Grid Representation of Time Series Data for Similarity Search

Guifang Duan Yu Suzuki Kyoji Kawagoe
Graduate School of Science and Engineering
Ritsumeikan University
gr042046@se. ritsumei.ac.jp yusuzuki@is.ritsumei.ac.jp kawagoe@is.ritsumei.ac.jp

ABSTRACT

Widespread interest in time-series similarity search has made more in need of efficient technique, which can reduce dimensionality of the data and then to index it easily using a multidimensional structure. In this paper, we introduce a technique, which we called grid representation, based on a grid approximation of the data. We propose a lower bounding distance measure that enables a bitmap approach for fast computation and searching. We also show how grid representation can be indexed with a multidimensional index structure, and demonstrate its superiority.

Keywords

Dimensionality Reduction, Grid Representation, Index, Similarity Search

1. INTRODUCTION

There have been much interests and research work on time series data. Widespread interest in time-series similarity search has made more in need of efficient technique, which can reduce dimensionality of the data and then to index it easily using a multidimensional structure, and it has been proved to be an efficient and promising way [2,5,7,9,10,25]. It is described in [28] that "the key to the efficiency and accuracy of the solution is to choose an appropriate data representation method". Many representation techniques for time series data have already been proposed, including Discrete Fourier Transform (DFT) [1], Discrete Wavelet Transform (DWT) [8,12,13], Singular Value Decomposition (SVD) [14,15], Piecewise Aggregate Approximation (PAA) [5,11], Symbolic Aggregate approXimation (SAX) [6, 16], etc. In this paper, we introduce a grid representation technique for time series dimensionality reduction and indexing. Our proposed method which we call Grid Representation is a completely competitive technique, as we will show in this paper.

We usually use a two-dimensional time-value space to process the time series, and most of existing representation techniques are based on this space. In [29, 30], the authors proposed a grid space to index the time series, which is completely different from a two-dimensional time-value space in other methods. Our method is also based on a grid space, by which a bitmap measure can be adopted for efficient computation, and the index can easily be constructed, alike the proposed method in [29,30].

The rest of this paper is organized as follows. After introducing the grid representation and the distance measures defined on it with some background research in Section 2, we discuss the dimensionality reduction approach in Section 3. Then we illustrate the grid bitmap approach of computation in Section 4. In section 5, the index construction is presented. Section 6 contains an experimental evaluation. Finally, in Section 7, we summarize this work.

2. THE GRID REPRESENTATION

Our proposed representation works by putting each time series into a grid space. Figure 1 gives the intuition of our idea.

Given a time series $C = \{c_1,\ldots,c_n\}$, we are able to produce an grid representation as

$$C = \{<Cg_1, Cgp_1>,\ldots,<Cg_n, Cgp_n>\} \quad (1)$$

where Cg_i is the row number, in which the grid involving the i^{th} point is seated (i.e. Cg_i is a natural number, such as 1, 2,3…) and the Cgp_i only can be either 1 or 0. We will discuss how the value of Cgp_i is set in the following subsection.

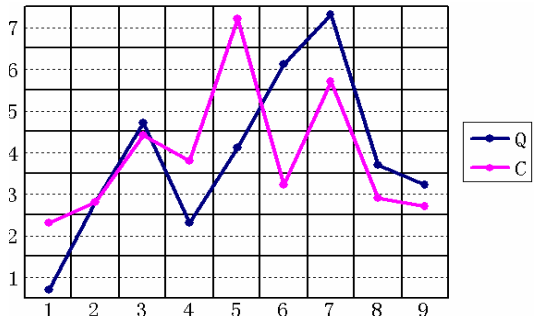

Figure 1: Grid Space, in which there are two time series Q and C with the same length of 9. Their grid representations respectively are *Q* = {<1, 0>, <3, 0>, <5, 0>, <2, 1>, <4, 1>, <6, 1>, <7, 1>, <4, 0>, <3, 1>} and *C* I {<2, 1>, <3, 0>, <4, 1>, <4, 0>, <7, 1>, <3, 1>, <6, 0>, <3, 0>, <3, 0>}

