Adaptive Global Time Sequence Averaging Method Using Dynamic Time Warping

Yu-Tao Liu, Yong-An Zhang, and Ming Zeng

***Abstract*—Time sequence averaging under dynamic time warping (DTW) is a non-trivial problem, and its exact solution requires unrealistic computational complexity in practice. The DTW barycenter averaging (DBA) method is one of the most effective iterative approximation solutions to date. However, there are still a few drawbacks in the DBA method. First, the length of the resulting average sequence depends on the selected initial average sequence; second, the discrepancy distance between the resulting average sequence and target sequence set is highly sensitive to the initialization, as we have demonstrated through the experiments described here. In this study, we propose an adaptive DBA (ADBA) algorithm to address these drawbacks. We define a scaling coefficient based on the DTW alignments such that the temporal aberrations between the average sequence and target sequence set can be qualitatively captured. The algorithm is realized by an iterative process. For each iteration, the temporary average sequence and target sequence set are partitioned into several aligned subsequence sets according to the variation in the signs of the scaling coefficients. Then, these partitioned average subsequences are adaptively compressed or stretched such that the average discrepancy distance and overall temporal aberration can be locally minimized. The comparison experiments carried out on the standard datasets illustrate that the proposed algorithm achieves lower average discrepancy distance, overall temporal aberration, and higher robustness than the available methods. Additionally, the proposed algorithm is verified by an accelerometer-based hand gesture recognitionsystem, where ADBA produces more effective gesture templates.**

***Index Terms*—Time sequence analysis, dynamic time warping, time sequence averaging, global averaging, DTW barycenter averaging, gesture recognition.**

I. Introduction

TIME sequence averaging is one of the prevalent subroutines for certain time sequence analysis tasks such as data mining [1], [2], pattern recognition [3]–[5], classification [6], [7] and clustering [8], [9]. An average sequence obtained from a class of time sequences provides a global description of the cluster it forms. For a pattern recognition system, the average sequence serving as a template has been successfully applied to improve recognition rate [10]–[12]. According to [6], [13]–[15], the time sequence averaging technique can substantially improve the performance of time sequence classification and query. Certain clustering algorithms, such as K-means clustering, require an averaging method to determine the cluster centers [13]. Another application is mentioned in [16]–[18], wherein the averaging operation is conducted for signal noise suppression. Therefore, there is an urgent demand for developing an accurate and robust time sequence averaging method.

Time sequence averaging is significantly more challenging than classical data analysis because the difference in sequence length and temporal aberrations between sequences should be considered [19]. The conventional point-by-point averaging technique is likely to result in inaccuracies and loss of information in the average sequence [17]. To address the temporal aberrations, the DTW technique is introduced as the similarity distance metric between time sequences. DTW was first developed for speech recognition [20], [21] and has been widely used in various applications such as sign language recognition [22], gestures recognition [23], [24], data mining, and time sequence clustering [25].

Time sequence averaging under DTW is significantly more challenging because the DTW distance is not a standard metric. The exact solution of time sequence averaging under DTW requires unrealistic computational complexity in practice [26]. Therefore, various iteration approximation algorithms have been developed to obtain a sub-optimal solution with acceptable computational complexity. A straightforward method to compute the average sequence is to first align the time sequences through DTW and then compute the average of these aligned elements [27].

The iteration pairwise averaging method is one of the most common methods. The differences between available iteration pairwise averaging methods are in terms of the strategy of average order and of the method by which they compute an average from two sequences [19], [27]. The nonlinear alignment and averaging filters (NLAAF) utilized a tournament scheme, where the time sequences were randomly paired and averaged [1]. Niennattrakul and Ratanamahatana [28], [29] proposed prioritized shape averaging (PSA) to prevent the bias caused by a randomly selected averaging order. In their method, the ascendant hierarchical clustering technique was employed to arrange the averaging order, and the most similar sequences were first aligned and averaged. These two methods used the symmetric average strategy, resulting in a longer average sequence in each iteration. This will negatively influence further processing because the complexity of DTW is directly related to the length of the sequence. Although a few re-sampling and uniform scaling methods can be applied to reduce the length of the resulting

Manuscript received July 23, 2018; revised November 17, 2018; accepted January 16, 2019. Date of publication February 6, 2019; date of current version March 15, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sotirios Chatzis. *(Corresponding author: Yong-An Zhang.)*

The authors are with the School of Astronautics, Harbin Institute of Technology, Harbin 150001, China (e-mail: liuyutao190160@126.com; zhangyongan76@163.com; zengming@hit.edu.cn).

Digital Object Identifier 10.1109/TSP.2019.2897958

average sequence, these operations are likely to cause information loss and severe performance degradation [19]. In [30], Sun et al. presented a degree-pruning dynamic programming algorithm to obtain an accurate average sequence of two sequences with a computational complexity of $O(n^3)$. The time sequence set was first divided into several subsets. Then, the average sequence of each subset was calculated by DBA, and the eventual average sequence was achieved by hierarchically merging these subset average sequences. It is noteworthy that these iterative pairwise averaging methods are extremely sensitive to the average order because the different average orders are likely to dramatically alter the average result.

To solve the ordering effect of the iterative pairwise averaging method, a global averaging approach that can average a set of sequences together is desirable. Petitjean *et al.* [19] presented a novel global averaging method named DTW barycenter averaging (DBA). Starting with a randomly selected initial average sequence, the temporary average sequence is iteratively refined to minimize the discrepancy distance. The evaluation in [19] demonstrates that DBA outperforms both NLAAF and PSA. However, the length of the resulting average sequence depends on the initial average sequence, and the performance is highly sensitive to the initialization. Although an adaptive scaling approach was further proposed to shorten the average sequence, there is no systematic method for selecting the length of the average sequence [19]. By linking the time sequence averaging to the multiple sequence alignment problem, the compact multiple alignment for sequence averaging (COMASA) method was proposed by Petitjean and Gançarski [31]; here, the genetic algorithm was used to search for the best compact multiple alignment. In order to speed up the convergence, DBA was employed for local optimization. However, this method incurs high computational cost. In [27], two subgradient-based optimization methods, majorize-minimize (MM) optimization method and stochastic subgradient (SSG) optimization method, were introduced by Schultz and Jain. MM is equivalent to DBA. Although empirical results indicate that SSG achieves higher performance than DBA, SSG is not a descent method in theory.

Taking inspiration from DBA and adaptive scaling technique, we propose an adaptive algorithm to overcome these drawbacks of DBA. The length of the temporary average is adjusted adaptively during the iteration process, and the proposed algorithm is robust to the initialization. The main contributions of this study can be summarized as followings:

1) We define a scaling coefficient based on the DTW alignments to quantify the temporal aberrations between the temporary average sequence and target sequence set.
2) A partition scheme is developed to partition the temporary average sequence and sequence set into several aligned subsequence sets according to the temporal aberrations.
3) An optimization procedure is proposed to refine these partitioned average subsequences by adaptively compressing or stretching, so that the average discrepancy distance and overall temporal aberration can be minimized.

The remainder of this paper is organized as follows. Section II discusses the problems of similarity measure between time sequences and time sequence averaging and then introduces the DBA method. In Section III, we detail the proposed ADBA algorithm. The experiments and results are described in Section IV. Finally, Section V concludes the article.

II. Preliminaries

Before elucidating the proposed ADBA algorithm, it is necessary to provide a brief overview of dynamic time warping, the time sequence averaging problem, and the DBA method.

A. Dynamic Time Warping

DTW has been established to be a highly effective similarity measure for time sequences because it is capable of determining the optimal alignments of two sequences despite of the temporal aberrations. Suppose we have two time sequences $\boldsymbol{X} = [x_1, \ldots, x_m]$ of length $m \in \mathbb{R}$ and $\boldsymbol{Y} = [y_1, \ldots, y_n]$ of length $n \in \mathbb{R}$, where $\boldsymbol{X}$ is the test sequence and $\boldsymbol{Y}$ is the reference sequence.

Definition 1: A warping path $\boldsymbol{P}$ between two sequences contains the indexes of the aligned elements, which is a sequence $\boldsymbol{P} = [p_1, \ldots, p_L]$ with $p_l = (i_l, j_l) \in [1 : m] \times [1 : n]$ for $l \in [1 : L]$ satisfying the following criteria:

- Boundary constraint: $p_1 = (1, 1), p_L = (m, n)$.
- Step size constraint: $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$ for $l \in [1 : L - 1]$.

The length of a warping path satisfies $\max(m, n) \leq L \leq m + n$. The set of all warping paths between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is denoted by $\mathbb{P}_{m,n}$.

Definition 2: The warping cost $d_{\boldsymbol{P}}(\boldsymbol{X}, \boldsymbol{Y})$ of a warping path $\boldsymbol{P}$ is defined as follows:

$$d_{\boldsymbol{P}}(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{l=1}^{L} d(x_{i_l}, y_{j_l}) \tag{1}$$

where $d(\cdot, \cdot)$ is the local distance measure. In this study, the squared Euclidean distance is utilized as the local distance measure, i.e., $d(x_i, y_j) = (x_i - y_j)^2$.

Definition 3: With the squared Euclidean distance as the local distance measure, the DTW distance $DTW(\boldsymbol{X}, \boldsymbol{Y})$ between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defined by

$$\begin{aligned} DTW(\boldsymbol{X}, \boldsymbol{Y}) &= \sqrt{d_{\boldsymbol{P}^*}(\boldsymbol{X}, \boldsymbol{Y})} \\ &= \min\{\sqrt{d_{\boldsymbol{P}}(\boldsymbol{X}, \boldsymbol{Y})} | \boldsymbol{P} \in \mathbb{P}_{m,n}\} \end{aligned} \tag{2}$$

where the warping path $\boldsymbol{P}^*$ incurring the minimum warping cost is called the optimal warping path.

In order to determine the optimal warping path, a $m \times n$ accumulated cost matrix $\boldsymbol{D}$ is constructed, which is calculated by the following formula:

$$D_{i,j} = d(x_i, y_j) + \min \begin{cases} D_{i-1,j-1} \\ D_{i-1,j} \\ D_{i,j-1} \end{cases} \tag{3}$$

The optimal warping path can be searched by back-tracking the accumulated cost matrix. DTW has a computational complexity of $O(m \cdot n)$ owing to the recursive operation. The DTW distance $DTW(\boldsymbol{X}, \boldsymbol{Y})$ between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is equal to $\sqrt{D_{m,n}}$.

The smaller the DTW distance, the more similar these two sequences are.

B. Time Sequence Averaging Problem

Let $\mathbb{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_N\}$ be a set of N time sequences to be averaged. In general, these sequences have different lengths. With the DTW distance as the similarity metric, the time sequence averaging issue can be formulated as an optimization problem.

