

Multi-dimensional Time Series Approximation Using Local Features at Thinned-out Keypoints

Yu Fang, Do Xuan Huy, Hung-Hsuan Huang, Kyoji Kawagoe*
Ritsumeikan University, Kusatsu, Shiga, Japan.

* Corresponding author. email: kawagoe@is.ritsumei.ac.jp
Manuscript submitted August 1, 2014; accepted September 25, 2014.
doi: 10.17706/jcp.10.1.1-11

Abstract: A multi-dimensional time-series is a sequence of vectors measured by many devices at points in time. Although many methods have been proposed to model and classify the data, these methods lead to a problematic relationship between cost and accuracy. In this paper, we propose a novel method for approximating multi-dimensional time-series, named multi-dimensional time-series Approximation with use of Local features at Thinned-out Keypoints (A-LTK), which enables an adequate accuracy value to be obtained even when reduced storage cost is a requirement. The main concepts of A-LTK are 1) reduction of time points and 2) construction of local features at the thinned-out keypoints. A preliminary evaluation indicated that with these points our proposed method was capable of achieving almost the same accuracy with less storage cost, compared to existing methods.

Key words: Multi-dimensions, times series, classification, approximation, keypoint extraction, local features.

1. Introduction

Rapid progress in the development of sensor devices has resulted in sensors with the ability to continuously acquire multiple time series data. For example, motion capture systems can output multiple streams of observed stream data on each marker position [1]. The latest smartphones and tablets can measure geographical locations which means that data from multiple sensors equipped with mobile devices such as these can be used for activity estimations. Sensors are also useful for predicting extreme weather, for example, some parts of the world are repeatedly struck and damaged by heavy storms such as hurricanes, typhoons, or cyclones. Because predicting the track of a storm is important for preventing damage, data of the characteristics for storms is measured on a continuous basis [2]. The data, which is generated as multi-dimensional time-series, include the location, speed, direction, max-wind-velocity, low-atmospheric-pressure, force-win-radius of the storm.

A multi-dimensional time-series is defined as a sequence of vectors and a dimensionality extension of one dimensional time-series, which is represented as a sequence of values. Many methods that uses multi-dimensional time-series for classifications, searching, and clustering, have been proposed to date. The crucial issues on the multi-dimensional time-series are 1) distance definition between two time series, and 2) approximation model development to achieve efficient processing with the retention of accuracy. There are currently two kinds of multi-dimensional time series approximation: point-based methods and contour-based methods. Methods in the first category include multi-dimensional regression [3] and DWT/DFT/LCS for multi-dimensional data [4]. The only existing method for the second category is

Symbolic Approximation (SAX) [5] that is used to approximate each time series composing multi-dimensional data [6].

In this paper, we propose a novel method for approximating a multi-dimensional time series as one of the second category methods, the contour-based method. The new method, the Approximation with use of Local features at Thinned-out Keypoints (A-LTK), is proposed to introduce two basic ideas which are 1) reduction of time points and 2) construction of local features at the thinned-out keypoints. Owing to the use of these ideas, a multi-dimensional time-series can be approximated with smaller storage costs.

2. Multi-dimensional Time Series

A multi-dimensional time-series is defined as a sequence of vectors, that is, $TS = \langle v_1, v_2, \dots, v_n \rangle$, where v_k is a d-dimensional vector at a time t_k . When $k=1$, TS is a time series of scalar values, that is one dimensional time series. For our purpose, the assumption is made that the length of TS, n , is a variable. A query Q is also defined as a sequence of d-dimensional vectors q_k . $Q = \langle q_1, q_2, \dots, q_m \rangle$.

Given a multi-dimensional time-series TS and a query Q , the similarity between TS and Q is defined as $\text{sim}(TS, Q)$. There are several ways to realize the function $\text{sim}(TS, Q)$. If the lengths of the two time series are the same, i.e. when $n=m$, the Euclidean can be used:

$$\text{sim}(TS, Q) = 1 / \left(\sqrt{\sum_{i=1}^n (d(v_i, q_i))^2} \right),$$

where v_i and q_i are the i-th vectors of TS and Q at a time t_i , respectively.

$d(v_i, q_i)$ is a distance function which returns the distance between v_i and q_i . Distance measures like the Euclidean or the cosine distance can be used as the distance function d .

