *Discrete Fourier Transform (DFT)* to data sequences, and take only the first two or three DFT coefficients as *organizing attributes*. Other than this ad-hoc approach, there have been no research efforts on devising a systematic guideline for choosing the best organizing attributes. This paper first points out the problems occurring in the previous methods, and proposes a novel solution to construct optimal multi-dimensional indexes. The proposed method analyzes the characteristics of a target time-series database, and identifies the organizing attributes having the best *discrimination power*. It also determines the optimal number of organizing attributes for efficient similarity search by using a cost model. Through a series of experiments, we show that the proposed method outperforms the previous ones significantly.

***key words:*** *time-series databases, similarity search, multi-dimensional indexes*

## 1. Introduction

A *time-series* database is a set of data sequences, each of which is an ordered list of elements. *Similarity search* is an operation that searches for the sequences or subsequences whose changing patterns are similar to that of a given query sequence [1], [2], [7]. For example, consider a time-series database that stores the stock history data. Similarity search is of growing importance in many new applications such as data mining and data warehousing [6].

Most approaches for similarity search regard a sequence with $n$ elements as a point in $n$-dimensional space, and define the basic similarity of the two sequences by using the *Euclidean distance* between their corresponding points [1], [7], [10]. They also use a *multi-dimensional index* such as the $R$-tree, $R^*$-tree, and $R^+$-tree for efficient handling of multi-dimensional points. To avoid the *dimensionality curse* [1], [4], [8] in a multi-dimensional index, they usually apply the *Discrete Fourier Transform (DFT)* [9] to data sequences, and select the first two or three DFT coefficients as organizing attributes of the multi-dimensional index [1], [7].

However, there have been no research efforts to devise a systematic approach for selecting the best organizing attributes from a set of DFT coefficients. This paper mainly focuses on this issue. First, we point out the performance problem of the previous methods, and then propose a novel method for constructing a multi-dimensional index as its solution. The proposed method analyzes the characteristics of a target time-series database, and then selects the organizing attributes of the multi-dimensional index with a high discrimination power. It also determines the optimal number of organizing attributes for efficient similarity search using the proposed cost model. We compare the proposed method with the previous ones to verify its effectiveness.

## 2. Previous Work

### 2.1 Similarity Search

The similarity measure of any two sequences, $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$, widely-used in a time-series database, is the *Euclidean distance* $D(X, Y)$ [1], [7] defined as follows:

$$D(X, Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{1}$$

Similarity search is defined as the operation that finds such data sequences $Y$ whose Euclidean distances to a query sequence $X$ are within a given distance tolerance $\varepsilon$.

### 2.2 Index Construction

Typically, previous methods [1], [2], [7], [10] regard sequences with $n$ elements as points in $n$-dimensional space, and utilize a multi-dimensional index for indexing them. The storage and computation costs of a multi-dimensional index grow exponentially with $n$ [4], [8]; The multi-dimensional index suffers from serious performance degradation when $n$ is large, which is general in a time-series database [1], [2]. We call it *dimensionality curse*.

To overcome such a problem, the previous work usually depends on dimensionality reduction techniques using the *Discrete Fourier Transform (DFT)* and the *Discrete Cosine Transform (DCT)* [9]. The DFT converts a sequence $X = (x_1, x_2, ..., x_n)$ of *real-valued* elements in time domain into a sequence $X_F = (x_{F_1}, x_{F_2}, ..., x_{F_n})$ of *complex-valued*

Manuscript received August 7, 2006.
Manuscript revised October 29, 2007.

[†] The author is with the Division of Information and Communications, Hanyang University, 17 Haengdang-dong, Sungdong-gu, Seoul, 133-791, Korea.
[††] The author is with the Department of Computer Science, Kangwon National University, Korea.
[†††] The author is with the Department of Computer Science, Yonsei University, Korea.
a) E-mail: wook@hanyang.ac.kr

DOI: 10.1093/ietcom/e91–b.4.1251