Definition 4: The average time sequence $\bar{\boldsymbol{A}}$ of a sequence set $\mathbb{S}$ is defined as a sequence with the minimum average discrepancy distance between $\bar{\boldsymbol{A}}$ and $\mathbb{S}$:

$$\bar{\boldsymbol{A}} \triangleq \arg\min d_D(\boldsymbol{A}, \mathbb{S}), \forall \boldsymbol{A} \in \mathbb{X}^{\ell}, \forall \ell \in [1, +\infty) \tag{4}$$

where $\mathbb{X}^{\ell}$ is the sequence space of all time sequences of length ℓ, and the average discrepancy distance $d_D(\boldsymbol{A}, \mathbb{S})$ is the mean of the squared DTW distances between $\boldsymbol{A}$ and $\boldsymbol{S}_j \in \mathbb{S}, j = 1, 2, \ldots, N$:

$$d_D(\boldsymbol{A}, \mathbb{S}) = \frac{1}{N} \sum_{j=1}^{N} DTW^2(\boldsymbol{A}, \boldsymbol{S}_j)$$

As explained previously, a polynomial-time algorithm for determining an optimal solution of Eq. (4) incurs an unrealistic computational cost, which is intractable for more than a few sequences. In the past few decades, numerous researchers have attempt to develop an effective approximate averaging method. Among these available approximate averaging methods, DBA is the most popular solution. Next, we provide a short overview of this method.

C. DTW Barycenter Averaging (DBA)

The DBA algorithm is an iterative method that repeatedly updates the temporary average sequence to minimize the discrepancy distance. With a randomly selected initial average sequence, the element of temporary average sequence is updated by averaging all the aligned elements together in each iteration. The detailed process of DBA is described in Algorithm 1, where $\bar{\boldsymbol{A}} = \mathrm{DBA}(\mathbb{S}, \boldsymbol{S}_n)$ is the average sequence of $\mathbb{S}$ with the initial average sequence $\boldsymbol{S}_n$.

In [15], Petitjean et al. have established that DBA is a descent algorithm. In order to obtain an effective result, the iteration processing repeats I times or terminates when the discrepancy distance converges to a certain value, i.e., $d_D(\bar{\boldsymbol{A}}_i, \mathbb{S}) - d_D(\bar{\boldsymbol{A}}_{i+1}, \mathbb{S}) < \epsilon$. From Algorithm 1, we observe that the length of the average sequence depends on the selected initial average sequence. This problem becomes significantly more pronounced for the sequence set with different lengths. Selecting an initial average sequence with appropriate length is still not a straightforward problem. Moreover, DBA can't reduce the temporal aberrations between the resulted average sequence and target sequence set. The overall performance is highly sensitive to the initialization, as will be demonstrated in the experiments.

Algorithm 1: $\mathrm{DBA}(\mathbb{S}, \bar{\boldsymbol{A}}_0)$.

```
Input: The target sequence set $\mathbb{S} = \{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_N\}$, the
    initial average sequence $\bar{\boldsymbol{A}}_0 = \boldsymbol{S}_n, n \in [1:N]$.
Output: Average sequence: $\bar{\boldsymbol{A}}$
for $i = 0 : I$ do
    Step 1: Align $\boldsymbol{A}_i$ and $\boldsymbol{S}_j, j = 1, 2, \ldots, N$ through DTW.
    Step 2: Identify the elements in $\boldsymbol{S}_j, j = 1, 2, \ldots, N$
        that align with $\bar{a}_i^k \in \bar{\boldsymbol{A}}_i, k = 1, 2, \ldots, l$
            $\bar{a}_i^k \leftarrow \boldsymbol{C}_k = [c_1, \ldots, c_{\gamma_k}]$
    Step 3: Update each element of $\bar{\boldsymbol{A}}_i$ by computing the
        average of all the aligned elements associated with it
            $\bar{a}_{i+1}^k = \frac{1}{\gamma_k} \sum_{j=1}^{\gamma_k} c_j$
    if $d_D(\bar{\boldsymbol{A}}_i, \mathbb{S}) - d_D(\bar{\boldsymbol{A}}_{i+1}, \mathbb{S}) < \epsilon$ then
        break
    end if
end for
return $\bar{\boldsymbol{A}} = \bar{\boldsymbol{A}}_{i+1}, d'_D(\bar{\boldsymbol{A}}, \mathbb{S})$
```

III. Adaptive Global Time Sequences Average Method

To address these drawbacks of DBA, a novel time sequence averaging algorithm, ADBA, is developed. Unlike DBA, the proposed ADBA algorithm averages the sequence set in amplitude space as well as in temporal space. For each iteration process, the length of the temporary average sequence is adjusted adaptively through local stretching and compression. We provide insight into the mechanism of the ADBA algorithm. First, an initial average sequence is randomly selected from the sequence set, i.e., $\bar{\boldsymbol{A}}_0 = \boldsymbol{S}_n, n \in [1:N]$. For each iteration, ADBA operates in the following steps:

1) Calculate the scaling coefficients according to the DTW alignments, and determine the temporal aberrations between the temporary average sequence $\bar{\boldsymbol{A}}_i$ and the sequence set $\mathbb{S}$.
2) Apply a partition scheme to partition $\bar{\boldsymbol{A}}_i$ and $\boldsymbol{S}_j \in \mathbb{S}, j = 1, 2, \ldots, N$ into several aligned subsequence sets based on the temporal aberrations.
3) Update these partitioned average subsequences through an optimization procedure.

A. Temporal Aberrations Detection

In order to detect the temporal aberrations between $\bar{\boldsymbol{A}}_i$ and $\mathbb{S}$, the alignments between $\bar{\boldsymbol{A}}_i$ and $\boldsymbol{S}_j \in \mathbb{S}, j = 1, 2, \ldots, N$ are calculated through DTW; here, $\boldsymbol{S}_j$ is considered as the reference sequence, and $\bar{\boldsymbol{A}}_i$ is regarded as the test sequence. Let $\boldsymbol{P}_{i,j}^*$ be the DTW warping path between $\bar{\boldsymbol{A}}_i$ and $\boldsymbol{S}_j$. According to the temporal aberrations, these alignments can be categorized into three classes: matching alignments (MAs), compression alignments (CAs), and stretch alignments (SAs), as shown in Fig. 1.

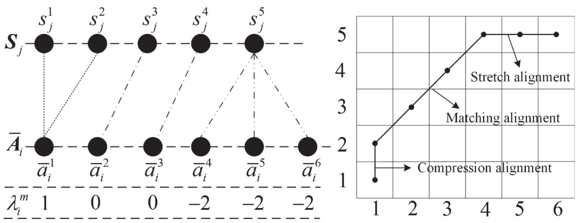


Fig. 1. Example of DTW alignments and alignment coefficients between two sequences.

The details of these three types of alignments are discussed below:

- MA: the warping path of a matching alignment traces the anti-diagonal direction. The elements from two sequences perfectly align one-to-one.
- CA: the warping path of a compression alignment is vertical. The single element in $\bar{\boldsymbol{A}}_i$ aligns with α elements in $\boldsymbol{S}_j$. $\bar{\boldsymbol{A}}_i$ is compressed with respect to $\boldsymbol{S}_j$ at that point.
- SA: the warping path of a stretch alignment is horizontal. The β elements in $\bar{\boldsymbol{A}}_i$ align with only one element in $\boldsymbol{S}_j$. $\bar{\boldsymbol{A}}_i$ is stretched with respect to $\boldsymbol{S}_j$ at that segment.

Definition 5: Based on the DTW alignment associated with $\bar{a}_i^m \in \bar{\boldsymbol{A}}_i$ with respect to $\boldsymbol{S}_j \in \mathbb{S}$, the alignment coefficient λ_j^m of $\bar{a}_i^m$ is defined by

$$\lambda_j^m = \begin{cases} 0 & \text{for the MA case} \\ \alpha - 1 & \text{for the CA case} \\ 1 - \beta & \text{for the SA case} \end{cases} \tag{5}$$

The alignment coefficient can be utilized to quantify the temporal aberrations between two sequences. To visually explain the relationship between DTW alignment and alignment coefficient, we use Fig. 1 as an example. The alignment coefficient of each element is presented in Fig. 1. The positive alignment coefficient indicates that the test sequence is compressed with respect to the reference sequence at that point. In contrast, the negative alignment coefficient indicates that the test sequence is stretched with respect to the reference sequence at that point. The amplitude represents the degree of temporal aberration. In order to provide a qualitative method for evaluating the temporal aberrations between $\bar{\boldsymbol{A}}_i$ and $\mathbb{S}$, we define a new indicator scaling coefficient.

Definition 6: The scaling coefficient ρ_m for $\bar{a}_i^m \in \bar{\boldsymbol{A}}_i$ with respect to the sequence set $\mathbb{S}$ is defined as follows:

$$\rho_m = \frac{1}{N} \sum_{j=1}^{N} \lambda_j^m \tag{6}$$

In the ideal case, each element in $\bar{\boldsymbol{A}}_i$ aligns with only one element in each sequence of $\mathbb{S}$, resulting in a total of N aligned elements. Therefore, the scaling coefficient is equal to zero. The positive scaling coefficient indicates that the aligned elements associated with $\bar{a}_i^m$ are more than N, and $\bar{\boldsymbol{A}}_i$ is compressed relative to $\mathbb{S}$ at the point $\bar{a}_i^m$. In contrast, the negative scaling coefficient indicates that the aligned elements associated with $\bar{a}_i^m$ is less than N, and $\bar{\boldsymbol{A}}_i$ is stretched with respect to $\mathbb{S}$ at the point $\bar{a}_i^m$. Fig. 2 illustrates an example of scaling coefficients ("clockwise" gesture training dataset), which is labeled with different colors based on the sign. The red curve indicates that the scaling coefficient is larger than zero. The blue curve indicates that the scaling coefficient is less than zero.

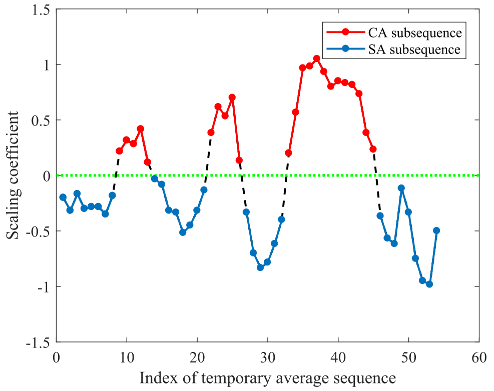


Fig. 2. An example of scaling coefficient curve.

B. Sequence Partition

As discussed above, the positive and negative scaling coefficients indicate the CA and SA, respectively. Based on the variation in the signs of the scaling coefficients, we partition the indexes of $\bar{\boldsymbol{A}}_i$ into k non-overlapping contiguous segments:

$$[1, 2, \ldots, l] \Rightarrow \{[l_0 + 1, \ldots, l_1], \ldots, [l_{k-1} + 1, \ldots, l_k]\}$$

where $l_0 = 0, l_k = l$ and $\text{sign}(\rho_m) = \text{sign}(\rho_n), \forall m, n \in [l_{x-1} + 1 : l_x], x = 1, 2, \ldots, k$.

Accordingly, $\bar{\boldsymbol{A}}_i$ can be partitioned into k subsequences:

$$\bar{\boldsymbol{A}}_i = \{\bar{\boldsymbol{A}}_i^1, \bar{\boldsymbol{A}}_i^2, \ldots, \bar{\boldsymbol{A}}_i^k\}$$

where $\bar{\boldsymbol{A}}_i^x = [\bar{a}_i^{l_{x-1}+1}, \ldots, \bar{a}_i^{l_x}], x = 1, 2, \ldots, k$. For the scaling coefficients shown in Fig. 2, $\bar{\boldsymbol{A}}_i$ is partitioned into seven subsequences.