Another way to determine the similarity is to use Dynamic Time Warping (DTW) [7]:

$$\begin{aligned} DTW(TS, Q) &= 0, \text{ if } n = 0 \text{ and } m = 0 \\ DTW(TS, Q) &= \infty, \text{ if } n = 0 \text{ or } m = 0 \\ DTW(TS, Q) &= d(\text{Head}(TS), \text{Head}(Q)) + \\ &\min(DTW(TS, \text{Rest}(Q)), DTW(\text{Rest}(TS), Q)), \\ &DTW(\text{Rest}(TS), \text{Rest}(Q)) \text{ otherwise} \end{aligned}$$

In the above definition, $\text{Head}(A)$ is the first vector of A and $\text{Rest}(A)$ is a sequence that remains after $\text{HEAD}(A)$ is removed from A .

Moreover, Longest Common Subsequence (LCS) can be used to calculate the similarity between multiple time series [8]:

$$\begin{aligned} LCS(TS, Q) &= 0, \text{ if } n = 0 \text{ or } m = 0 \\ LCS(TS, Q) &= 1 + LCS(\text{Rest}(TS), \text{Rest}(Q)) \\ &\text{if } d(\text{Head}(TS), \text{Head}(Q)) \leq \varepsilon \text{ and } |n - m| < \delta \\ LCS(TS, Q) &= \max(LCS(\text{Rest}(TS), Q), \\ &LCS(TS, \text{Rest}(Q)) \text{ otherwise} \end{aligned}$$

In the definitions of both DTW and LCS, the distance between $\text{HEAD}(A)$ and (Q) needs to be calculated for which either the cosine distance or Euclidean can be used.

Finally, the Edit distance can be applied to the distance between two multi-dimensional time series. The method is called Edit Distance on Real sequence (EDR) [9] and is defined as follows:

$$EDR(TS, Q) = n \text{ if } m = 0$$

$$EDR(TS, Q) = m \quad \text{if } n = 0$$

$$EDR(TS, Q) = \min(EDR(Res\tau(TS), Res\tau(Q)) + C, \\ EDR(Res\tau(TS), Q) + 1, EDR(TS, Res\tau(Q)) + 1) \\ \text{otherwise}$$

Some extended methods have been proposed based on these basic methods. One example of the extensions is Angular Metric for Shape Similarity (AMSS) that is defined as follows [10], [11]:

In AMSS, the difference time series is first calculated from the original data. The difference ΔA of the time series $A = \langle a_1, a_2, \dots, a_n \rangle$ is defined as

$$\Delta A = \langle \Delta A_1, \Delta A_2, \dots, \Delta A_{n-1} \rangle,$$

where $\Delta A_k = a_{k+1} - a_k$ for any $k = 1, \dots, n-1$

Then,

$$S(\Delta TS[1, i], \Delta Q[1, j]) = Sim(\Delta v_i, \Delta q_j) \quad \text{if } i = 1 \text{ and } j = 1$$

$$S(\Delta TS[1, i], \Delta Q[1, j]) = -\infty \quad \text{if } i = 1 \text{ or } j = 1$$

$$S(\Delta TS[1, i], \Delta Q[1, j]) = \\ \max(S(\Delta TS[1, i-1], \Delta Q[1, j-1]) + Sim(\Delta v_i, \Delta q_j), \\ S(\Delta TS[1, i], \Delta Q[1, j-1]) + Sim(\Delta v_i, \Delta q_j), \\ S(\Delta TS[1, i-1], \Delta Q[1, j]) + Sim(\Delta v_i, \Delta q_j))$$

$$AMSS(\Delta TS, \Delta Q) = S(\Delta TS[1, n-1], \Delta Q[1, m-1]),$$

where $Sim(\Delta v_i, \Delta q_j) = \cos \theta = (\Delta v_i \cdot \Delta q_j) / (|\Delta v_i| |\Delta q_j|)$

As shown in the above definition, AMSS determines the set of difference vector pairs producing the most similarity.

All of the above methods are processed for all the vectors of a sequence. In other words, vectors at all time-points are used to calculate time series similarities. Any of the methods, DTW, LCS, EDR and AMSS, is calculated by using a certain kind of dynamic programming scheme. Therefore, the cost of $O(n^2)$ is necessary to obtain a distance value where n is the number of time-points in the time series.

There are several methods, such as APCA [12] and SAX [5], could be used to reduce the cost in the single dimension time series, although a cost reduction method has not yet been developed for multi-dimensional time series. Therefore, in this paper, a novel method for approximating a multi-dimensional time series, called A-LTK, is proposed together with a technique that introduces thinned-out keypoints. This technique is capable of reducing the storage cost in A-LTK in addition to the computational cost. The basic principles of A-LTK are discussed in the next section.