2.1 The Grid Space.

Basically we always use a two-dimensional value-time space to locate time series data. However, in our method we make a little change of the space usage. As shown in the figure 1, we divide the two-dimensional value-time space into many snuggled rectangles with no overlaps and with the same size by two mutually orthogonal equidistant sets of lines that parallel to vertical axis and to time axis respectively. The distance between the adjacent lines paralleling to vertical axis, called the length (see figure 2 (a)) of the grid, is easy to fix, just equal to the time axis directional distance between two adjacent original points in a time series. So we can guarantee that all points in the same time sequence can be located in different columns. The distance between the adjacent lines paralleling to time axis, called the width (see figure 2 (a)) of the grid, depends on the extent of the time series. For example, assume that the difference between maximum and minimum values of a time series is R, and the number of the rows is m. Then, the width should be R/m. It goes without saying that m depends on the application and that larger m causes better quality, but lower speed for similarity-based searching.

Having transformed a two-dimensional value-time space into a grid space, every point of a time series will be involved in one grid in the grid space. For more definite position of the point in a grid, we divide the grid in two half-size rectangles by a middle line, as shown in the figure 2, if there is a point in it. If the i-th point of a time series data is seated in the upper rectangle in the grid, Cgp_i is set to 1, conversely, otherwise, the Cgp_i is set to 0, as shown in the figure 2

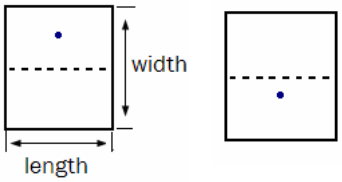

Figure 2: the grid is divided into two same rectangles. When the point is in the upper one, like (a), the Cgp is 1, otherwise, like (b), the Cgp will be 0.

2.2. Lower Bounding Euclidean Distance

Firstly we will review some existing definition on the distance measure, before we define the lower bounding distance for our grid representation.

Suppose that we have two time series data, a query $Q = \{q_1,\ldots,q_n\}$ and a data set in DB $C = \{ c_1,\ldots,c_n \}$, and we compare these two time series data set, we can use the following ubiquitous Euclidean distance as described in such as.

$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n}(Q_i - C_i)^2} \qquad (2)$$

As usual, we remove the square root calculation to in the above Euclidean distance function, because the square root function is monotonic and concave, which can be shown in [28].

$$D(Q,C) \equiv \sum_{i=1}^{n}(Q_i - C_i)^2 \qquad (3)$$

According to Ratanamahatana, Keogh, Bagnall, and Lonardi [28] "other optimizations are allowed to slightly speed up the calculations or get better quality while working with this latter distance measure".

Then we will define the square distance for the grid representation, which can lower bound the square Euclidean distance. Now suppose that Q and C are the grid representation of Q and C respectively, where $Q = \{<Qg_1, Qgp_1>,\ldots,<Qg_n, Qgp_n>\}$, $C = \{<Cg_1, Cgp_1>,\ldots,<Cg_n, Cgp_n>\}$. LB_grid($Q$,$C$), lower bound distance, is defined as:

$$\text{LB\_grid}(Q,C) = \sum_{i=1}^{n} \begin{cases} (Qg_i - Cg_i)^2 Hg^2 & \text{if } (Qg_i - Cg_i)(Qgp_i - Cgp_i) > 0 \\ (Qg_i - Cg_i - 1)^2 Hg^2 & \text{if } Qg_i - Cg_i \neq 0, \text{ and} \\ & (Qg_i - Cg_i)(Qgp_i - Cgp_i) \leq 0 \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

where Hg is the width of the grid, which we defined before. We will prove the claim of lower bounding the grid representation.

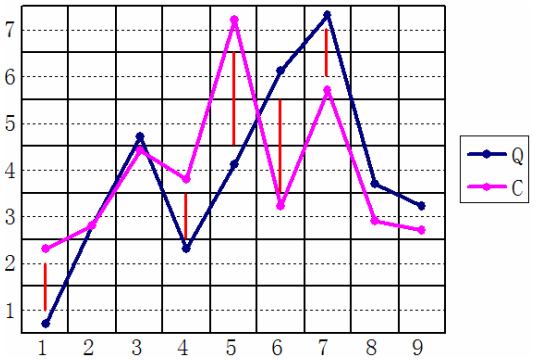

Figure 3: The lower bounding function LB_grid(Q,C). The red lines deputize for the distance between Q and C

Proposition 1: LB_grid(Q,C) lower bounds the square Euclidean distance between original subsequences.

