Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching

Pierre-François Marteau, *Member*, *IEEE Computer Society*

Abstract—In a way similar to the string-to-string correction problem, we address discrete time series similarity in light of a time-series-to-time-series-correction problem for which the similarity between two time series is measured as the minimum cost sequence of edit operations needed to transform one time series into another. To define the edit operations, we use the paradigm of a graphical editing process and end up with a dynamic programming algorithm that we call Time Warp Edit Distance (TWED). TWED is slightly different in form from Dynamic Time Warping (DTW), Longest Common Subsequence (LCSS), or Edit Distance with Real Penalty (ERP) algorithms. In particular, it highlights a parameter that controls a kind of stiffness of the elastic measure along the time axis. We show that the similarity provided by TWED is a potentially useful metric in time series retrieval applications since it could benefit from the triangular inequality property to speed up the retrieval process while tuning the parameters of the elastic measure. In that context, a lower bound is derived to link the matching of time series into downsampled representation spaces to the matching into the original space. The empiric quality of the TWED distance is evaluated on a simple classification task. Compared to Edit Distance, DTW, LCSS, and ERP, TWED has proved to be quite effective on the considered experimental task.

Index Terms—Pattern recognition, time series, algorithms, similarity measures.

1 Introduction

More and more computer applications are faced with the problem of searching large data sets for time series that are close to a given query element under some similarity criteria. Financial and stock data analysis [37], moving objects identification [5], [27] astronomy, medicine [19], meteorology, data mining [1], time-stamped event data processing [36], and network monitoring [28] are but a few of the numerous examples that could be cited. All these applications embed time series in a representation space and exploit some similarity measure defined for this space. Similarity measures fall basically into three categories:

1. nonelastic metrics such as Lp-norms that do not support time shifting, such as Euclidean Distance (ED) and Correlation,
2. elastic similarity measures that tolerate time shifting but are not metrics, such as Dynamic Time Warping (DTW) [33], [31] or Longest Common Subsequence (LCSS) [6], [34], and
3. elastic metrics that tolerate time shifting, such as Edit distance with Real Penalty (ERP) [4].

When considering time series information retrieval, working in a metric space can be appealing because a lot of data structures (essentially tree-based structures) and algorithms (partitioning, pivoting, etc.) have been optimized and made available for efficiently indexing and retrieving objects in metric spaces: see [3] for a review. All these structures and algorithms take advantage of the triangle inequality that allows for the efficient pruning of a large number of time series that are too far away from the query. For some nonmetric measures, all these data structures can still be used if a lower bounding approximation, which needs to be a metric, is available. A lower bound of the sort exists for both LCSS and DTW, as detailed in [34]. Furthermore, the need for processing time-stamped data (event data or data that are not sampled coherently) is becoming particularly significant [36], [28] in stock analysis, network monitoring, fault analysis, etc. In this paper, we address the case of elastic metrics, namely, elastic similarity measures that jointly exploit time shifting (measured using time stamps or sample indices) and possess all the properties of a distance, in particular, the triangle inequality. Our contribution is basically fourfold:

- The first contribution of this paper is the proposal of a new elastic metric, which we call Time Warp Edit Distances (TWED). This contribution has to be placed in the perspective of former works that seek to combine Lp-norms with the edit distance, in particular, in the light of the ERP distance [4] that can support local time shifting while being a metric. Other elastic similarity measures that belong to the DTW category are not metrics since they do not satisfy the triangle inequality. Section 2 promotes the need for triangle inequality to process time series in a data compression context based on a downsampling perspective.
- The second contribution is related to the introduction of a parameter we call *stiffness*, which controls the *elasticity* of TWED, placing this kind of distance in between the ED (somehow a distance with *infinite stiffness*) and DTW (somehow a similarity measure with *no stiffness* at all). One of the differences

- *The author is with the Université Européenne de Bretagne-Sud—VALORIA, Campus de Tohannic-Bât. Yves Coppens, BP 573, 56017 Vannes Cedex, France. E-mail: pierre-francois.marteau@univ-ubs.fr.*

Manuscript received 18 Sept. 2006; revised 13 July 2007; accepted 11 Mar. 2008; published online 24 Mar. 2008.
Recommended for acceptance by J. Oliensis.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0665-0906.
Digital Object Identifier no. 10.1109/TPAMI.2008.76.

between TWED and former similarity measures is the use of time-stamp differences between compared samples as part of the local matching costs. The motivation for such a characteristic is also given in Section 2.