SAX is a popular and effective method for approximating single-dimension time series. Several methods have been proposed that use SAX for approximating multi-dimensional time series. However, in this research, SAX is not considered an alternative for A-LTK, because it is a symbolic approximation method for of time series, whereas A-LTK is essentially a method approximated by reducing the number of time points. While SAX could also be implemented as a part of A-LTK, the latter method uses a local feature-based approximation instead of symbols.

A well-known method for dimensionality reduction, Principal Component Analysis (PCA) can be applied to our A-LTK in the same way as the above methods can be applied. It is because PCA is a method for reducing the dimensionalities of the multi-dimensional time series independently. In this paper, the use of

PCA is not mentioned because it can easily employed before our method is used for processing, as a pre-processing step.

3. Approximation Using Local Features at Thinned-out Keypoints

There are two important concepts with A-LTK: 1) thinned-out keypoints and 2) local feature construction.

3.1. Basic Concepts

The basic concepts are illustrated in Fig. 1. First, only those time points which are necessary for representing multi-dimensional time-series are selected and used as time-series approximation. The remaining time points are those around which local features are constructed and named keypoints. No features are constructed at the removed time points what were removed.

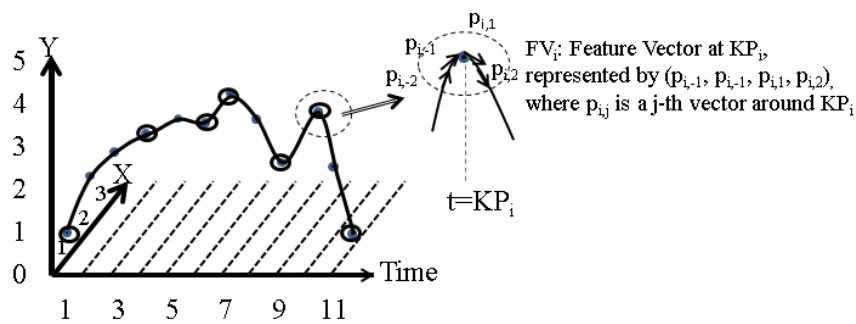


Fig. 1. Basic concepts of A-LTK.

The purpose of thinned-out keypoints is to represent an original multi-dimensional time series using vectors that are as small as possible. Extraction of a smaller number of keypoints can reduce the computational cost to calculate a dynamic programming scheme such as that used in DTW or LCS.

The purpose of local feature extraction is to retain time series characteristics from locality viewpoints. It is of course important to retain global characteristics in a time series. Using A-LTK, the global characteristics can be represented by combining the local features.

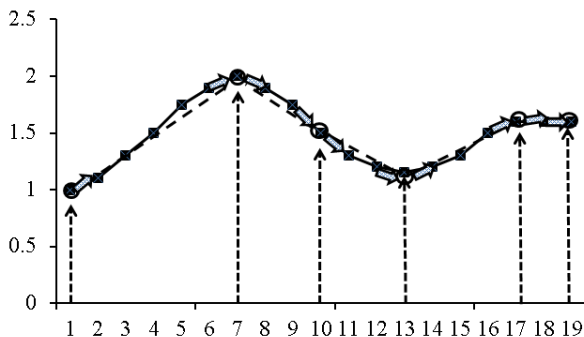


Fig. 2. An example of A-LTK.

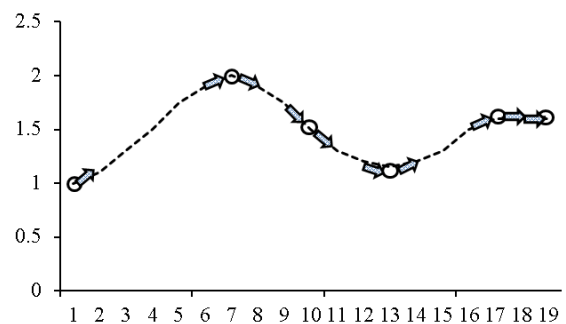


Fig. 3. A-LTK (Local feature vectors).

Fig. 2 shows a simple example of thinned-out keypoints and local features at the key-points, using the single dimension case. In this example, a time series is a sequence of values at 19 time points. When only 6 time points, $t=1, 7, 10, 13, 17,$ and $19,$ are chosen, the time series is approximated by a sequence of only five piecewise linear curves. The appropriate set of keypoints is selected so as to reduce the error and to retain the original time series characteristics. Moreover, as in this figure, the two vectors around each keypoint,

which are the prior direction and the next direction, indicate the original time series characteristics. Therefore, these characteristics can be represented with a sequence of local feature vectors at keypoints, as shown in Fig. 3.