Next, each sequence in $\mathbb{S}$ is partitioned based on the partition of $\bar{\boldsymbol{A}}_i$ and the DTW alignments between $\bar{\boldsymbol{A}}_i$ and $\boldsymbol{S}_j \in \mathbb{S}, j = 1, 2, \ldots, N$. Because each element in $\boldsymbol{S}_j$ aligns with only one element in $\bar{\boldsymbol{A}}_i$ for the MA and CA, the partition can be performed directly when the partition point belongs to one of these two alignments. However, the partition is largely complex when the partition point belongs to an SA because multiple elements in $\bar{\boldsymbol{A}}_i$ align with only one element $s_j^n \in \boldsymbol{S}_j$ for the SA. In order to maintain the original alignments after partition, an element $\tilde{s}_j^n = s_j^n$ is inserted into the middle of s_j^n and s_j^{n+1} in $\boldsymbol{S}_j$. Therefore, this SA can be partitioned into two independent alignments, as shown in Fig. 3. The elements of $\bar{\boldsymbol{A}}_i$ on the left side of the partition point align with s_j^n, and the remaining elements in $\bar{\boldsymbol{A}}_i$ align with $\tilde{s}_j^n$. In this manner, the sequence $\boldsymbol{S}_j \in \mathbb{S}$ can be partitioned into k subsequences:

$$\boldsymbol{S}_j \Rightarrow \tilde{\boldsymbol{S}}_j = \{\boldsymbol{S}_j^1, \boldsymbol{S}_j^2, \ldots, \boldsymbol{S}_j^k\}, j = 1, 2, \ldots, N$$

where the subsequence $\boldsymbol{S}_j^x \in \boldsymbol{S}_j$ aligns with the subsequence $\bar{\boldsymbol{A}}_i^x, x = 1, 2, \ldots, k$. The sequence $\tilde{\boldsymbol{S}}_j$ is the combination of k subsequences $\boldsymbol{S}_j^x, x = 1, 2, \ldots, k$. It can be regarded as the

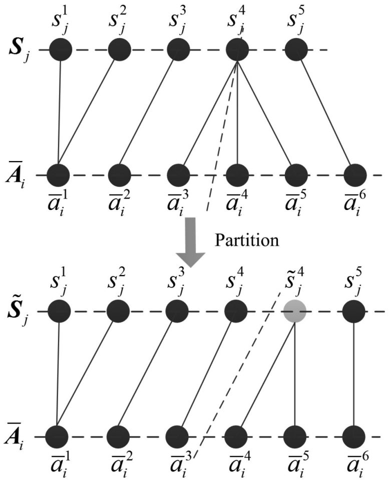


Fig. 3. Partition operation for the stretch alignment.

variation of $\boldsymbol{S}_j$ with certain special inserted elements. Specifically, it is equal to $\boldsymbol{S}_j$ if no partition between $\bar{\boldsymbol{A}}_i$ and $\boldsymbol{S}_j$ crosses an SA.

Correspondingly, the warping path between $\bar{\boldsymbol{A}}_i$ and $\boldsymbol{S}_j \in \mathbb{S}$ can be partitioned into k segments:

$$\boldsymbol{P}^*_{i,j} \Rightarrow \{\boldsymbol{P}^1_{i,j}, \boldsymbol{P}^2_{i,j}, \ldots, \boldsymbol{P}^k_{i,j}\}, j = 1, 2, \ldots, N$$

where $\boldsymbol{P}^x_{i,j}$ is the warping path between the subsequences $\bar{\boldsymbol{A}}^x_i$ and $\boldsymbol{S}^x_j, x = 1, 2, \ldots, k$.

Lemma 1: Suppose we have two time sequences $\boldsymbol{S} = [s_1, s_2, \ldots, s_m]$ and $\boldsymbol{A} = [a_1, a_2, \ldots, a_n]$. A new sequence $\tilde{\boldsymbol{S}}$ is defined as

$$\tilde{\boldsymbol{S}} = [s_1, \ldots, s_j, \tilde{s}_j, s_{j+1}, \ldots, s_m]$$

where the inserted element satisfies $\tilde{s}_j = s_j, j \in [1 : m-1]$.

Then, we have

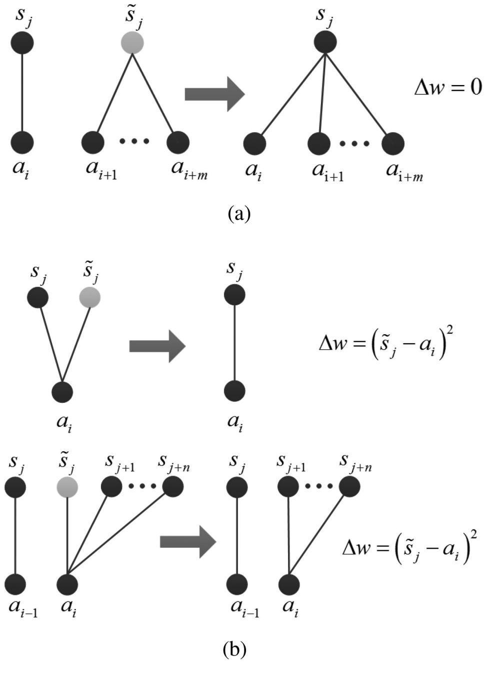
$$DTW(\boldsymbol{A}, \boldsymbol{S}) \leq DTW(\boldsymbol{A}, \tilde{\boldsymbol{S}}) \tag{7}$$

Proof: Comparing with $\boldsymbol{S}$, these two adjacent elements s_j and $\tilde{s}_j$ in $\tilde{\boldsymbol{S}}$ can be considered as two "singular points". Based on the DTW warping path $\tilde{\boldsymbol{P}}^*$ between $\boldsymbol{A}$ and $\tilde{\boldsymbol{S}}$ ($\boldsymbol{A}$ is the test sequence and $\tilde{\boldsymbol{S}}$ is the reference sequence), we can obtain a warping path $\boldsymbol{P}$ between $\boldsymbol{A}$ and $\boldsymbol{S}$ by deleting one of these two singular points and re-mapping the alignments associated with the deleted element. Let $\triangle w$ be the warping cost related to the re-mapped alignment path. The DTW distance between $\boldsymbol{A}$ and $\tilde{\boldsymbol{S}}$ can be re-expressed as

$$\begin{aligned} DTW(\boldsymbol{A}, \tilde{\boldsymbol{S}}) &= \sqrt{d_{\tilde{\boldsymbol{P}}^*}(\boldsymbol{A}, \tilde{\boldsymbol{S}})} \\ &= \sqrt{d_{\boldsymbol{P}}(\boldsymbol{A}, \boldsymbol{S}) + \triangle w} \end{aligned} \tag{8}$$

Because this deleting operation does not change the alignments associated with the other elements except for these two singular points, we only focus on the alignments associated with these two singular points. The deleting operation can produce either of two feasible results according to the alignments associated with these two singular points.

In the first case, none of the alignments associated with these two singular points is a CA. The inserted element $\tilde{s}_j$ is removed from $\tilde{\boldsymbol{S}}$, and the elements in $\boldsymbol{A}$ aligned with $\tilde{s}_j$ are re-matched to s_j, as shown in Fig. 4(a). Because $s_j = \tilde{s}_j$, the re-mapped alignments does not change the warping cost, i.e., $\triangle w = 0$.



(a)

(b)

Fig. 4. Illustration of deleting operation for a compression alignment. (a) Illustration of deleting operation in the first case. (b) Illustration of deleting operation in the second case.

In the other case, at least one alignment associated with these two singular points is a CA. The element which belongs to CA is removed. Simultaneously, the alignment associated with it is also removed, resulting in a reduction in the warping cost, i.e., $\triangle w = (\tilde{s}_j - a_i)^2$, as shown in Fig. 4(b).

Based on the analysis of these two cases, we observe that the reduction in the warping cost owing to the deleting operation is at least zero, i.e., $\triangle w \geq 0$. Suppose $\boldsymbol{P}^*$ is the DTW warping path between $\boldsymbol{A}$ and $\boldsymbol{S}$; we have $d_{\boldsymbol{P}^*}(\boldsymbol{A}, \boldsymbol{S}) \leq d_{\boldsymbol{P}}(\boldsymbol{A}, \boldsymbol{S})$. Based on Eq. (8), we can establish that

$$DTW(\boldsymbol{A}, \boldsymbol{S}) \leq DTW(\boldsymbol{A}, \tilde{\boldsymbol{S}})$$

■

Lemma 2: Given two sequences $\bar{\boldsymbol{A}}_i$ and $\boldsymbol{S}_j$, they can be partitioned into k pairs of aligned subsequences $\{\bar{\boldsymbol{A}}^x_i, \boldsymbol{S}^x_j\}, x = 1, 2, \ldots, k$ through the above partition method. Then, we have

$$DTW^2(\bar{\boldsymbol{A}}_i, \boldsymbol{S}_j) = \sum_{x=1}^{k} DTW^2(\bar{\boldsymbol{A}}^x_i, \boldsymbol{S}^x_j) \tag{9}$$

Proof: Because the partition of $\bar{\boldsymbol{A}}_i$ and $\boldsymbol{S}_j$ does not change the DTW alignments, the squared DTW distance between $\bar{\boldsymbol{A}}_i$ and $\boldsymbol{S}_j$ can be re-expressed as

$$\begin{aligned} DTW^2(\bar{\boldsymbol{A}}_i, \boldsymbol{S}_j) &= d_{\boldsymbol{P}^*_{i,j}}(\bar{\boldsymbol{A}}_i, \boldsymbol{S}_j) \\ &= \sum_{x=1}^{k} d_{\boldsymbol{P}^x_{i,j}}(\boldsymbol{A}^x_{i,j}, \boldsymbol{S}^x_j) \end{aligned} \tag{10}$$

Let $\hat{\boldsymbol{P}}_{i,j}^{x}$ be the DTW warping path between the subsequences $\bar{\boldsymbol{A}}_i^x$ and $\boldsymbol{S}_j^x$. Because the DTW warping path has the minimum warping cost, we have

$$\begin{aligned} DTW^2(\bar{\boldsymbol{A}}_i^x, \boldsymbol{S}_j^x) &= d_{\hat{\boldsymbol{P}}_{i,j}^x}(\bar{\boldsymbol{A}}_i^x, \boldsymbol{S}_j^x) \\ &\le d_{\boldsymbol{P}_{i,j}^x}(\bar{\boldsymbol{A}}_i^x, \boldsymbol{S}_j^x) \end{aligned} \tag{11}$$

Combining Eq. (10) and Eq. (11), we obtain

$$DTW^2(\bar{\boldsymbol{A}}_i, \boldsymbol{S}_j) \ge \sum_{x=1}^{k} DTW^2(\bar{\boldsymbol{A}}_i^x, \boldsymbol{S}_j^x) \tag{12}$$