- The third contribution proposes a lower bound for the TWED measure, which allows one to link the evaluation of the matching of two time series into downsampled representation spaces to the evaluation of their matching into their original representation spaces.
- The fourth contribution of the paper is an empiric evaluation of the quality of TWED based on a simple classification experiment that provides some highlights on the effectiveness of TWED compared to the ED, DTW, LCSS, and ERP. The influence of the *stiffness* parameter on classification error rates is also analyzed.

The paper is organized as follows: Section 2 addresses the motivation aspects. Section 3 briefly presents the main relevant founding works on elastic distances for time series matching. Section 4 details the definition and implementation of the TWED with stiffness adjustment that is proposed in this paper. Section 5 details a lower bounding procedure we suggest to speed up range queries processing. Section 6 describes a classification experiments that shows the empirical effectiveness of TWED comparatively to the ED and other classical elastic measures. Section 7 concludes the paper and proposes some perspectives.

2 Motivation for a Similarity Measure that Verifies the Triangle Inequality and Takes Time-Stamp Differences into Account

The use of elasticity theory to model the behavior of nonrigid curves, surfaces, and solids as function of time has given rise to a lot of applications in medical image analysis, vision, or computer graphics (see [32] and [25] for surveys). These models are fundamentally dynamic and unify the description of shape and the description of motion. On the another hand, elastic distances have been proposed to define similarity measures that are tolerant to object deformations, in particular stretching or shrinking. Although the analogy with physical models of deformable objects makes sense, we do not extend it too far, since, in the physical sense, the laws that should govern the matching of deformable objects are not always available or costly to cope with. Our motivation for using time stamps (or sample indices) is related to the way we want to control the elasticity of the measure. Differences of indices between match samples have been successfully used to improve elastic measures such as DTW [31] or LCSS measures [11]. The general idea is to limit the elasticity of the measure by using a threshold: If the index difference between two samples that are candidates for a match is lower than the value of the threshold, then matching is allowed; otherwise, it is forbidden. This binary decision might, in some cases, limit the effectiveness of the measure. Keeping in mind the mechanical analogy of a spring (for which the deformation effort is proportional to the stretching or the shrinking), instead of using a threshold, we suggest using the range of the sample index difference to linearly penalize the matching of samples for which the index values are too far and to favor the matching of samples for which the index values are close. In the case where time series are sampled using nonuniform or varying sampling rates, one can benefit from time stamps instead of sample indices since this approach does not require sampling of the data. The second motivation for defining a measure that exploits time stamps (or sample indices) while verifying the triangle inequality is twofold: First, it provides an effective solution for comparing approximated representations of time series but not necessarily by using uniform downsampling methods; second, it establishes a useful relationship between the matching performed in the downsampled space and the matching performed in the original space (Fig. 2). Approximation of multidimensional discrete curves has been widely studied [7], [14], [29] essentially to speed up the data processing required by resource demanding applications. Among other approaches, polygonal approximation of discrete curves has been quite popular recently [29], [17]. The problem can be informally stated as follows: Given a digitized curve X of $N \geq 2$ ordered samples, find K (in general, $K \ll N$) dominant samples among them that define a sequence of piecewise linear segments that most closely approximate the original curve. This problem is known as the $min\text{-}\epsilon$ problem [12]. Numerous algorithms have been proposed for more than 30 years to solve this optimization problem efficiently. Most of them belong either to graph-theoretic dynamic programming or to heuristic approaches. See, for instance [14], [29], [17], [24], among others for details. Such approaches can be used to adaptively downsample time series. For instance, in [23], polygonal curves approximations have been used to downsample gesture signals optimally, and in [24], an elastic matching procedure has been proposed to compare two time series with a linear time complexity. For these approaches, a downsampled time series is a reduced sequence of tuples (sample and time stamps) that corresponds to the end extremities of the polygonal segments. The sampling rate for such downsampled time series is not generally uniform, and time-stamps differences between two successive samples are varying. Downsampling time series can be used to drastically reduce the dimension of the space in which we could potentially process the time series. Nevertheless, one difficulty emerges: How can we compare downsampled time series using nonuniform (e.g. varying) sampling frequencies? Not taking into account the occurring time of the samples could introduce discrepancies between the original space and the downsampled space. For instance, phase or frequency information is potentially lost or at least damaged, as well as the slope of spikes. In this context, the triangle inequality is also of great importance since it maintains distance relations between the original space and the downsampled space. Let X and Y be two time series in the original space and their downsampled counter parts. If δ is a measure for which the triangle inequality holds, then we have $\left|\delta(X,Y)-\delta(\tilde{X},\tilde{Y})\right| \leq \delta(X,\tilde{X})+\delta(Y,\tilde{Y})$ In the case where $\delta(X,\tilde{X})$ and $\delta(Y,\tilde{Y})$ are maintained small by the similarity measure, $\delta(X,\tilde{X})$ and $\delta(Y,\tilde{Y})$ are comparable, and the following inequality gives an exploitable lower bound to the $\delta(X,Y)$ measure:

$$\delta(\tilde{X},\tilde{Y})-\delta(X,\tilde{X})-\delta(Y,\tilde{Y}) \leq \delta(X,Y). \tag{1}$$

This lower bound can be used to significantly speed up the time series information retrieval process since a pruning strategy can be proposed in the downsampled space. We will come back to this issue in Section 5.

3 Elastic Similarity in Light of the Symbolic Edit Distance

In this section, we succinctly present the main elastic measures developed in the literature, from founding work to more recent studies. The Levenshtein Distance (LD) proposed in 1966 [18], also known as the edit distance, is the smallest number of insertions, deletions, and substitutions required to change one string into another. For more than 30 years, the ideas behind LD have been largely reused and extended by various research communities. The main contributions are rapidly reviewed below. In 1974, Wagner and Fisher [35] developed a computationally efficient algorithm to calculate LD in $O(n.m)$ using dynamic programming [2]. Meanwhile, DTW, which shares many similarities with LD despite the fact that it is not a metric, was proposed in 1970 [33] and 1971 [31] to align speech utterances, namely, time series with time shift tolerances. The LCSS similarity measure initially defined for string matching [11] has also been adapted for time series matching [6], [34]. Recently, a lot of fruitful research dealing with DTW and LCSS has been carried out to propose efficient computation and pruning strategies that are required to process massive data [34], [15], [37]. Some work has also been conducted to provide the triangle inequality to DTW: The ERP VII has been proposed as an edit distance-based metric for time series matching with time shift tolerance. The edit distance principle has also been proposed to develop 1D-Point-Patterns Matching (PPM) (point patterns are ascending lists of real values) [20], [21]. The measure proposed to match 1D-PPM is shown to be a metric that can be extended to the multidimensional case at the price of a nonpolynomial complexity. Hereinafter, we present DTW, ERP, and LCSS in the light of the edit distance and develop the TWED metrics as an alternative to ERP.

3.1 Definitions

Let U be the set of finite time series: $U=\{A_1^p|p\in N\}$. A_1^p is a time series with discrete time index varying between 1 and p. We note Ω the empty time series (with null length) and by convention $A_1^0=\Omega$ so that Ω is member of set U. Let A be a finite discrete time series. Let a'_i be the ith sample of time series A. We will consider that $a'_i \in S\times T$, where $S\subset R^d$ with $d\geq 1$ embeds the multidimensional space variables, and $T\subset R$ embeds the time-stamp variable, so that we can write $a'_i=(a_i,t_{a_i})$, where $a_i\in S$ and $t_{a_i}\in T$, with the condition that $t_{a_i}>t_{a_j}$ whenever $i>j$ (time stamp strictly increase in the sequence of samples).