3.2. Thinned-out Keypoints

The features of the technique of thinned-out processing are briefly described here. Two types of thinned-out conditions are introduced to select keypoints, namely, a condition on difference with the average and a condition on the second difference.

First, determining whether the condition on the difference with the average is met, given a time point t_i , is composed of the following three steps: 1) the average vector $\text{avg}(t_i)$ is calculated, 2) the difference between v_i and $\text{avg}(t_i)$ is checked, 3) if $|v_i - \text{avg}(t_i)|$ is greater than a predefined threshold ε_1 , then this time point t_i is regarded as a keypoint.

Second, determining whether the condition on the second difference is met, given a time point t_i , is composed of the following two steps: 1) the second difference $\Delta^2(t_i)$ is calculated, where $\Delta^2(t_i)$ is the difference of the first difference $\Delta^2(t_i) = \Delta(t_{i+1}) - \Delta(t_i)$ and $\Delta(t_i) = v_i - v_{i-1}$, 2) if $|\Delta^2(t_i)|$ is less than a predefined threshold ε_2 , then the time point t_i is treated as a keypoint.

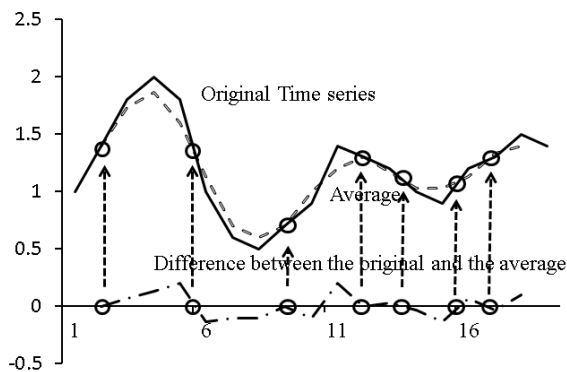


Fig. 4. A condition on the difference with the average.

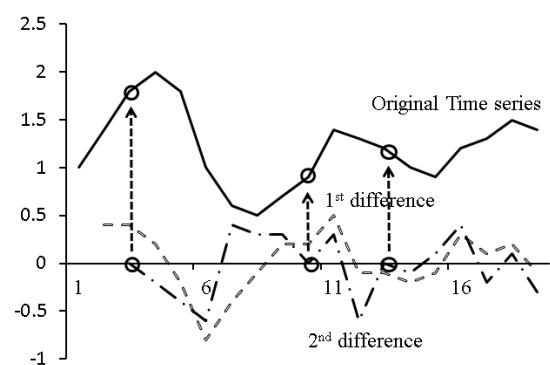


Fig. 5. A condition on the second difference.

Fig. 4 and 5 show examples of the two conditions.

As the setting that is selected for these two thresholds affects the number of keypoints, the thresholds should be determined by using the ratio of the number of keypoints to the total number of time points, that is $(\#_of_keypoints) / (\#_of_time_points)$. This ratio parameter is defined as the β -ratio and has a value in the range of 0.0 to 1.0. Given the β -ratio, the respective values of ε_1 and ε_2 are set so as to extract $(\#_of_keypoints * \beta\text{-ratio} / 2)$.

Although the above method for extracting thinned-out keypoints is introduced here for the first time, there are other method that serve the same purpose that could also be applied in our method, such as piecewise linear approximation and approximation by spline functions. They will be employed in future research.

3.3. Local Feature Constructions

Once a set of keypoints is selected, a feature vector is constructed for each keypoint. As many types of feature vectors can be defined, the description presented here is limited to those that are 1) basic, 2) difference, 3) combination of 1) and 2).

In the basic type of the feature vectors, the original value vector around the keypoint is used as the feature vector. The simple feature vector f_k at a keypoint $K_j (=t_k)$ is equal to the value vector v_k . The adjacent

value vectors can be added to the value vector v_k . In general, the p -degree (basic) feature vector is defined as $(v_{k-p-1}^T, \dots, v_{k-1}^T, v_k^T, v_{k+1}^T, \dots, v_{k+p-1}^T)^T$ with $(2p+1)*d$ as its vector degree. This type of feature vector is named pB .

The simple difference type of feature vectors is defined as follows: f_k , which is a feature vector at a keypoint $K_j (=t_k)$ is $f_k = (v_k^T - v_{k-1}^T, v_{k+1}^T - v_k^T)^T$. Therefore, in general, q -degree (difference) feature vector is defined by $(v_{k-q-1}^T - v_{k-q}^T, v_k^T - v_{k-1}^T, v_{k+1}^T - v_k^T, v_{k+q}^T - v_{k+q-1}^T)^T$ with $2q*d$ as its vector degree. This type of feature vector is named $q\Delta$.