As mentioned previously, $\tilde{\boldsymbol{S}}_j$ is a variation of $\boldsymbol{S}_j$ with certain special inserted elements. The combination of the DTW warping paths between these aligned subsequences $\tilde{\boldsymbol{P}}_{i,j} = \{\hat{\boldsymbol{P}}_{i,j}^1, \hat{\boldsymbol{P}}_{i,j}^2, \ldots, \hat{\boldsymbol{P}}_{i,j}^x\}$ constructs a warping path between $\bar{\boldsymbol{A}}_i$ and $\tilde{\boldsymbol{S}}_j$. By comparing with the DTW warping path $\tilde{\boldsymbol{P}}_{i,j}^*$ between $\bar{\boldsymbol{A}}_i$ and $\tilde{\boldsymbol{S}}_j$, we obtain

$$\begin{aligned} DTW^2(\bar{\boldsymbol{A}}_i, \tilde{\boldsymbol{S}}_j) &\le d_{\tilde{\boldsymbol{P}}_{i,j}}(\bar{\boldsymbol{A}}_i, \tilde{\boldsymbol{S}}_j) \\ &\le \sum_{x=1}^{k} d_{\hat{\boldsymbol{P}}_{i,j}^x}(\bar{\boldsymbol{A}}_i^x, \boldsymbol{S}_j^x) \\ &\le \sum_{x=1}^{k} DTW^2(\bar{\boldsymbol{A}}_i^x, \boldsymbol{S}_j^x) \end{aligned} \tag{13}$$

Based on Lemma 1 and Eq. (13), we have

$$\begin{aligned} DTW^2(\bar{\boldsymbol{A}}_i, \boldsymbol{S}_j) &\le DTW^2(\bar{\boldsymbol{A}}_i, \tilde{\boldsymbol{S}}_j) \\ &\le \sum_{x=1}^{k} DTW^2(\bar{\boldsymbol{A}}_i^x, \boldsymbol{S}_j^x) \end{aligned} \tag{14}$$

Combining Eq. (12) and Eq. (14), we establish the following:

$$DTW^2(\bar{\boldsymbol{A}}_i, \boldsymbol{S}_j) = \sum_{x=1}^{k} DTW^2(\bar{\boldsymbol{A}}_i^x, \boldsymbol{S}_j^x)$$

■

Let $\mathbb{S}^x = \{\boldsymbol{S}_1^x, \boldsymbol{S}_2^x, \ldots, \boldsymbol{S}_N^x\}$ be the set of subsequences aligned with $\bar{\boldsymbol{A}}_i^x$, $x = 1, 2, \ldots, k$. Thus, $\bar{\boldsymbol{A}}_i$ and $\mathbb{S}$ can be partitioned into k aligned subsequence sets:

$$\{\bar{\boldsymbol{A}}_i, \mathbb{S}\} \Rightarrow \{\{\bar{\boldsymbol{A}}_i^1, \mathbb{S}^1\}, \{\bar{\boldsymbol{A}}_i^2, \mathbb{S}^2\}, \ldots, \{\bar{\boldsymbol{A}}_i^k, \mathbb{S}^k\}\}$$

According to the sign of the scaling coefficients, these aligned subsequence sets can be categorized into three classes:

- MA subsequence set: the scaling coefficients of $\bar{\boldsymbol{A}}_i^x$ are equal to zero. The length of $\bar{\boldsymbol{A}}_i^x$ is equal to the average length of $\mathbb{S}^x$.
- CA subsequence set: the scaling coefficients of $\bar{\boldsymbol{A}}_i^x$ are larger than zero. The length of $\bar{\boldsymbol{A}}_i^x$ is shorter than the average length of $\mathbb{S}^x$, and $\bar{\boldsymbol{A}}_i^x$ is likely to be compressed with respect to $\mathbb{S}^x$.
- SA subsequence set: the scaling coefficients of $\bar{\boldsymbol{A}}_i^x$ are less than zero. The length of $\bar{\boldsymbol{A}}_i^x$ is longer than the average length of $\mathbb{S}^x$, and $\bar{\boldsymbol{A}}_i^x$ is likely to be stretched relative to $\mathbb{S}^x$.

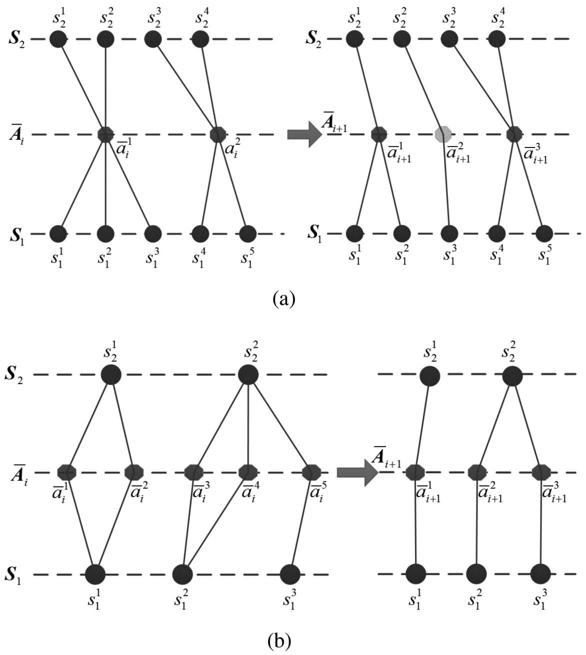


Fig. 5. Illustration of subsequence optimization. (a) Illustration of stretch optimization, one element is inserted into the average subsequence. (b) Illustration of compression optimization, three consecutive elements in the average subsequence are merged to form one element.

C. Subsequence Optimization

After the sequence partition, a novel subsequence optimization algorithm is proposed to address these aligned subsequence sets with different temporal aberrations. The objective of the optimization procedure is to refine each average subsequence such that the local average discrepancy distance and temporal aberrations can be minimized.

Taking into account these three classes of aligned subsequence sets, different optimization strategies are developed for each of the classes. For the MA subsequence set, because the length of $\bar{\boldsymbol{A}}_i^x$ is equal to the average length of $\mathbb{S}^x$, the DBA method is utilized directly to optimize this temporary average subsequence; i.e., $\bar{\boldsymbol{A}}_{i+1}^x = \mathrm{DBA}(\mathbb{S}^{x^-}, \bar{\boldsymbol{A}}_i^x)$. For the CA subsequence set, the short temporary average subsequence should be stretched, i.e., one or more elements are inserted into the temporary average subsequence. This optimization scheme is defined as stretch optimization, as shown in Fig. 5(a). For the SA subsequence set, the long temporary average subsequence should be compressed; i.e., certain elements in the temporary average subsequence are merged. This optimization operation is referred to as compression optimization, as shown in Fig. 5(b).

There are three major challenges to the stretch and compression optimization approaches:

1) How many elements are to be inserted or merged?
2) Where are these elements inserted, or which elements are merged?
3) How does one determine the inserted elements, or how does one merge these elements?

Focusing on the first problem, an iteration procedure is introduced. In each iteration, only one element is inserted into the temporary average subsequence for the stretch optimization process, and two adjacent elements in the temporary average subsequence are merged into one element for the compression optimization process. Then, DBA is used to optimize the modified temporary average subsequence. Let $+\bar{\boldsymbol{A}}_i^x(r) = \mathrm{DBA}(\mathbb{S}^x, +\boldsymbol{A}_i^x(r))$ be the refined average subsequence of $\bar{\boldsymbol{A}}_i^x$ with r inserted elements, where $+\boldsymbol{A}_i^x(r)$ is calculated by inserting an element in $+\bar{\boldsymbol{A}}_i^x(r-1)$. Let $-\bar{\boldsymbol{A}}_i^x(r) = \mathrm{DBA}(\mathbb{S}^x, -\boldsymbol{A}_i^x(r))$ be the refined average subsequence of $\bar{\boldsymbol{A}}_i^x$ with r merging iterations, where $-\boldsymbol{A}_i^x(r)$ is obtained by merging two consecutive elements in $-\bar{\boldsymbol{A}}_i^x(r-1)$.

An intuitive method to address the second problem is to insert the element at the point with the maximum compression ratio for the stretch optimization or to merge the two consecutive elements with the maximum stretching ratio for the compression optimization. As explained previously, the scaling coefficient ρ_m indicates the compression ratio or stretching ratio; it can be used to regulate these optimization processes.

For the stretch optimization, the added element $\tilde{a}_i^y$ is inserted between the two adjacent elements $\bar{a}_i^y$ and $\bar{a}_i^{y+1}$ having the maximum sum of scaling coefficients; here, $\tilde{a}_i^y$ is calculated as follows:

$$\tilde{a}_i^y = \frac{\bar{a}_i^y + \bar{a}_i^{y+1}}{2} \tag{15}$$

For the compression optimization, the two adjacent elements $\bar{a}_i^y$ and $\bar{a}_i^{y+1}$ having the minimum sum of scaling coefficients are merged to form one element $\tilde{a}_i^y$; here, $\tilde{a}_i^y$ is the mean of these two elements.

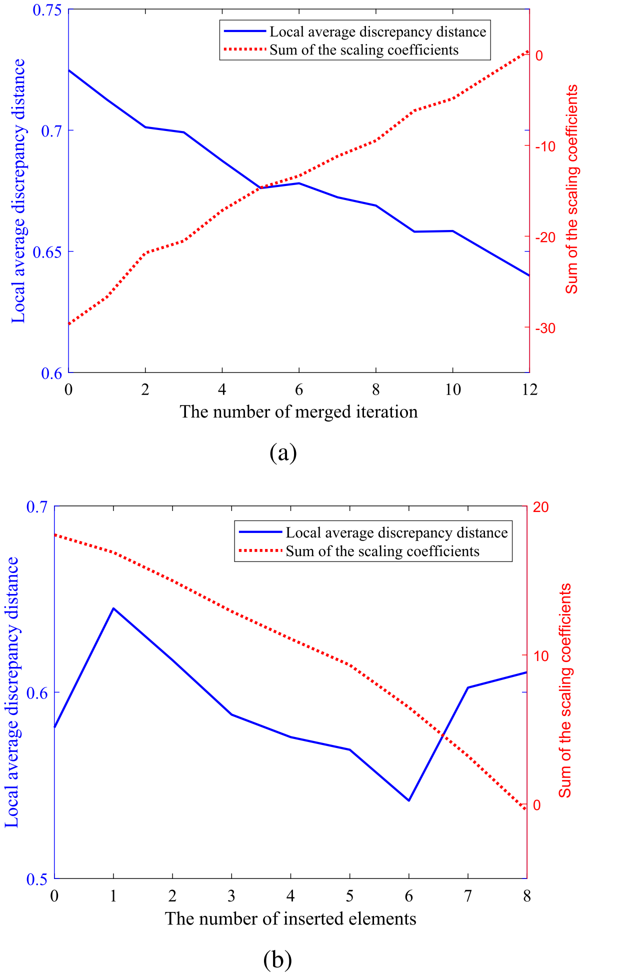
In order to prevent over-compression or over-stretching during the optimization process, the compression or stretch iteration optimization procedure is terminated when the sum of the scaling coefficients changes sign. Fig. 6 illustrates the variation in the local average discrepancy distance and sum of the scaling coefficients in the compression and stretch optimization processes, respectively. After the iteration optimization process, the refined average subsequence with the minimal local average discrepancy distance is selected as the updated average subsequence; this results in more effective alignments between $\bar{\boldsymbol{A}}_{i+1}^x$ and $\mathbb{S}^x$.

For an iterative algorithm, the convergence property is quite important, which can ensure that the updated average sequence produces a non-increasing average discrepancy distance at each iteration. Next, we will provide the proof of convergence for ADBA.

Theorem 1: ADBA is a convergent algorithm. For each iteration, it has a no-increasing average discrepancy distance, i.e., $d_D(\bar{\boldsymbol{A}}_{i+1}, \mathbb{S}) \le d_D(\bar{\boldsymbol{A}}_i, \mathbb{S})$.