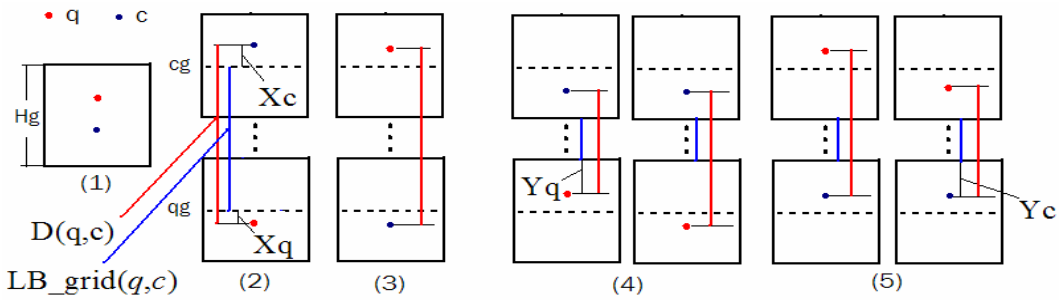

Figure 4: The 5 possible cases of two points q in time series Q and c in time series C for distance calculation in the grid space. D(q, c) is squared length of the red line, and LB_grid(q,c) is the squared length of the blue line.

Proof. We present a proof for the case of only one point in the grid representation. The more general proof for the n points case can be obtained by applying the proof to each of the n points.

Let q and c is one point of two time series Q and C, q and c is the grid representation of q and c respectively. Firstly we note that by function (3), D(q,c) $\geqslant$ 0. We then consider the five possible cases.

(1) $qg = cg$

(2) $qg < cg$, $cgp = 1$, $qgp = 0$

(3) $qg > cg$, $cgp = 0$, $qgp = 1$

(4) $qg < cg$, $cgp = 0$, $qgp = 1$ or 0

(5) $qg > cg$, $qgp = 1$, $cgp = 1$ or 0

In case (1), by definition (4), LB_grid(q,c) = 0, hence LB_grid(q,c) $\leqslant$ D(q,c).

In case (2) and (3), LB_grid(q,c) = $(qg - cg)^2 Hg^2$. We assume the distance q and c to the middle line in the grid is Xq and Xc, obviously Xq $\geqslant$ 0, Xc $\geqslant$ 0.

Then D(q,c) = ($|qg - cg|$ Hg + Xc +Xq)^2,

Because Hg is the width of the grid,

Hg > 0

Since Xq $\geqslant$ 0, Xc $\geqslant$ 0 and $|qg - cg| > 0$

Then ($|qg - cg|$ Hg + Xc +Xq)^2 $\geqslant$ $[(qg - cg)Hg]^2 = (qg - cg)^2 Hg^2$

So LB_grid(q,c) $\leqslant$ D(q,c).

The proof of the cases left is very similar with above, just instead the Xq and Xc by Yq and Yc, which is the distance of q and c to the according top or bottom of the grid (showed in figure 4). So here we will not give more proof, it is easy to get LB_grid(q,c) $\leqslant$ D(q,c) in case (4) and (5). This concludes the proof.

Figure 4 illustrates the visual intuition of the proof. The extension of this proof to the Euclidean distance is trivial, and will be omitted due to space limitations.

3. DIMENSIONALITY REDUCTION

In order to make it easy to understand the proposed method, we just describe how to directly generate grid presentation based on the original time series. However, our grid technique also allows a time series data with arbitrary n points to be reduced to a sequence with arbitrary 2m points, where 2m < n, typically 2m << n, by adding an intermediate step in a process of generating grid representation from the original time series. This step is used to reduce the dimensionality efficiently. We utilize the PAA at the intermediate step, which we review in the next section.

3.1 A Brief Review of PAA

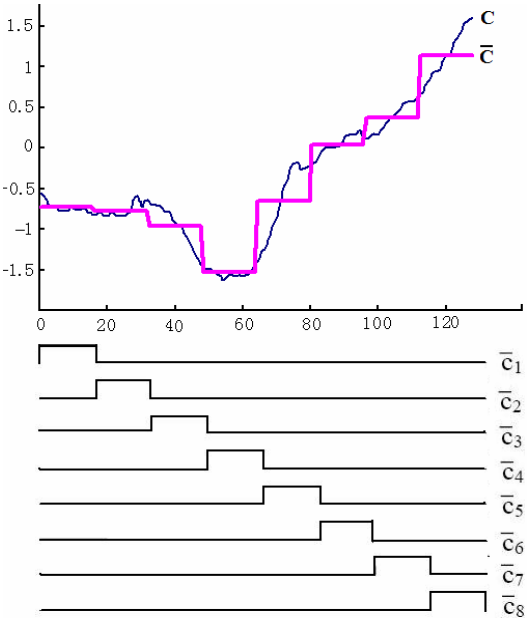

Figure 5: The PAA Representation. This original figure is taken from [5].