The combination type of feature vectors is a combination of p -degree basic vector and q -degree difference vector, with a vector degree of $2p+2q-1$. This type of feature vectors is named $pB+q\Delta$.

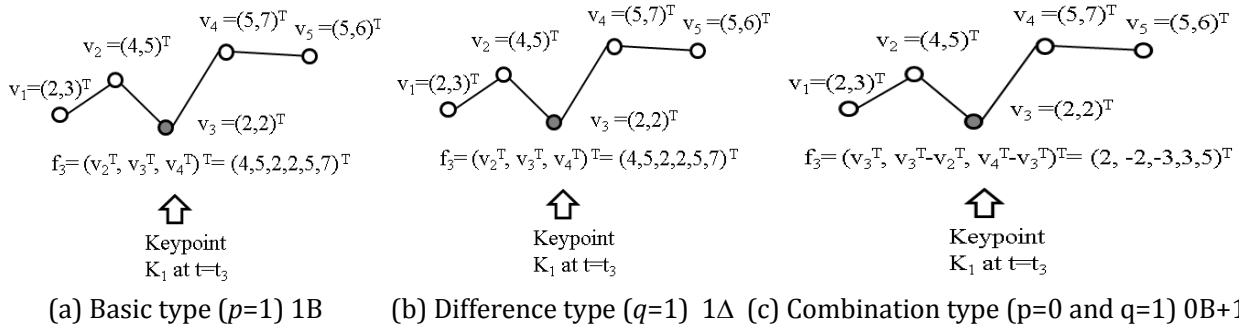


Fig. 6. A-LTK (Local feature vectors).

Fig. 6 shows some examples of these two conditions for the specific cases of $1B$, 1Δ , and $0B+1\Delta$. These three types of local features were introduced, because 1) the conventional time series are represented by a vector whose i -th element are a value at the i -th time, therefore the value can also be used as a local feature; and 2) as in some types of time series such as stock price trends or moving object trajectories, it is necessary to capture value change information, therefore its difference element should be taken into consideration as the local features. However, it should be noted that the higher the values of p or q , the larger the storage cost for local features.

Then, using the local features that were constructed, an approximated sequence, $AS(TS)$, of the original multi-dimensional time series, TS , is defined as follows: the j -th element of the sequence of $AS(TS)$, a_j , is a vector f_k at a time t_k such that t_k is the j -th keypoint. For example, suppose that two keypoints $t=t_2$ and $t=t_4$ are extracted for a $TS = \langle v_1, v_2, v_3, v_4, v_5 \rangle$. The approximated sequence of this TS , $AS(TS)$ is $\langle a_1, a_2 \rangle$, where a_1 and a_2 are local feature vectors at $t=t_2$ and $t=t_4$, respectively.

3.4. Multi-dimensional Time Series Approximation and Applications

First, a similarity function is introduced using our A-LTK. A similarity function $\text{Sim}_{A-LTK}(AS_1, AS_2)$, where $AS_1 = \langle a_1, \dots, a_N \rangle$ and $AS_2 = \langle b_1, \dots, b_M \rangle$ is defined as follows:

$$\begin{aligned} \text{SIM}_{A-LTK}(AS_1, AS_2) &= \infty, \text{ if } N = 0 \text{ and } M = 0 \\ \text{SIM}_{A-LTK}(AS_1, AS_2) &= 0, \text{ if } N = 0 \text{ or } M = 0 \\ \text{SIM}_{A-LTK}(AS_1, AS_2) &= \\ &\text{COS}(\text{Head}(AS_1), \text{Head}(AS_2)) + \\ &\max(\text{SIM}_{A-LTK}(AS_1, \text{Rest}(AS_2)), \\ &\quad \text{SIM}_{A-LTK}(\text{Rest}(AS_1), AS_2), \\ &\quad \text{SIM}_{A-LTK}(\text{Rest}(AS_1), \text{Rest}(AS_2))), \\ &\text{otherwise} \end{aligned}$$

The above definition is similar to the definition of DTW, which was described previously.

Secondly, A-LTK can be used to classify a multi-dimensional time series using the above similarity function. The simple classification task is 1-NN, where, given a time series TS, after finding the most similar time series from the database, the class of the found data is considered to be a class of the given data. When the A-LTK is applied to perform clustering of multi-dimensional time series, the above similarity definition can also be used to calculate the distance between two series. In this case any clustering method can be employed, such as K-Means, DB-Scan or a hierarchical clustering method.