Proof: Based on Lemma 2, the average discrepancy distance between $\bar{\boldsymbol{A}}_i$ and $\mathbb{S}$ can be expressed as

$$\begin{aligned} d_D(\bar{\boldsymbol{A}}_i, \mathbb{S}) &= \frac{1}{N}\sum_{j=1}^{N} DTW^2(\bar{\boldsymbol{A}}_i, \boldsymbol{S}_j) \\ &= \frac{1}{N}\sum_{j=1}^{N}\sum_{x=1}^{k} DTW^2(\bar{\boldsymbol{A}}_i^x, \boldsymbol{S}_j^x) \\ &= \sum_{x=1}^{k} d_D(\bar{\boldsymbol{A}}_i^x, \mathbb{S}^x) \end{aligned} \tag{16}$$



(a)

(b)

Fig. 6. Illustration of subsequence optimization process. (a) Illustration in the compression optimization process. (b) Illustration in the stretch optimization process.

Similar to Eq. (13) and Eq. (14) in Lemma 2, the relationships between $DTW^2(\bar{\boldsymbol{A}}_{i+1}, \boldsymbol{S}_j)$ and $DTW^2(\bar{\boldsymbol{A}}_{i+1}^x, \boldsymbol{S}_j^x), x = 1, \ldots, k$ satisfy

$$DTW^2(\bar{\boldsymbol{A}}_{i+1}, \boldsymbol{S}_j) \le \sum_{x=1}^{k} DTW^2(\bar{\boldsymbol{A}}_{i+1}^x, \boldsymbol{S}_j^x) \tag{17}$$

Based on Eq. (17), the average discrepancy distance between $\bar{\boldsymbol{A}}_{i+1}$ and $\mathbb{S}$ can be expressed as

$$\begin{aligned} d_D(\bar{\boldsymbol{A}}_{i+1}, \mathbb{S}) &= \frac{1}{N}\sum_{j=1}^{N} DTW^2(\bar{\boldsymbol{A}}_{i+1}, \boldsymbol{S}_j) \\ &\le \frac{1}{N}\sum_{j=1}^{N}\sum_{x=1}^{k} DTW^2(\bar{\boldsymbol{A}}_{i+1}^x, \boldsymbol{S}_j^x) \\ &\le \sum_{x=1}^{k} d_D(\bar{\boldsymbol{A}}_{i+1}^x, \mathbb{S}^x) \end{aligned} \tag{18}$$

Because DBA is a convergent method and the updated average subsequence has the minimum local average discrepancy

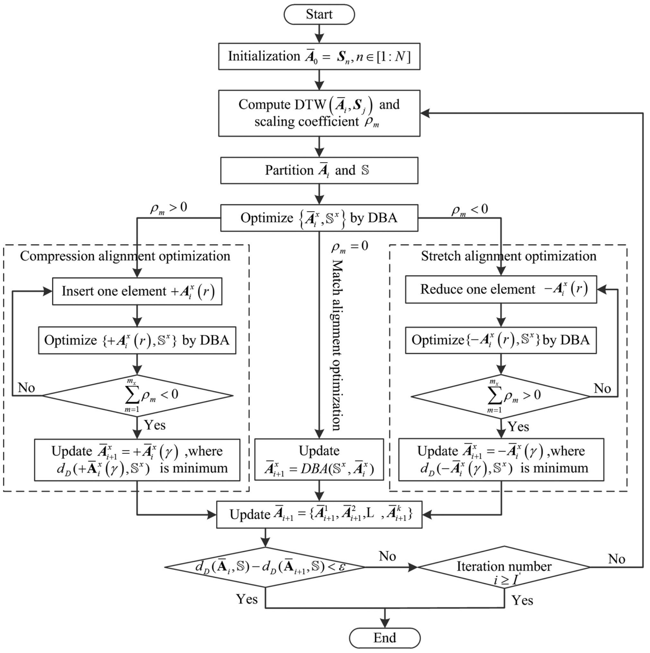

Start
Initialization $\bar{A}_0 = S_n, n \in [1:N]$
Compute DTW$(\bar{A}_i, S_j)$ and scaling coefficient ρ_m
Partition $\bar{A}_i$ and $\mathbb{S}$
Optimize $\{\bar{A}_i^x, \mathbb{S}^x\}$ by DBA
$\rho_m > 0$
$\rho_m < 0$
$\rho_m = 0$
Compression alignment optimization
Insert one element $+A_i^x(r)$
Optimize $\{+A_i^x(r), \mathbb{S}^x\}$ by DBA
$\sum_{m=1}^{m_x} \rho_m < 0$
No
Yes
Update $\bar{A}_{i+1}^x = +\bar{A}_i^x(\gamma)$,where $d_D(+\bar{\mathbf{A}}_i^x(\gamma), \mathbb{S}^x)$ is minimum
Match alignment optimization
Update $\bar{A}_{i+1}^x = DBA(\mathbb{S}^x, \bar{A}_i^x)$
Stretch alignment optimization
Reduce one element $-A_i^x(r)$
Optimize$\{-A_i^x(r), \mathbb{S}^x\}$by DBA
$\sum_{m=1}^{m_x} \rho_m > 0$
No
Yes
Update $\bar{A}_{i+1}^x = -\bar{A}_i^x(\gamma)$,where $d_D(-\bar{A}_i^x(\gamma), \mathbb{S}^x)$ is minimum
Update $\bar{A}_{i+1} = \{\bar{A}_{i+1}^1, \bar{A}_{i+1}^2, \text{L} \ , \bar{A}_{i+1}^k\}$
$d_D(\bar{\mathbf{A}}_i, \mathbb{S}) - d_D(\bar{\mathbf{A}}_{i+1}, \mathbb{S}) < \varepsilon$
No
Yes
Iteration number $i \geq I'$
No
Yes
End

Fig. 7. Flow chart of ADBA algorithm.

distance, we have

$$d_D(\bar{\boldsymbol{A}}_{i+1}^x, \mathbb{S}^x) \leq d_D(\bar{\boldsymbol{A}}_i^x, \mathbb{S}^x), x = 1, 2, \ldots, k \tag{19}$$

Combining Eq. (16), Eq. (18), and Eq. (19), we obtain

$$d_D(\bar{\boldsymbol{A}}_{i+1}, \mathbb{S}) \leq d_D(\bar{\boldsymbol{A}}_i, \mathbb{S}) \tag{20}$$

Thus, ADBA is a convergent method. ■

By Theorem 1, the ideal termination criterion of ADBA is that there is no change in the updated average sequence, i.e., $\bar{\boldsymbol{A}}_{i+1} = \bar{\boldsymbol{A}}_i$. However, this termination criterion may result in infeasible computational cost in practices. A compromised termination criterion between precious and efficiency is proposed. The iteration procedure terminates when a maximum number of iterations I' has been exceed, or when the average discrepancy distance converges to a certain value after several iterations, i.e., $d_D(\bar{\boldsymbol{A}}_i, \mathbb{S}) - d_D(\bar{\boldsymbol{A}}_{i+1}, \mathbb{S}) < \varepsilon$. The flow chart of the proposed ADBA algorithm is summarized in Fig. 7.

IV. Experiment Analysis

This section describes the performance evaluation of the proposed ADBA algorithm. First, different tests are carried out on the standard datasets from the UCR time series classification archive [32]. Then, the proposed algorithm is verified by applying it to an accelerometer-based hand gesture recognition system.

Evaluating an average sequence is not a straightforward task because no ground truth of the optimal average sequence is available. In our experiments, the average discrepancy distance is utilized to quantify an averaging method. Moreover, a new indicator, overall temporal aberration τ, is defined to quantify the overall temporal aberration between the average sequence and target sequence set; it is calculated by the following formula:

$$\tau = \frac{1}{\bar{l}} \sum_{m=1}^{\bar{l}} |\rho_m| \tag{21}$$

where $\bar{l}$ is the length of the resulting average sequence.

A. Experiments on UCR Datasets

In these experiments, 20 datasets of the UCR time series classification archive are selected, as shown in Table III. These datasets contain several types of data, such as images, motion data, ECG data, and simulated data. The number of classes for each dataset varies between two and 50 according to the dataset. The sequences of the same dataset have identical length. It should be noted that the sequences in the datasets have been normalized with Z-score.

For the reproducibility of these experiments, the details of the experimental settings are provided as follows:

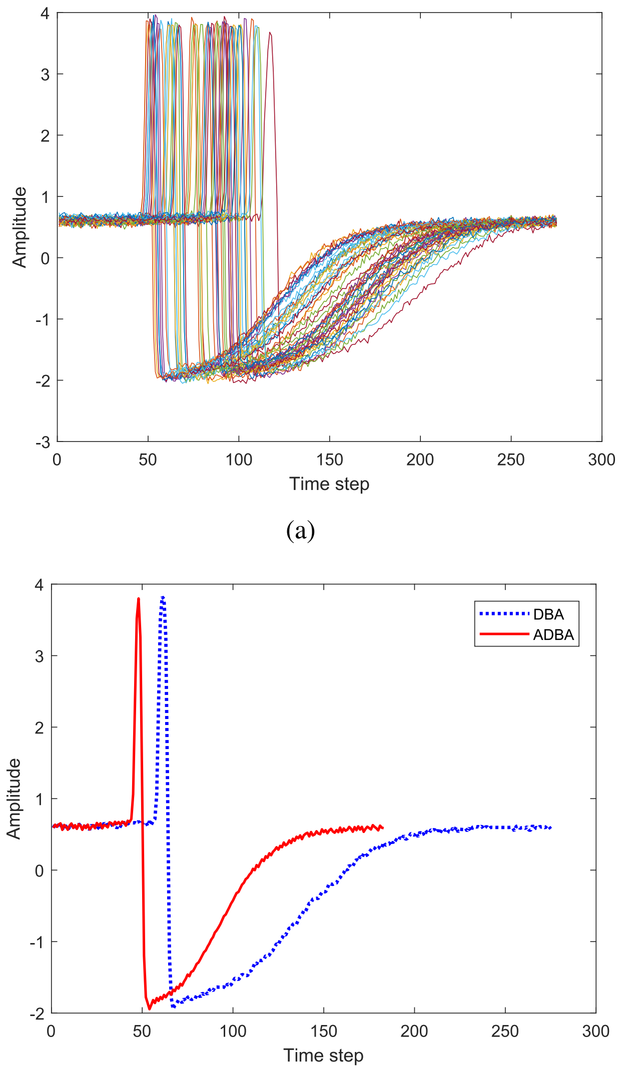
- The training data and test data are merged for each class of each dataset, and the average sequence is computed for each class.
- The local distance between two elements of sequences is the squared Euclidean distance.
- The initial average sequence is randomly selected from the same class, and both BDA and ADBA use similar initialization. In order to reduce randomness, ten trials with different random initializations are conducted to obtain the mean result for each class. The result of each dataset is obtained by averaging the results of all classes.
- The maximum number of iterations used for DBA and ADBA is 15, and the value of the convergence threshold used for DBA and ADBA is 0.01.
- For the purpose of quantitative analysis, the percentage of improvement c_p is calculated as follows:

$$c_p = \frac{|\ V_{ADBA} - V_i\ |}{V_i} \times 100\% \tag{22}$$

where V is the average discrepancy distance or overall temporal aberration. V_{ADBA} is the result obtained by ADBA, and V_i is the result obtained by available methods such as DBA, SSG and COMASA.