A time series C of length n can be represented in an N-dimensional space by a

vector $\overline{C} = \overline{c}_1, \ldots, \overline{c}_N$ .The i^{th} element of $\overline{C}$ is calculated by the following equation:

$$\overline{c}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{N}{n}j} c_j \qquad (5)$$

It is very simple that we divide the data into N equal-length frames and calculate the mean value of the data in them. The $\overline{c}_i$ is just the vector of these value. Thus the dimensionality is reduced from n to N.

Keogh, Chakrabarti. Pazzani, and Mehrotra [5] use a good way of visualization as "approximating the original time series with a linear combination of box basis function", which is shown in Figure5. The complicated subscripting in Eq. 5 insures that the original sequence is divided into the correct number and size of frames.

The PAA transformation is much faster in computing, can be defined for arbitrary length queries, and is able to handle many different distance measures [17,18]. It has been shown that the PAA is strongly competitive with more sophisticated dimensionality reduction techniques like Fourier transforms and wavelet [5,8,11].

3.2 The Grid Representation Based On Dimensionality Reduction

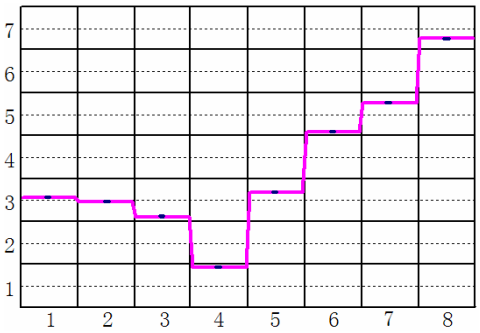

Figure 6: A PAA representation is transformed to grid representation. In this case, we take the middle point of every segment in $\overline{C}$ which we showed in Figure5, then put them in to the grid space whose grid length is just equal to the length of the segments. Thus, we can get the dimensionality-reduced grid representation of C easily as *C* = {<3, 1>, <3, 0>, <3, 0>, <1, 1>, <3, 1>, <5, 0>, <5, 1>, <7, 0>}

We can apply a further transformation to obtain a dimensionality-reduced grid representation after transforming a time series data into the PAA. Suppose a time series data C of length n is represented $\overline{C} = \overline{c}_1, \ldots, \overline{c}_N$ by a PAA representation, we just take the middle point of every segment $\overline{c}_i$ to stand for the segment. Then we put these N points to in the grid space appropriately whose grid length is equal to that of the PAA segment, thus the dimensionality- reduced grid representation of the time series data can easily be obtained as $Q = \{<Qg_1, Qgp_1>, \ldots, <Qg_N, Qgp_N>\}$, $C = \{<Cg_1, Cgp_1>, \ldots, <Cg_N, Cgp_N>\}$. Figure 6 illustrates this notation. For the rest of the paper, we just say grid representation, which always means dimensionality-reduced grid representation.

3.3 Dimensionality-Reduced Grid Representations Distance Measure

We can define a new square distance measure RLB_grid(*Q*,*C*) between grid representation $Q = \{<Qg_1, Qgp_1>, \ldots, <Qg_N, Qgp_N>\}$, $C = \{<Cg_1, Cgp_1>, \ldots, <Cg_N, Cgp_N>\}$ of a query $Q = \{q_1, \ldots, q_n\}$, a candidate match $C = \{c_1, \ldots, c_n\}$ as the following format:

$$\text{RLB\_grid}(Q,C) = \sum_{i=1}^{n} \begin{cases} \frac{n}{N}(Qg_i - Cg_i)^2 Hg^2 & \text{if } (Qg_i - Cg_i)(Qgp_i - Cgp_i) > 0 \\ \frac{n}{N}(Qg_i - Cg_i - 1)^2 Hg^2 & \text{if } Qg_i - Cg_i \neq 0, \text{ and } (Qg_i - Cg_i)(Qgp_i - Cgp_i) \leq 0 \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