Lastly, subsequent searching can be realized based on our A-LTK in the same way as for other methods. Suppose two long multi-dimensional time series are given and A-LTK is used to calculate the similarity between the two. After similar pairs of keypoints are detected, the longest interval of contiguous detected similar pairs can be determined. The details are beyond the scope of this paper. The 1-NN classification task, which is a basic and simple application of time series for comparison, will be considered in the next evaluation section.

4. Evaluations

4.1. Test Data Set

Two dataset: the Mouse-Dataset and the Mixed-2D-Dataset, were used in our experiments. The Mouse-Dataset contains the sketches of simple pictures, and is composed of four categories: circles, squares, triangles, and waves. Some examples of this dataset are shown in Fig. 7. The number of data items in each category is 20. The Mixed-2D-Dataset was constricted by collecting trajectory data from eight different application fields where each field contains four trajectories of different lengths. Some data in this dataset were extracted from UCR Time series [13]. As these experiments were conducted as preliminary experiments, we only focused on two-dimensional time series. The experiments of multi-dimensional time series with more than three dimensions are underway.

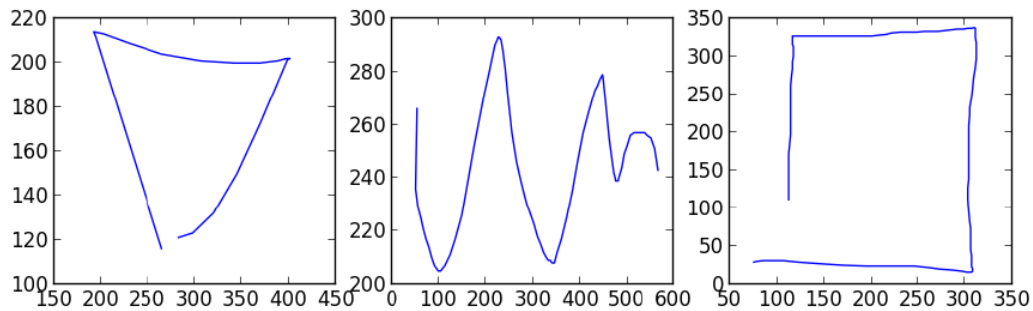


Fig. 7. Mouse-dataset samples.

4.2. Experiments

The following four experiments were conducted: thinned-out accuracy (EX1), 1-NN processing time (EX2T), 1-NN accuracy (EX2A), and clustering accuracy (EX3). The Mouse Dataset was used for EX1, EX2T, and EX2A, while the Mixed-2D Dataset was used for EX3.

EX1: This experiment checks the extent to which the expected keypoints were able to approximate the original multi-dimensional data, depending on β -ratio values. The approximation error for each time series is defined as the following:

$$(1/n) \left(\sum_i d(v_i, \text{interpolation}_i) / |v_i| \right),$$

where n is the number of time points, $d()$ is the Euclid distance function between two vectors, and

interpolation_i is the interpolated value at a time point *i* which is calculated from two vectors at the two closest keypoints. The average of the error was calculated at the end.

EX2T: This experiment determined the processing time required for 1-NN test using the Mouse-Dataset. The performance of our A-LTK was compared with those of the DTW and AMSS methods. In addition, five variations of A-LTK were compared by changing the β-ratio values.

EX2A: This experiment checks the 1-NN classification test from the standpoint of its accuracy using the Mouse-Dataset. The accuracy is defined as (Number of successful classification) / (Number of classification trials). The number of trials equaled to the number of test data set was is 80. The accuracy of the five A-LTK variations was also compared with that of the DTW and AMSS methods.

EX3: This experiment checked the clustering accuracy of the A-LTK compared with DTW and AMSS using the Mixed-2D-Dataset. The accuracy in this experiment was defined as the average of the accuracy for each obtained cluster *R_i* compared with the actual cluster *A_h* including the maximal common data as follows:

$$\sum_i (\max_h |R_i \cap A_h| / |R_i|)$$

The Ward method is used to do clustering in EX3.

4.3. Environment

Our experiment was performed in a Windows 7 (Pro) PC with the following specifications: Intel Core i5 3.0GHZ, 8GB Memory, 0.5TB Hard-disk.

4.4. Results

EX1: The results of EX1 are shown in Fig. 8 and Table 1. As shown in the Fig. 8, the shape almost retained its original shape even when a lot of data was removed. A smaller number of keypoints were enough to represent the original data. Table 1 contains the result of approximation accuracy determination with β-ratio values. From this table, it is obvious that it is possible to maintain the accuracy even when a small β-ratio (=0.036) is used.