1) General Performance Comparison: Figure 8(a) shows the sequences from the first class of the Trace dataset. The average sequences $\bar{\boldsymbol{A}}_{DBA}$ and $\bar{\boldsymbol{A}}_{ADBA}$ are calculated by DBA and ADBA with similar initialization, as shown in Fig. 8(b). Figure 8(b) shows that the waveforms of these two average sequences are similar, whereas $\bar{\boldsymbol{A}}_{ADBA}$ has a shorter length than $\bar{\boldsymbol{A}}_{DBA}$. Because the computational complexity of DTW is related to the length of the sequences, an average sequence with shorter length is likely to benefit further processing. The average discrepancy distances of $\bar{\boldsymbol{A}}_{DBA}$ and $\bar{\boldsymbol{A}}_{ADBA}$ are 0.48 and 0.43, respectively. ADBA achieves a lower average discrepancy distance with the improvement of 10.42%. The smaller average discrepancy distance indicates that ADBA produces a more accurate average result.

Next, we further investigate these two methods based on the criterion of overall temporal aberration. The scaling coefficients



(a)

(b)

Fig. 8. Example of time sequence averaging. (a) Target sequence set(the first class in the Trace dataset). (b) Average sequences calculated by DBA and ADBA with similar initialization.

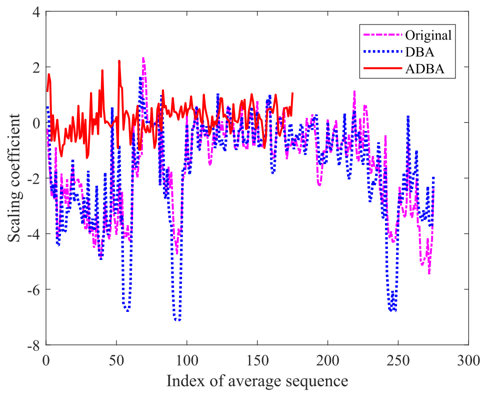


Fig. 9. Scaling coefficients of inertial average sequence and average sequences calculated by DBA and ADBA.

TABLE I
Comparison Results of Intraclass Inertia Over Ten Trials

Dataset	DBA	SSG	COMASA	ADBA
50Words	6.21	6.13	4.71	**4.62**
Adiac	0.17	0.17	0.15	**0.14**
Beef	9.5	9.32	6.12	**5.93**
CBF	13.34	12.59	12.51	**11.83**
Coffee	0.55	0.53	0.47	**0.46**
ECG200	6.95	6.76	6.29	**5.96**
FaceAll	14.73	14.51	13.83	**13.39**
FaceFour	24.87	24.6	21.91	**20.17**
FISH	1.02	0.99	0.94	**0.91**
Gun Point	2.46	2.51	2.03	**1.98**
Lighting2	77.57	71.97	67.46	**66.06**
Lighting7	28.77	29.11	26.27	**26.02**
OliveOil	0.018	0.017	0.017	**0.017**
OSULeaf	22.69	21.4	19.55	**19.21**
SwedishLeaf	2.21	1.91	1.81	**1.78**
Synthetic Control	9.28	8.94	8.7	**8.15**
Trace	0.92	0.74	0.6	**0.56**
Two Patterns	8.66	8.37	7.42	**6.97**
Wafer	30.4	22.58	24.27	**17.24**
Yoga	37.27	11.53	11.1	**10.84**

of the initial average sequence are illustrated in Fig. 9. After the averaging process, the scaling coefficients of the average sequences obtained with DBA and ADBA are shown in Fig. 9. The overall temporal aberrations of these two average sequences are 1.89 and 0.47. Apparently, $\bar{\boldsymbol{A}}_{ADBA}$ has a smaller overall temporal aberration than $\bar{\boldsymbol{A}}_{DBA}$. By comparing the scaling coefficients before and after the averaging process, we observe that ADBA significantly reduces the temporary aberrations compared to DBA. For $\bar{\boldsymbol{A}}_{DBA}$, the majority of the scaling coefficients are significantly smaller than zero. It indicates that certain segments of $\bar{\boldsymbol{A}}_{DBA}$ are stretched with respect to the original sequence set. For $\bar{\boldsymbol{A}}_{ADBA}$, the scaling coefficients fluctuate marginally around zero. This is the reason why $\bar{\boldsymbol{A}}_{ADBA}$ is shorter than $\bar{\boldsymbol{A}}_{DBA}$.

Table I presents the comparisons of the intraclass inertia obtained by DBA [19], SSG [27], COMASA [31], and ADBA for each dataset. The smaller the value of average intraclass inertia is, the better the averaging result is. Table I illustrates that ADBA achieves a lower intraclass inertia with respect to the other methods for all the datasets. In contrast, ADBA outperforms DBA, SSG, and COMASA methods, with an average improvement of 21.54%, 14.0%, and 4.9%, respectively. Compared to DBA, ADBA reduces the intraclass inertia from 5.56% for the OliveOil dataset to 70.91% for the Yoga dataset. For the OliveOil dataset, the three improved algorithms yield a marginally lower intraclass inertia than DBA. Further investigation of this particular dataset reveals that there is almost no difference between these corresponding sequences.

The comparisons of the overall temporal aberration for 20 datasets are reported in Table II. Table II illustrates that ADBA achieves a lower overall temporal aberration compared to DBA

TABLE II
COMPARISONS OF OVERALL TEMPORAL ABERRATION τ OVER TEN TRIALS

Dataset	DBA	ADBA	$c_p(\%)$
50Words	10.37	**1.04**	89.99
Adiac	0.86	**0.35**	58.88
Beef	20.54	**1.63**	92.06
CBF	4.06	**0.51**	87.37
Coffee	2.52	**0.42**	83.39
ECG200	4.97	**0.67**	86.54
FaceAll	1.96	**0.75**	61.73
FaceFour	4.03	**0.65**	83.92
FISH	2.74	**0.38**	86.22
Gun Point	20.3	**1.46**	92.82
Lighting2	83.22	**2.34**	97.19
Lighting7	27.02	**1.65**	93.89
OliveOil	0.51	**0.15**	70.45
OSULeaf	11.81	**1.12**	90.48
SwedishLeaf	2.18	**0.88**	59.63
Synthetic Control	2.43	**0.92**	62.14
Trace	28.13	**1.31**	95.35
Two Patterns	7.99	**0.93**	88.32
Wafer	40.04	**3.91**	90.24
Yoga	16.78	**1.12**	93.33

for all the datasets, with an average reduction of 83.2%. For each dataset, the overall temporal aberration computed by ADBA is substantially smaller than that by DBA, and a majority of them are less than one. Because there are marginally temporal variations in the OliveOil dataset, the overall temporal aberrations obtained with both the DBA and ADBA algorithms are more lower than those for the other datasets. Table II indicates that ADBA is capable of reducing the temporal aberrations between the average sequence and target sequence set significantly. The experimental results in Table I and II demonstrate the superiority of the proposed ADBA algorithm over the DBA method with respect to both criteria of intraclass inertia and overall temporal aberration.

2) Impact of Initialization: Similar to DBA, ADBA starts with an initial average sequence. In this experiment, we evaluate the impact of initialization by randomly selecting different sequences from the sequence set as the initial average sequence.

For the second class of the Trace dataset, each sequence of this dataset is selected as the initial average sequence. With these initializations, the average sequences calculated by DBA and ADBA are shown in Fig. 10(a) and Fig. 10(b), respectively. We observe that the waveforms of the average sequences calculated by ADBA exhibit a higher consistency with respect to DBA. The lengths of the average sequences obtained by ADBA are significantly shorter than those obtained by DBA, by almost two-thirds. As explained previously, sequences of the same class have identical length, and there are redundancies in the dataset. ADBA is capable of removing these redundancies to obtain a more precise and concise average sequence. Figure 11(a) and Fig. 11(b) present the variation in the average discrepancy distance and overall temporal aberration computed by DBA and ADBA. For each initialization, the average discrepancy distance and overall temporal aberration computed by ADBA are substantially smaller than those computed by DBA. With different initializations, the average discrepancy distance and overall temporal aberration computed by ADBA are more stable than those computed by DBA.

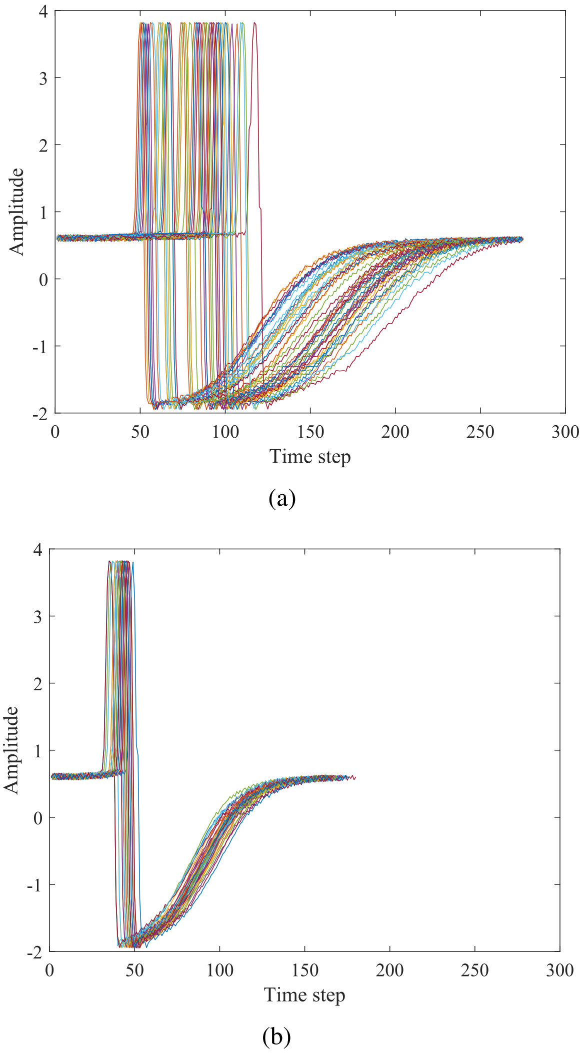
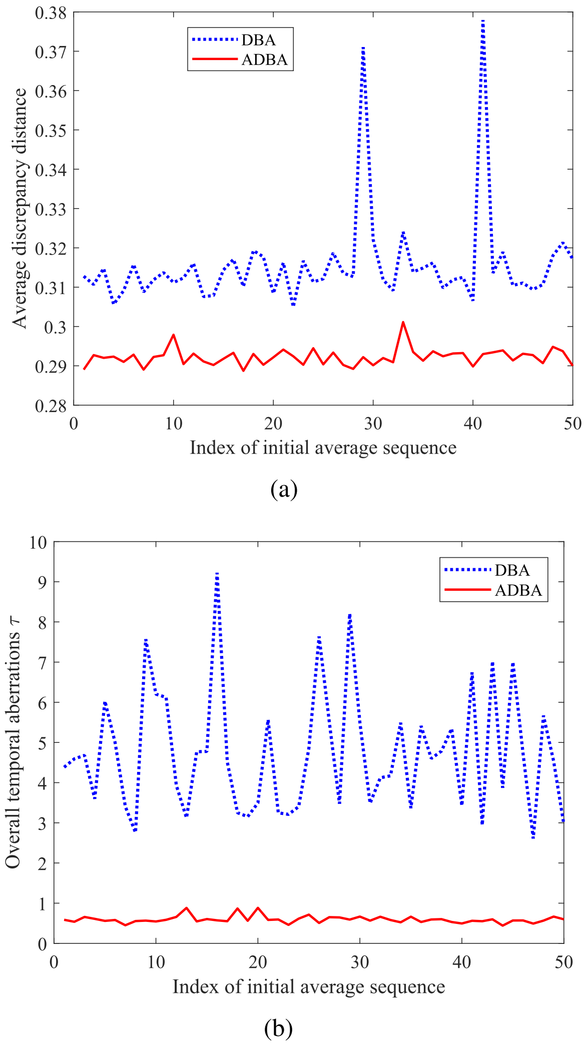


Fig. 10. Average sequences computed with different initializations. (a) Average sequences computed with DBA. (b) Average sequences obtained with ADBA.