This square distance is also lower bounds the square Euclidean distance between original subsequences. Suppose $\overline{Q}$ and $\overline{C}$ are the PAA representations of Q and C, then we can obtain a lower bounding distance of the square Euclidean distance between original subsequences [5]:

$$\text{SDR}(\overline{Q}, \overline{C}) \equiv \frac{n}{N} \sum_{i=1}^{N} (\overline{q}_i - \overline{c}_i)^2 \qquad (7)$$

In fact, this distance is also the square Euclidean distance between $\overline{Q}$ and $\overline{C}$. Hereby RLB_grid(*Q*,*C*) lower bounds $\text{SDR}(\overline{Q}, \overline{C})$ according Proposition1. That is to say:

$$\text{RLB\_grid}(Q,C) \leq \text{SDR}(\overline{Q}, \overline{C}) \leq \text{D}(Q, C) \qquad (8)$$

where D(Q, C) is the square Euclidean distance between original subsequences which is showed in Equ.3. Therefore, we can have

Proposition 2: RLB_grid(*Q*,*C*) lower bounds the square Euclidean distance between original subsequences.

4. EFFICIENT GRID BITMAP APPROACH

In our distance function described in Eq. (6) above, n/N and Hg^2 are constant, and Qg_i and Cg_i are positive integers which are not so large, usually smaller than 100, because the useful range of for

most spatial index structures is just 8 to 20 dimensionalities [19,20]. Thus the computation will become much easier. In order to make the distance function more competitive and the computation much faster, we will propose a grid bitmap approach to find those Qg_i which are not equal to their corresponding Cg_i. Since in Eq. (6), we define the distance between point q_i and c_i is set to 0 when $Qg_i = Cg_i$, i.e. q_i and c_i are located in the same grid, so it is easier and faster to compute the distance if we know when $Qg_i \neq Cg_i$. And a major advantage of bitmap approach is that bitmap manipulations using bit-wise operators (AND, OR, XOR, NOT) can be very simple and very efficient, usually supported by hardware. Our grid bitmap is defined as the following:

Given a grid transformation, in the grid space we mark 1 in a grid if there is a point in it, else we mark 0.

For example, we transform the grid space in Figure 6 to a grid bitmap by the definition as shown in Figure 7.

7	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	1	0
4	0	0	0	0	0	0	0	0
3	1	1	1	0	1	0	0	0
2	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0
	1	2	3	4	5	6	7	8

Figure 7: grid bitmap

Then if we match two grid bitmaps of a query Q and a time series C by applying the XOR operator, the result will show in which column of the grid space $Qg_i \neq Cg_i$. Figure 8 illustrates the processing.

Like shown in Figure 8, the grid representation of Q is Q = {<2, 0>, <1, 1>, <2, 1>, <2, 0>}, that of C is C = {<2, 0>, <1, 0>, <1, 0>, <2, 1>}. We transform them to grid bitmaps, then match two bitmaps by applying XOR operator, the result is that only in column 3 there are two "1"s, which means only $Qg_3 \neq Cg_3$. So when we compute RLB_grid(Q,C), we can just focus on column 3. Since $Qg_3 = 2$, $Qgp_3 = 1$, $Cg_3 = 1$ and $Cgp_3 = 0$, $(Qg_3 - Cg_3)(Qgp_3 - Cgp_3) = 1 > 0$. Hence,

$$\text{RLB\_grid}(Q,C) = \frac{n}{N}(Qg_3 - Cg_3)^2 Hg^2 = \frac{n}{N} Hg^2$$

where N is always 4, n is the length of original time series Q and C, Hg is the width of the grid, also a constant.