A_i^j with $i\leq j$ is the sub time series consisting of the ith through the jth samples (inclusive) of A. Therefore, $A_i^j=a'_ia'_{i+1},\dots,a'_j$. $|A|$ denotes the length (the number of samples) of A. Λ denotes the null sample. A_i^j with $i>j$ is the null time series noted Ω.

An edit operation is a pair $(a',b')\neq(\Lambda,\Lambda)$ of time series samples, written as $a'\to b'$. Time series B results from the application of the edit operation $a\to b$ into time series A, written as $A\Rightarrow B$ via $a'\to b'$, if $A=\sigma a'\tau$ and $B=\sigma b'\tau$ for some time series σ and τ. We call $a'\to b'$ a match operation if $a'\neq\Lambda$ and $b'\neq\Lambda$, a delete operation if $b'=\Lambda$, and an insert operation if $a'=\Lambda$. Similarly, to the edit distance defined for string [18], [35], we define the similarity between any two time series A and B of finite length, respectively, p and q, as

$$\delta(A_1^p,B_1^q)=Min\begin{cases}\delta(A_1^{p-1},B_1^q)+\Gamma(a'_p\to\Lambda) & \mathit{delete},\\ \delta(A_1^{p-1},B_1^{q-1})+\Gamma(a'_p\to b'_q) & \mathit{match},\\ \delta(A_1^p,B_1^{q-1})+\Gamma(\Lambda\to b'_q) & \mathit{insert},\end{cases}$$

where $p\geq 1$, $q\geq 1$, and Γ is an arbitrary cost function, which assigns a nonnegative real number $\Gamma(a'_p\to b'_q)$ to each edit operation $a'_p\to b'_q$. The recursion is initialized by setting for $i>0$ and $j>0$:

$$\delta(A_1^0,B_1^0)=0,$$
$$\delta(A_1^0,B_1^j)=\infty,$$
$$\delta(A_1^i,B_1^0)=\infty.$$

DTW and ERPs and 1D-PPM and LCSS are special cases of the previous definitions that we present below.

3.2 The DTW Special Case

The DTW similarity measure δ_{dtw} [33], [31] is defined according to the previous notations as

$$\delta_{dtw}(A_1^p,B_1^q)=d_{LP}(a_p,b_q) + Min\begin{cases}\delta_{dtw}(A_1^{p-1},B_1^q),\\ \delta_{dtw}(A_1^{p-1},B_1^{q-1}),\\ \delta_{dtw}(A_1^p,B_1^{q-1}),\end{cases} \tag{2}$$

where $d_{LP}(a_p,b_q)$ is the Lp norm in R^d, and so, for DTW, $\Gamma(a'_p\to\Lambda)=\Gamma(a'_p\to b'_q)=\Gamma(\Lambda\to b'_q)=d_{LP}(a_p,b_q)$. One may note that the time-stamp values are not used; therefore, the costs of each edit operation involve vectors a and b in S instead of vectors a' and b' in $S\times T$. One of the main restrictions of δ_{dtw} is that it does not comply with the triangle inequality as shown by the following example [4]:

$$A_1^1=[1];B_1^2=[1,2];C_1^3=[1,2,2](*);$$
$$\delta_{dtw}(A_1^1,B_1^2)=1;\delta_{dtw}(B_1^2,C_1^3)=0;\delta_{dtw}(A_1^1,C_1^3)=2$$
$$\Rightarrow\delta_{dtw}(A_1^1,C_1^3)>\delta_{dtw}(A_1^1,B_1^2)+\delta_{dtw}(B_1^2,C_1^3).$$

(*) 1D time series with no stamp value given.

3.3 The ERP Special Case

$$\delta_{erp}(A_1^p, B_1^q) = Min\begin{cases} \delta_{erp}(A_1^{p-1}, B_1^q) + \Gamma(a'_p \rightarrow \Lambda), \\ \delta_{erp}(A_1^{p-1}, B_1^{q-1}) + \Gamma(a'_p \rightarrow b'_q), \\ \delta_{erp}(A_1^p, B_1^{q-1}) + \Gamma(\Lambda \rightarrow b'_q), \end{cases} \quad (3)$$

with

$$\Gamma(a'_p \rightarrow \Lambda) = d_{LP}(a_p, g),$$
$$\Gamma(a'_p \rightarrow b'_q) = d_{LP}(a_p, b_q),$$
$$\Gamma(\Lambda \rightarrow b'_q) = d_{LP}(g, b_q),$$

where g is a constant in S, and $d_{LP}(x, y)$ is the Lp norm of vector $(x - y)$ in S.