Because this experiment requires substantial computation, we perform 30 trials with different randomly selected initial average sequences on a few datasets. The means and standard deviations of the average discrepancy distance and overall temporal aberration calculated by DBA and ADBA for eight datasets are shown in Fig. 12 and Fig. 13, respectively. These figures reveal that ADBA produces a lower average discrepancy distance and overall temporal aberration for all of the datasets irrespective of the selected initial average sequence. Even the best result obtained by DBA is worse than the worst result obtained by ADBA. The



(a)

(b)

Fig. 11. Performance comparison with different initializations. (a) Comparison of overall temporal aberration. (b) Comparison of average discrepancy distance.

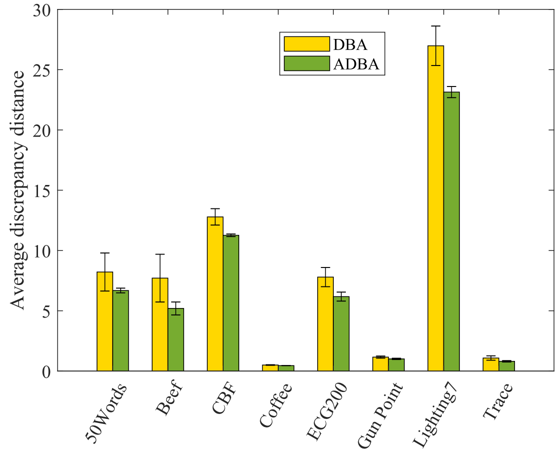


Fig. 12. Average discrepancy distance of resulting average sequences obtained by DBA and ADBA with different initializations.

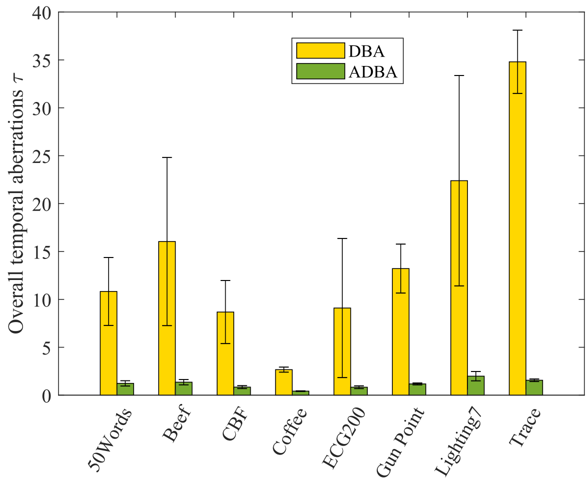


Fig. 13. Overall temporal aberration of resulting average sequences obtained by DBA and ADBA with different initializations.

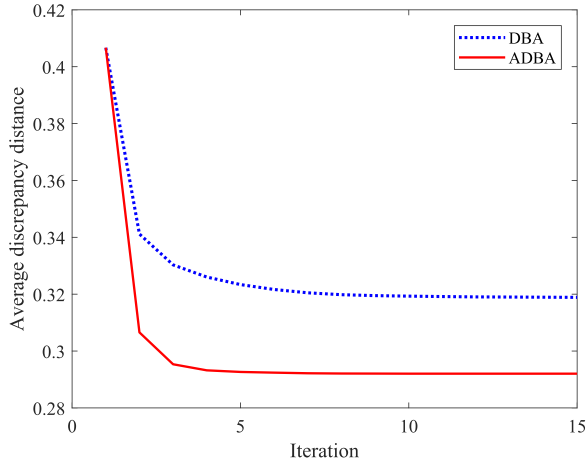


Fig. 14. Typical converging behaviors of DBA and ADBA.

standard deviation of the average discrepancy distance and overall temporal aberration can reflect the sensitivity of an averaging algorithm to different initializations. The experimental results indicate that ADBA is more robust to initialization than DBA is.

3) Convergence of the Iteration Process: Because both DBA and ADBA are iterative algorithms, we evaluate the convergence properties of the two methods. Given similar initialization, the typical converging behaviors of DBA and ADBA on the first class of the Trace dataset are reported in Fig. 14. Figure 14 shows that the final convergence value obtained with ADBA is smaller than that with DBA. Moreover, ADBA converges to the final value more rapidly than DBA does. The number of iterations of the ADBA algorithm is less than ten, including for the worse case. The experimental result reveals that ADBA improves both the speed of convergence and the quality of solution from those for DBA.

4) Runtime: The ADBA and DBA algorithms were programmed and implemented on a desktop (Intel Core i5-7400 CPU @ 3.0 GHz with 8-GB RAM) using Matlab R2017b. Table III reports the runtimes of ADBA compared to DBA. The runtime depends on the number of classes C, the length of

TABLE III
THE CHARACTERISTICS OF UCR DATASETS, AND COMPARISONS OF RUNTIME (IN SECONDS) BETWEEN DBA AND ADBA OVER TEN ITERATIONS. THE NUMBER OF CLASSES C, THE LENGTH OF SEQUENCE n, AND THE SAMPLE SIZE N

Method	C	n	N	Runtime(s)	
				DBA	ADBA
50Words	50	270	905	0.26	20.1
Adiac	37	176	781	0.08	1.69
Beef	5	470	60	0.45	5.12
CBF	3	128	925	1.74	85.96
Coffee	3	286	56	0.37	5.17
ECG200	2	96	200	0.39	14.14
FaceAll	14	131	2570	1.64	18.32
FaceFour	4	350	112	0.95	42.6
FISH	7	463	350	0.62	12.01
Gun Point	2	150	200	0.75	16.13
Lighting2	2	637	121	3.96	95.0
Lighting7	7	319	143	0.37	7.22
OliveOil	4	570	60	2.29	61.04
OSULeaf	6	427	442	0.14	1.28
SwedishLeaf	15	128	1125	0.40	9.75
Synthetic Control	6	60	600	0.69	10.12
Trace	4	275	200	6.94	384.59
Two Patterns	4	128	5000	0.24	9.8
Wafer	2	152	7174	28.03	1424.2
Yoga	2	426	3300	57.4	2179.4

sequence n, and the sample size N. As expected, ADBA is significantly more time-consuming than DBA because the subsequence optimization processes require numerous operations for each iteration. For each dataset, ADBA requires more runtime than DBA does, with an increase of 9.3-55.4 times. Compared to COMASA [31], ADBA achieves better results with less computation time.

For most applications, the averaging processing is a subroutine of the training stage, which can be conducted off-line. For the application that is sensitive to execution time, there are numerous techniques to speed up ADBA. First, the global constraints such as Sakoe-Chiba [21] band and Itakura parallelogram [33] can be utilized to speed up the DTW computation. Secondly, the subsequence optimization processes can be parallelly implemented for multi-core CPU systems. Thirdly, a heuristic strategy that can merge or insert multiple elements in an iteration is introduced to the subsequence optimization process.

B. Application to Hand Gesture Recognition

In this subsection, we evaluate our approach by a DTW-based template matching (DTW-TM) hand gesture recognition system. In this experiment, a hand gesture vocabulary of eight gesture classes (up, down, left, right, clockwise, anticlockwise, tick, and cross) is defined, as shown in Fig. 15.

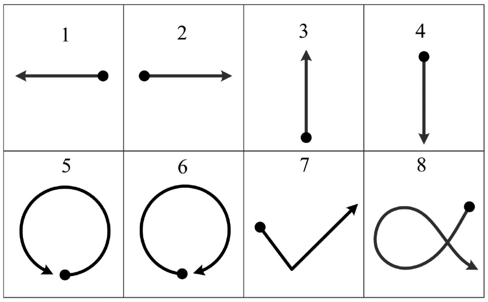


Fig. 15. Hand gesture vocabulary.

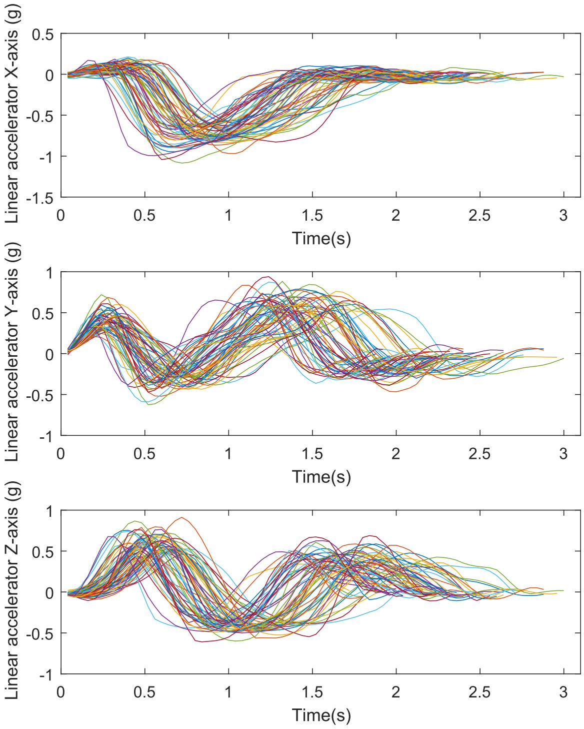


Fig. 16. 3D acceleration signals of gesture ”clockwise”. The sampling rate is 50 Hz.

The gesture dataset was acquired from five participants (two females and three males) utilizing a wrist-worn inertial sensor containing 2,400 gestures samples. Figure 16 illustrates the 3D acceleration signals of the gesture “clockwise” performed by a participant. Unlike the UCR dataset, the gesture sequences exhibit different length because different individuals perform gestures with different speeds, amplitudes, and styles. The average length of these gesture samples is 57.2.

For the sequences shown in Fig. 16, a shortest sequence and a longest sequence are selected as the initial average sequences. The average sequences calculated by DBA and ADBA with these initializations are shown in Fig. 17. The detailed results are presented in Table IV. The lengths of the average sequences obtained by ADBA with two initializations are almost equal to the average length of the sequence set. Table IV illustrates that ADBA achieves a lower average discrepancy distance and overall temporal aberration irrespective of how the initial average sequence is selected. The gesture templates calculated

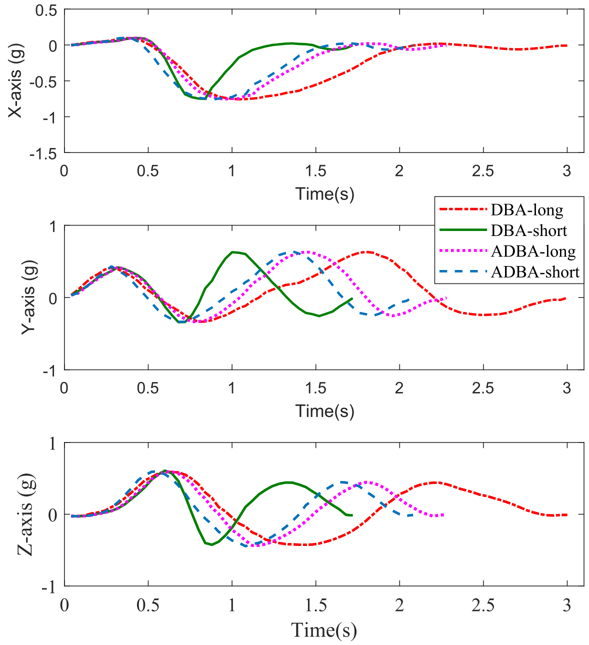


Fig. 17. The average sequences of gesture "clockwise" training samples obtained with DBA and ADBA.