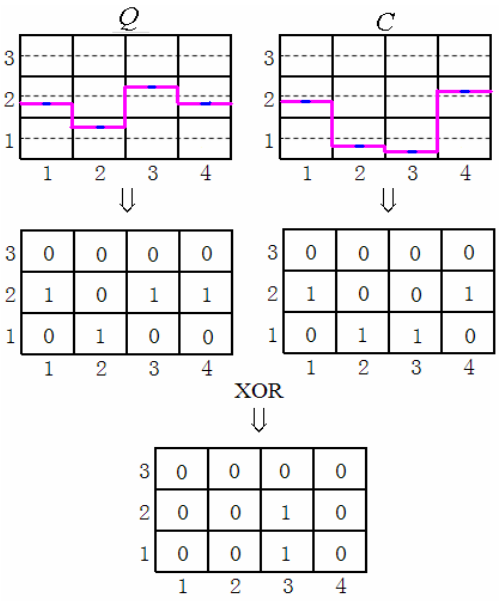

Figure 8: Processing of grid bitmap operation

5. INDEX CONSTRUCTION

In this section we describe how to index grid presentation using a multidimensional structure, such as R-tee. We define a Grid Bounding Region (GBR) and a lower bounding distance measure for it instead of traditional Minimum Bounding Rectangle (MBR) [24] for the index.

5.1 Grid Bounding Regions

Given group *SC* of n grid representations of length 2N, $SC = \{C_1,\ldots, C_n\}$, the Grid Bounding Region of *SC* is defined like Skyline Bounding Region (SBR) in Skyline index [21], a two-dimensional region surrounded by top and bottom skylines and two vertical lines connecting the two skylines at the start and end times. The top (TS) and bottom (BS) skylines of *S* are defined as follows:

$TS = \{<TSg_1, TSgp_1>,\ldots, <TSg_N, TSgp_N>\}$

$BS = \{< BSg1, BSgp1>,\ldots, < BSgN, BSgpN>\}$

Where, for $1 \le i \le N$,$TSg_i = \max\{C_1g[i],\ldots, C_ng[i]\}$, $BSg_i = \min\{ C_1g[i],\ldots, C_ng[i]\}$, $C_jg[i]$ is the i^{th} Cg value of j^{th} grid representation in *SC*, and TSgp and BSgp are the counterparts of TSg and BSg in *C* respectively.

Figure 9 shows the grid bounding region for 3 grid representations. The same as SBR, GBR is only one region. Therefore, GBR is free of internal overlap [21], which is efficient for index.

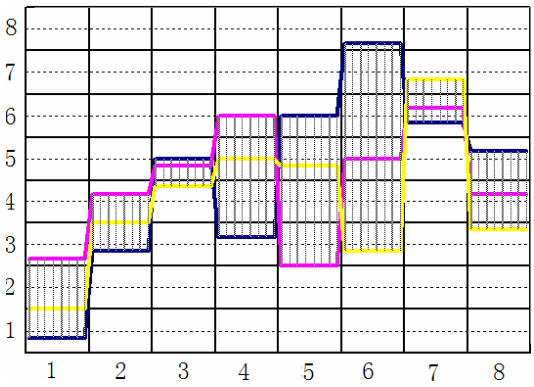

Figure 9: Grid Bounding Region of grid representations.

5.2 Distance Function for Grid Bounding Regions

Given a query's grid representation Q and a time series' C, according to function 6 we can know the distance between two corresponding grids of Q and C is as follows,

$$\mathrm{RLB\_grid}(Q_i, C_i) = \begin{cases} \frac{n}{N}(Qg_j - Cg_j)^2 Hg^2 & if\ (Qg_j - Cg_j)(Qgp_i - Cgp_i) > 0 \\ \frac{n}{N}(Qg_j - Cg_j - 1)^2 Hg^2 & if\ Qg_j - Cg_j \neq 0,\ and \\ & (Qg_j - Cg_j)(Qgp_i - Cgp_i) \leq 0 \\ 0 & otherwise \end{cases} \tag{9}$$

Then we can define a new distance function (LB_{GBR}) to lower bound the distance from Q to a GBR R for using the GBR representation in a multidimensional index as following:

$$\mathrm{LB_{GBR}}(Q, R) = \sum_{i=1}^{N} \begin{cases} \mathrm{RLB\_grid}(Q_i, TS_i) & if\ (Qg_i > TSg_i) \\ \mathrm{RLB\_grid}(Q_i, BS_i) & if\ (Qg_i < BSg_i) \\ 0 & otherwise \end{cases} \tag{10}$$

where TS_i is TS in the i^{th} grid and the same with BS_i, N is half length of each grid representation.

Through [23] we can see the function 10 satisfies the group lower bound property if RLB_grid() is replaced by a square Euclidean distance in it. And according to Proposition 2, RLB_grid() lower bounds square Euclidean distance, thus it is obvious that LB_{GBR} (Q, R) satisfied. We don't do more redundant proof here.