Note that the time stamp coordinate is not taken into account; therefore, δ_{erp} is a distance on S but not on $S \times T$. Thus, the cost of each edit operation involves vectors a and b in R^d instead of vectors a' and b' in R^{d+1}.

According to the authors of ERP [4], the constant g should be set to 0 for some intuitive geometric interpretation and in order to preserve the mean value of the transformed time series when adding gap samples.

3.4 The LCSS Special Case

The LCSS similarity measure has been first defined for string matching purposes [11] and then extended for times series [6], [34]. LCSS is recursively defined in [34] as follows:

$$LCSS_{\epsilon,\delta}(A_1^p, B_1^q) = $$
$$Min\begin{cases} 0 \text{ if } p < 1 \text{ or } q < 1, \\ 1 + LCSS_{\epsilon,\delta}(A_1^{p-1}, B_1^{q-1}) \text{ if } & \left[\begin{array}{l} d_{LP}(a_p, b_q) < \epsilon \text{ and} \\ |p - q| < \delta, \end{array}\right. \\ Max\begin{cases} LCSS_{\epsilon,\delta}(A_1^{p-1}, B_1^q) \\ LCSS_{\epsilon,\delta}(A_1^p, B_1^{q-1}) \end{cases} & \text{otherwise.} \end{cases} \quad (4)$$

For LCSS, the match reward is 1, while no reward is offered for *insert* or *delete* operations. The LCSS measure is transposed into a normalized dissimilarity measure $D_{\epsilon,\delta}$, which is close in its formal structure to the ERP measure:

$$D_{\epsilon,\delta} = 1 - \frac{LCSS_{\epsilon,\delta}(A_1^p, B_1^q)}{Min\{p, q\}}. \quad (5)$$

3.5 The 1D-PPM Special Case

For Point-Pattern matching problems [20], A_1^p and B_1^q are $1D$ ascending lists of real values:

$$\delta_{ppm}(A_1^p, B_1^q) = Min\begin{cases} \delta_{ppm}(A_1^{p-1}, B_1^q) + \Gamma(a'_p \rightarrow \Lambda), \\ \delta_{ppm}(A_1^{p-1}, B_1^{q-1}) + \Gamma(a'_p \rightarrow b'_q), \\ \delta_{ppm}(A_1^p, B_1^{q-1}) + \Gamma(\Lambda \rightarrow b'_q), \end{cases} \quad (6)$$

with

$$\Gamma(a'_p \rightarrow \Lambda) = a_p - a_{p-1},$$
$$\Gamma(a'_p \rightarrow b'_q) = |(a_p - a_{p-1}) - (b_q - b_{q-1})|,$$
$$\Gamma(\Lambda \rightarrow b'_q) = b_q - b_{q-1},$$

for $p > 1$ and $q > 1$.

Mäkinen [20] shows that δ_{ppm} is a metric that calculates the minimum amount of space needed to delete or insert between pairs of points to convert one point pattern into another. It can be noted that if successive increments are considered instead of the initial values, δ_{ppm} coincides with δ_{erp} when applied to the sequence of positive increments. Here again, time stamps are not used.

3.6 Symbolic Sequence Alignment with Affine Gap Penalty

In biomolecular sequences (DNA, RNA, or amino acid sequences), high sequence similarity usually implies significant functional or structural similarity. The basic mutational processes behind the evolution of such sequences are substitutions, insertions, and deletions, the latter two giving rise to gaps. Various similarity models based on dynamic programming have been developed by the bioinformatics community. Among them, the affine gap model [10], [8] that extends the Needleman-Wunsch algorithm [26] should be mentioned. The originality of this model is to penalized gap sequences according to the affine equation $\gamma(g) = -d - (g - 1) \cdot e$, where g is the length of the gap, d is the open-gap penalty, and e is the gap-extension penalty. The recursion is given in (8):