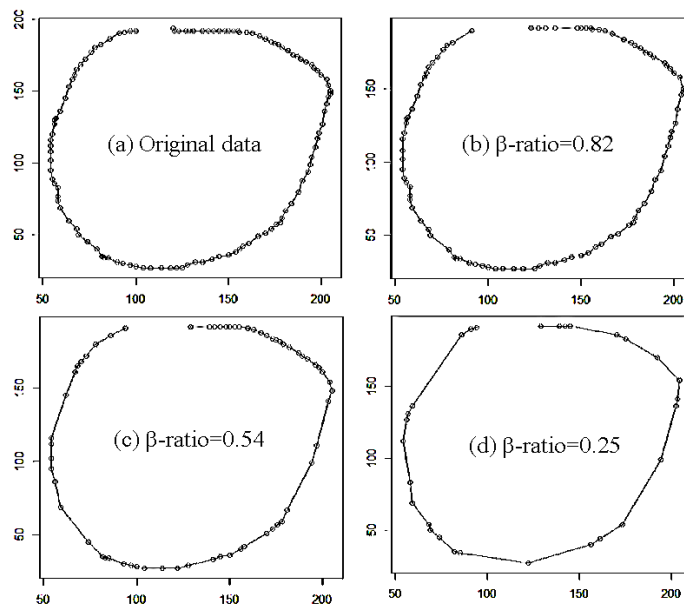


Fig. 8. Examples of thinned-out keypoint detection.

EX2T: The processing time is shown in Table 2 from which it is clear that the DTW and AMSS required much more processing time compared with A-LTK when the β-ratio values were small. The longer

processing time are ascribed to a dynamic programming computational cost, $O(n^2)$ in general. A shorter processing time is considered preferable provided the accuracy is maintained. EX2A was designed to verify this.

Table 1. Approximation Error

β -ratio(%)						
85	58	38	22	13	6.9	3.6
0.35%	6.1%	16.5%	23.3%	32.3%	123.1%	153.1%

Table 2. Computation Time (Seconds)

	β -ratio(%)						
	85	58	38	22	13	6.9	3.6
DTW	1,410						
AMSS	1,445						
1 Δ	1,132	519	223	78	29	10	<u>6</u>
2 Δ	1,123	523	226	80	30	10	<u>6</u>
1B	1,015	492	218	75	30	9	<u>6</u>
5B	1,065	519	222	77	31	10	<u>6</u>
5B+1 Δ	1,184	518	226	78	30	10	<u>6</u>

Table 3. 1NN Accuracy (%)

	β -ratio(%)						
	85	58	38	22	13	6.9	3.6
DTW	22.5						
AMSS	50.0						
1 Δ	75.0	62.5	50.0	50.0	70.0	57.5	47.5
2 Δ	76.3	68.8	50.0	70.0	70.0	57.5	45.0
1B	25.0	25.0	25.0	23.8	25.0	23.8	25.0
5B	25.0	25.0	25.0	23.8	25.0	23.8	25.0
5B+1 Δ	25.0	25.0	25.0	23.8	25.0	23.8	25.0

Table 4. Clustering Accuracy (%)

	β -ratio(%)						
	85	58	38	22	13	6.9	3.6
DTW	43.8						
AMSS	34.3						
1 Δ	34.3	34.3	34.3	34.3	34.3	34.3	34.3
1B	43.8	43.8	43.8	46.9	43.8	43.8	37.5
5B	46.9	43.8	43.8	43.8	53.1	40.6	40.6
5B+1 Δ	43.8	43.8	43.8	46.9	43.8	40.6	43.8

EX2T: The processing time is shown in Table 2 from which it is clear that the DTW and AMSS required much more processing time compared with A-LTK when the β -ratio values were small. The longer processing time are ascribed to a dynamic programming computational cost, $O(n^2)$ in general. A shorter processing time is considered preferable provided the accuracy is maintained. EX2A was designed to verify this.

EX2A: Table 3 shows the result of the 1-NN classification accuracy determination. These results show that the accuracy of the A-LTK method is similar or superior to that obtained by DTW and AMSS methods even when small β -ratio values were used, such as β -ratio=0.013 or 0.069 in the case of 1 Δ and 2 Δ . The highest accuracy was obtained in the case of A-LTK(2 Δ) with β -ratio=0.85.

EX3: In Table 4, the clustering accuracy results are shown. The results show that the A-LTK (5B) method with a β -ratio=0.13 obtained the best results in the clustering accuracy experiment. The second best results were obtained from A-LTK (1B) and A-LTK (5B+1 Δ) with a β -ratio of 0.22. This means that the accuracy displayed by A-LTK(1 Δ) was similar to that of AMSS, although worse than that of DTW.