TABLE IV
COMPARISONS OF AVERAGE DISCREPANCY DISTANCE, OVERALL TEMPORAL ABERRATION τ AND LENGTH OBTAINED BY DBA AND ADBA WITH DIFFERENT INITIALIZATIONS

Indicator	Shortest initialization		Longest initialization	
	DBA	ADBA	DBA	ADBA
Distance	0.12	0.09	0.16	0.11
τ	0.62	0.3	0.70	0.34
Length	43	53	75	57

by ADBA exhibit moderate speed and amplitude, which can more effectively preserve the pattern and characteristics of hand motion.

The database was randomly divided into two groups: training data (20%) and testing data (80%). The training data was utilized for generating the gesture templates, and the testing data was utilized for evaluating the gesture recognition algorithm. The gesture template was selected in four manners:

- Random selection method: the gesture template is randomly selected from the training samples;
- Minimum selection method: the sample having the smallest intraclass inertia is selected as the gesture template;
- DBA selection method: the average sequence calculated by DBA is selected as the gesture template;
- ADBA selection method: the average sequence calculated by ADBA is selected as the gesture template.

The recognition rates of four template generation methods are reported in Table V. Obviously, the gesture templates calculated by ADBA outperform the random, minimum, and DBA template generation methods for both user-dependent and user-independent tests.

TABLE V
RECOGNITION RATES OF THREE TEMPLATE GENERATION METHOD

Method	Recognition rates(%)	
	user-dependent	user-independent
Random	80.2	74.5
Minimum	87.3	82.4
DBA	93.1	86.7
ADBA	98.6	92.8

V. CONCLUSIONS

As a popular similarity measure, DTW has been a commonly used tool for sequence analysis in numerous applications. However, time sequence averaging under DTW is highly challenging. Several attempts have been made to obtain a sub-optimal solution with acceptable computational complexity. Compared to these available approaches, DBA has emerged as a remarkable global approximation algorithm. Notwithstanding the success of DBA, the selection of the length of the average sequence is still not a straightforward problem. In [19], an adaptive scaling technique is introduced to shorten the length of the average sequence according to the reduction in the discrepancy distance. This technique is a compressing scheme.

In this work, we introduce a scaling coefficient to quantify the temporal aberrations between the temporary average sequence and the sequences set. A systematic adaptive scheme is proposed for optimizing the length and values of the average sequence. We have demonstrated that both sequence compression and sequence stretching can optimize the resulting average sequence. The length of the resulted average sequence is not constrained by the length of the selected inertial average sequence and can be adjusted adaptively through local stretching and compression during the iteration process. As a result, the average discrepancy distance and overall temporal aberration are minimized.

Experiments carried out on the standard datasets illustrate that the proposed ADBA algorithm is robust to initialization and achieves higher performance compared with the available methods on all the tested datasets. Additionally, the application to an accelerometer-based hand gesture recognition system demonstrates the superiority of our proposed method over the current available approaches. In these experiments, we also note that a large amount of computation is required for the iteration calculations, and this makes ADBA marginally more time-consuming. Hence, it is necessary to further develop a fast ADBA algorithm for practical applications.

REFERENCES

[1] L. Gupta, D. L. Molfese, R. Tammana, and P. G. Simos, "Nonlinear alignment and averaging for estimating the evoked potential," *IEEE Trans. Biomed. Eng.*, vol. 43, no. 4, pp. 348–356, Apr. 1996.

[2] T. C. Fu, "A review on time series data mining," *Eng. Appl. Artif. Intell.*, vol. 24, no. 1, pp. 164–181, 2011.

[3] L. R. Rabiner and J. G. Wilpon, "Considerations in applying clustering techniques to speaker-independent word recognition," *J. Acoust. Soc. Amer.*, vol. 66, no. 3, pp. 663–673, 1979.

[4] S. Boudaoud and H. Rix, "Corrected integral shape averaging applied to obstructive sleep apnea detection from the electrocardiogram," *EURASIP J. Adv. Signal Process.*, vol. 2007, 2007, Art. no. 032570.

[5] S. Seto, W. Zhang, and Y. Zhou, "Multivariate time series classification using dynamic time warping template selection for human activity recognition," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2015, pp. 1399–1406.

[6] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh, "Dynamic time warping averaging of time series allows faster and more accurate classification," in *Proc. IEEE Int. Conf. Data Mining*, 2014, pp. 470–479.

[7] M. Morel, C. Achard, R. Kulpa, and S. Dubuisson, "Time-series averaging using constrained dynamic time warping with tolerance," *Pattern Recognit.*, vol. 74, pp. 77–89, 2018.

[8] S. Soheily-Khah, A. Douzal-Chouakria, and E. Gaussier, "Generalized k-means-based clustering for temporal data under weighted and kernel time warp," *Pattern Recognit. Lett.*, vol. 75, pp. 63–69, 2016.

[9] V. Hautamaki, P. Nykanen, and P. Franti, "Time-series clustering by approximate prototypes," in *Proc. 19th Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.

[10] W. H. Abdulla, D. Chow, and G. Sin, "Cross-words reference template for DTW-based speech recognition systems," in *Proc. Conf. Convergent Technol. Asia–Pac. Region*, 2003, vol. 4, pp. 1576–1579.

[11] W. Meesrikamolkul, V. Niennattrakul, and C. A. Ratanamahatana, "Shape-based clustering for time series data," in *Proc. Pac.–Asia Conf. Adv. Knowl. Discovery Data Mining*, 2012, pp. 530–541.

[12] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014.

[13] P. Sathianwiriyakhun, T. Janyalikit, and C. A. Ratanamahatana, "Fast and accurate template averaging for time series classification," in *Proc. 8th Int. Conf. Knowl. Smart Technol.*, 2016, pp. 49–54.

[14] V. Niennattrakul, D. Srisai, and C. A. Ratanamahatana, "Shape-based template matching for time series data," *Knowl.-Based Syst.*, vol. 26, pp. 1–8, 2012.

[15] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. Keogh, "Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm," *Knowl. Inf. Syst.*, vol. 47, no. 1, pp. 1–26, 2016.

[16] P. Jonsson and L. Eklundh, "Seasonality extraction by function fitting to time-series of satellite sensor data," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 8, pp. 1824–1832, Aug. 2002.

[17] L. Zhu and L. Najafizadeh, "Dynamic time warping-based averaging framework for functional near-infrared spectroscopy brain imaging studies," *J. Biomed. Opt.*, vol. 22, no. 6, 2017, Art. no. 066011.

[18] M. Kotas, T. Pander, and J. M. Leski, "Averaging of nonlinearly aligned signal cycles for noise suppression," *Biomed. Signal Process. Control*, vol. 21, pp. 157–168, 2015.

[19] F. Petitjean, A. Ketterlin, and P. Ganarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognit.*, vol. 44, no. 3, pp. 678–693, 2011.

[20] H. Sakoe and S. Chiba, "A dynamic programming approach to continuous speech recognition," in *Proc. 7th Int. Congr. Acoust.*, 1971, vol. 3, pp. 65–69.

[21] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978.

[22] J. F. Lichtenauer, E. A. Hendriks, and M. J. T. Reinders, "Sign language recognition by combining statistical DTW and independent classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 2040–2046, Nov. 2008.

[23] A. Akl, C. Feng, and S. Valaee, "A novel accelerometer-based gesture recognition system," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6197–6205, Dec. 2011.

[24] Y. L. Hsu, C. L. Chu, Y. J. Tsai, and J. S. Wang, "An inertial pen with dynamic time warping recognizer for handwriting and gesture recognition," *IEEE Sensors J.*, vol. 15, no. 1, pp. 154–163, Jan. 2015.

[25] V. Niennattrakul and C. A. Ratanamahatana, "On clustering multimedia time series data using K-means and dynamic time warping," in *Proc. Int. Conf. Multimedia Ubiquitous Eng.*, 2007, pp. 733–738.

[26] M. Brill *et al.*, "Exact mean computation in dynamic time warping spaces," *Data Mining Knowl. Discovery*, vol. 33, no. 1, pp. 252–291, Jan. 2019.

[27] D. Schultz and B. Jain, "Nonsmooth analysis and subgradient methods for averaging in dynamic time warping spaces," *Pattern Recognit.*, vol. 74, pp. 340–358, 2018.

[28] V. Niennattrakul and C. A. Ratanamahatana, "Shape averaging under time warping," in *Proc. 6th Int. Conf. Elect. Eng./Electron., Comput., Telecommun. Inf. Technol.*, 2009, vol. 2, pp. 626–629.

[29] S. Ongwattanakul and D. Srisai, "Contrast enhanced dynamic time warping distance for time series shape averaging classification," in *Proc. 2nd Int. Conf. Interact. Sci*, 2009, pp. 976–981.

[30] T. Sun, H. Liu, H. Yu, and C. L. Chen, "Degree-pruning dynamic programming approaches to central time series minimizing dynamic time warping distance," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1719–1729, Jul. 2017.

[31] F. Petitjean and P. Ganarski, "Summarizing a set of time series by averaging: From Steiner sequence to compact multiple alignment," *Theor. Comput. Sci.*, vol. 414, no. 1, pp. 76–91, 2012.

[32] Y. Chen *et al.*, "The UCR time series classification archive," Jul. 2015. [Online]. Available: www.cs.ucr.edu/ eamonn/time_series_data/

[33] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-23, no. 1, pp. 67–72, Feb. 1975.

Yu-Tao Liu received the B.E. degree in electrical engineering and automation from Henan University of Urban Construction, Pingdingshan, China, in 2012, and the M.S. degree in control science and engineering, in 2014, from the Harbin Institute of Technology, Harbin, China, where he is currently working toward the Ph.D. degree in control science and engineering. His current research interests include wearable systems and human motion analysis.

Yong-An Zhang received the B.E. and Ph.D. degrees in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 1999 and 2004, respectively. From 2005 to 2006, he was a Postdoctoral Scholar with the HIT Shenzhen Graduate School, Shenzhen, China. From 2007 to 2008, he was with the School of Astronautics, Beihang University, Beijing, China. Since 2008, he has been an Associate Professor with the School of Astronautics, Harbin Institute of Technology. His current research interests include nonlinear estimation, human motion analysis, and robot motion control.

Ming Zeng received the B.E. degree in automation engineering and M.S. degree in theoretical electricity from Harbin University of Science and Technology, Harbin, China, in 1981 and 1988, respectively, and the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 1999. He is currently a Professor with the Space Control and Inertial Technology Research Center, Harbin Institute of Technology. His research interests include signal detection and processing, inertial technology testing, and intelligent control.