5.3 Indexing Time Series

So far, the most important issues for indexing have been solved, then just following the GEMINI [22] paradigm, the approximate representations of time series can be indexed by a spatial access method. We build the index based on the R-tree structure [24]. In the index, each entry in an internal node consists of the approximate representation of a GBR and a pointer to a child node. On the other hand, an entry in a leaf node consists of the approximate representation of and a pointer to a time series data object. Following the R-tree algorithm proposed by Guttman [24], we can easily construct the index.

6. EXPERIMENTAL EVALUATION

We will provide an empirical comparison between our proposed grid representation approach and the art techniques such as PAA to demonstrate it improved performance. Since our proposed method is specialized in efficient computation, we measure the elapsed time of implementing k-NN similarity search.

We used two data sets for the experiment, which are collected from various sources of real world applications and synthetic data. One data set is FOETUS_ECG data set from PhysioNet [26] and Pierre JALLON's website [27], and the other one is a mixed data set generated from various sources like ocean, ERP, and financial applications. For each of them we created three cases with the length of time series is 1024, 512, 256, and each contains 10,000 time series of the same length. And we reduce the dimensionality to 64, 32, and 16 respectively.

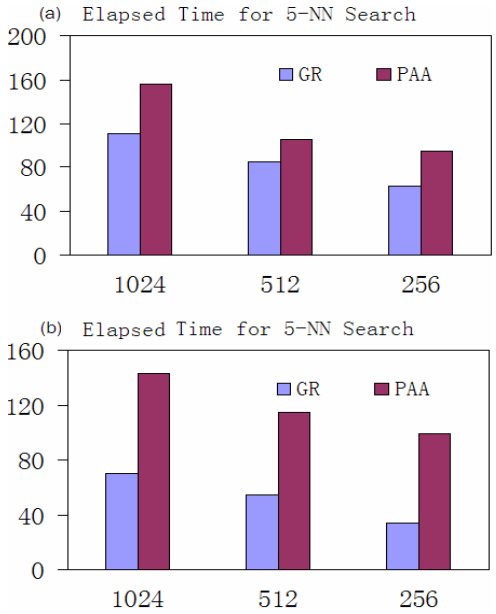

Figure 10: comparison of GR and PAA in terms of elapsed time (seconds) for 5-NN search, (a) is for FOETAL_ECG data set and (b) is for the mixed data set.

Figure 10 compares the Grid Representation (GR) and PAA techniques based on the elapsed time for 5-NN search. The Grid Representation significantly outperforms the PAA technique. It is more efficient than PAA, which argues that grid representation is a

completely competitive technique.

7. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new dimensionality reduction technique, i.e. Grid Representation. We showed that it was more efficient by using a bitmap approach and easy for index building. In the future, we will use this technique to do more experiments on clustering or other application field. We also intend to extend the work to anomaly detection, which is like what [3] described.

ACKNOWLEDGMENTS

This work was supported by MEXT.HAITEKU (2005)

REFERENCE

[1] R. Agrawal, C. Faloutsos, and A. Swami: Efficient Similarity Search in Sequence Databases, *Proc. Int'l Conf. Foundations of Data Organizations and Algorithms*, pp. 69-84, Oct, 1993.

[2] J. Aach & G. Church: Aligning gene expression time series with time warping algorithms, *Bioinformatics (17)*, 495-508, 2001

[3] Keogh, E., Lin, J. & Fu, A.: HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. *Proc. of the 5th IEEE International Conference on Data Mining*, New Orleans, LA. Nov 27-30, 2005.

[4] Nitin Kumar, Venkata Nishanth Lolla, Eamonn J. Keogh, Stefano Lonardi, Chotirat (Ann) Ratanamahatana: Time-series Bitmaps: a Practical Visualization Tool for Working with Large Time Series Databases. *SDM 2005*

[5]Keogh, E,. Chakrabarti, K,. Pazzani, M. & Mehrotra: Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information Systems*, pp 263-286, 2005

[6] Patel, P., Keogh, E., Lin, J., & Lonardi, S.: Mining Motifs in Massive Time Series Databases. *Proc. of IEEE International Conference on Data Mining.* Maebashi City, Japan. Dec 9-12, 2002.