From these experiments, it can be concluded that 1) thinned-out keypoints lead to an improved approximation and thus a smaller processing cost requirement and 2) local features at keypoints improve the accuracy of classification and clustering.

5. Conclusion and Future Work

In this paper, we propose a novel method of multi-dimensional time series approximation using local features at thinned-out keypoints. The preliminary experiments, supports to conclude that our method was

capable of attaining similar or superior accuracy compared with other existing methods with the additional advantage of a smaller processing cost.

Our future work include more detailed experiments using time series with more than three dimensions, as well as a dataset from a real application.

Acknowledgment

This work was partially supported by JSPS Grant Number 24300039 and by MEXT-Supported Program for the Strategic Research Foundation at Private Universities 2013-2017.

References

- [1] Avci, A., Bosch, S., *et al.* (2010). Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. *Proceedings of 23rd International Conference on Architecture of Computing Systems (ARCS)* (pp. 1-10).
- [2] Ho, S.-S., Tang W., & Liu, W. T. (2010). Tropical cyclone event sequence similarity search via dimensionality reduction metric learning. *Proceedings of ACM KDD2010* (pp. 135-144).
- [3] Chen, Y., *et al.* (2002). Multi-dimensional regression analysis of time-series data streams. *Proceedings of the 28th International Conference on Very Large Data Bases*.
- [4] Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., & Keogh, E. (2003). Indexing multi-dimensional time-series with support for multiple distance measures. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 216-225).
- [5] Keogh, E., Chakrabarti, K., Pazzani, M., & Mehrotra, S. (2000). Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information Systems*, 3, 263–286.
- [6] Vahdatpour, A., Amini, N., & Sarrafzadeh, M. (2009). Toward unsupervised activity discovery using multi-dimensional motif detection in time series. *Proceedings of IJCAI 2009* (pp. 1261-1266).
- [7] Holt, G. T., Reinders, M., & Hendriks, E. (2007). Multi-dimensional dynamic time warping for gesture recognition, *Proceedings of Annual Conference on the Advanced School for Computing and Imaging*.
- [8] Vlachos, M., Kollios, G., & Gunopulos, D. (2002). Discovering similar multidimensional trajectories. *Proceedings of International Conference on Data Engineering* (pp. 673-684).
- [9] Chen, L., & Ng, R. (2004). On the marriage of Lp-norms and edit distance. *Proceedings of the Thirtieth International Conference on Very Large Data Bases* (pp. 792-803).
- [10] Nakamura, T., Taki, K., Nomiya, H., & Uehara, K. (2008). AMSS: A similarity measure for time series data. *IEICE Transactions on Information and Systems*, J91-D(11), 2579-2588.
- [11] Tanaka, Y., Iwamoto, K., & Uehara, K. (2005). Discovery of time-series motif from multi-dimensional data based on MDL principle. *Machine Learning*, 58(2-3), 269-300,
- [12] Chakrabarti, K., Keogh, E., Mehrotra, S. & Pazzani, M. (2002). Locally adaptive dimensionality reduction for indexing large time series databases, *ACM Trans. Database Syst.*, 27(2), 188-228.
- [13] Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei L., & Ratanamahatana, C. A. (2011). From The UCR Time Series Classification/ Clustering, Homepage: http://www.cs.ucr.edu/~eamonn/time_series_data/



Yu Fang received the B.S. degree from Dalian Jiaotong University in 2013. Her research interests are in the area of databases and data mining.



Hung-Hsuan Huang received his B.Sc. degree in computer science from National Chen-Chi University, Taiwan in 1998 and the M.Sc. degree from National Taiwan University, Taiwan in 2000. He received his PhD degree from the Kyoto University in 2009. Currently he is an associate professor at the Ritsumeikan University, Japan. His research interests include embodied conversational agent and multimodal HCI. He is one of the members of JSAI, IPSJ, HIS, ACM, and IEICE.



Kyoji Kawagoe received the B.Eng. and M.Eng. degrees in electronic engineering from Osaka University in 1975 and 1977, respectively. He also received a Ph.D degree from Tsukuba University in 1992. He joined Ritsumeikan University in 1997, while he had worked for NEC Corporation since 1977. He is currently a full professor of Collage of Information Science and Engineering, Ritsumeikan University. His research interests include multimedia databases, ubiquitous computing and network related software research and development. He is a member of IEEE, ACM, ACM SIGMOD, Database Society of Japan, IEICE, and IPSJ.