[7] D. Berndt & J. Clifford: Using dynamic time warping to find patterns in time series. *AAAI-94 Workshop on Knowledge Discovery in Databases*. pp.229-248, 1994.

[8] Chan, K.& Fu, W.: Efficient time series matching by wavelets. *Proc. of the 15th IEEE International Conference on Data Engineering*, 1999.

[9] B. Chiu, E. Keogh, & S. Lonardi: Probabilistic Discovery of Time Series Motifs. *Proc. of the 9th ACM SIGKDD*. 2003

[10] E. Keogh & Kasetty, S.:On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. *Proc. of the 8th ACM SIGKDD*,pp. 102-111. 2002.

[11 Yi, B, K., & Faloutsos, C.. Fast time sequence indexing for arbitrary Lp norms. *Proc. of the 26st Intl Conference on VLDB*. pp 385-394. 2000

[12] Wu, Y., Agrawal, D.& Abbadi, A.: A Comparison of DFT nd DWT based Similarity Search in Time-Series Databases. *Proc. of the 9th International Conference on Information and Knowledge Management*,2000

[13] Kahveci, T. & Singh: A Variable length queries for time series data. *Proc.of the 17th International Conference on Data Engineering.* Heidelberg, Germany. 2001.

[14] Korn, F., Jagadish, H & Faloutsos. C.: Efficiently supporting ad hoc queries in large datasets of time sequences. *Proc. of SIGMOD '97*, Tucson, AZ, pp 289-300, 1997

[15] Kanth, K.V., Agrawal, D., & Singh: A. Dimensionality reduction for similarity searching in dynamic databases. *Proc.of the ACM SIGMOD Conf.*, pp. 166-176. 1998.

[16] Lin, J., Keogh, E., Patel, P. & Lonardi, S.: Finding Motifs in Time Series. *In proceedings of the 2nd Workshop on Temporal Data Mining, Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. July 23-26, 2002.

[17] Ge, X. & Smyth, P.: Deformable Markov model templates for time-series pattern matching. *In proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Boston, MA, Aug 20-23. pp 81-90, 2000.

[18] Perng, C., Wang, H., Zhang, S., & Parker, S.: Landmarks: a new model for similarity-based pattern querying in time series databases. *Proc. of the 16th International Conference on Data Engineering*., 2000

[19] Chakrabarti, K & Mehrotra, S.: The Hybrid Tree: An index structure for high dimensional feature spaces. *Proc. of the 15th IEEE International Conference on Data Engineering.*, 1999

[20] Hellerstein, J. M., Papadimitriou, C. H., & Koutsoupias, E.: Towards an analysis of indexing schemes. *Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1997

[21] Quanzhong Li, Lopez Lopez, Bongki Moon: Skyline Index or Time Series Data. *IEEE Trancactions on knowledge and data engineering*, Vol. 16, No. 6, pp.669-684, 2004

[22] C. Faloutsos, Searching Multimedia Databases Content. *Boston: Kluwer Academic*, 1996.

[23] E. Keogh: Exact Indexing of Dynamic Time Warping, Proc. *28th Very Large Databases (VLDB) Conf.*, pp. 406-417, Aug., 2002

[24] Guttman, A.: R-trees: A dynamic index structure for spatial searching. *Proc. ACM SIGMOD Conference.* pp 47-57., 1984

[25] B K. Yi & C. Faloutsos: Fast time sequence indexing for arbitrary Lp norms. *VLDB*. pp. 385-394, (2000)

[26] PhysioNet, http://www.physionet.org/

[27] Pierre JALLON's website, ttp://www-syscom.univ-mlv.fr/~jallon

[28] Chotirat Ann Ratanamahatana, Eamonn Keogh, Anthony J. Bagnall, and Stefano Lonardi. A Novel Bit Level Time Series Representation with Implications of Similarity Search and Clustering. *In proceedings of the Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, Hanoi, Vietnam, May 18-20, 2005

[29] Jiyuan Ana, Yi-Ping Phoebe Chena,b, Hanxiong Chen: DDR: an index method for large time-series datasets, Information Systems 30 (2005) 333–348, 2005

[30] J. An, H. Chen, K. Furuse, N. Ohbo, E. Keogh,: Grid-based indexing for large time series databases, in: Proceedings of Fourth International Conference on Intelligent Data Engineering and Automated Learning, , pp. 614–621. 2003