$$M(A_1^p, B_1^q) = \Gamma(a'_p \rightarrow b'_q) + Max\begin{cases} I_x(A_1^{p-1}, B_1^{q-1}), \\ M(A_1^{p-1}, B_1^{q-1}), \\ I_y(A_1^{p-1}, B_1^{q-1}), \end{cases} \quad (7)$$

$$I_x(A_1^p, B_1^q) = Max\begin{cases} M(A_1^{p-1}, B_1^q) - d, \\ I_x(A_1^{p-1}, B_1^q) - e, \end{cases}$$

$$I_y(A_1^p, B_1^q) = Max\begin{cases} M(A_1^p, B_1^{q-1}) - d, \\ I_y(A_1^p, B_1^{q-1}) - e. \end{cases}$$

Here, $\Gamma(a'_p \rightarrow b'_q) = \Gamma(a_p \rightarrow b_q)$ (since time stamps are not used) is an integer value either positive when a_p and b_p are similar symbols or negative when a_p and b_p are dissimilar symbols. $M(A_1^p, B_1^q)$ is the best score up to (a_p, b_p), given that a_p is aligned to b_q. $I_x(A_1^p, B_1^q)$ is the best score up to (a_p, b_p), given that a_p is aligned to a gap. Finally, $I_y(A_1^p, B_1^q)$ is the best score up to (a_p, b_p), given that b_p is aligned to a gap. The previous recursions are initialized as follows:

$$M(A_1^0, B_1^0) = 0, I_x(A_1^0, B_1^0) = I_x(A_1^0, B_1^0) = \infty$$
$$M(A_1^i, B_1^0) = I_x(A_1^i, B_1^0) = -d - (i - 1)e,$$
$$I_y(A_1^i, B_1^0) = -\infty, \text{ for } i = 1, \ldots, p$$
$$M(A_1^0, B_1^j) = I_y(A_1^0, B_1^j) = -d - (j - 1)e,$$
$$I_x(A_1^0, B_1^j) = -\infty, \text{ for } j = 1, \ldots, q.$$

4 The TWED Distance

We propose an alternative way of defining the edit operations for time series alignment, which leads to the definition of the new similarity measure TWED. To understand the semantic associated to the edit operations for TWED, we reconsider the editing analogy with strings and suggest some differences. The edit distance between two

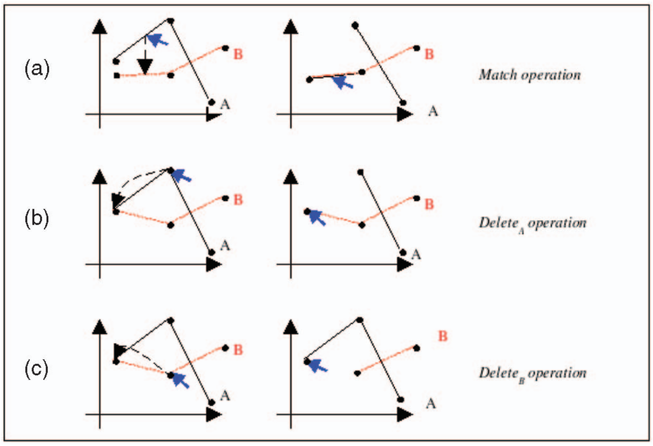

Fig. 1. The edit operations in the graphical editor paradigm.

strings is defined as the minimal transformation cost allowing for the transformation of the first string into the second one. For string edition, a transformation is a finite sequence of edit operations whose associated cost is the sum over the sequence of edit operations of the elementary costs Γ associated to each edit operation.

4.1 Graphical Editor Paradigm

For discrete time series, we are seeking a sequence of edit operations allowing for the simultaneous transformation of two time series to superimpose them with a minimal cost. If we use a graphical editor paradigm, we can imagine a 2D representation of time series for which the horizontal axis represents the time scale or the time-stamp coordinate, and the vertical axis represents a spatial coordinate scale displaying the projection of the d spatial coordinates of the samples onto a 1D scale. In this display, discrete time series are considered as a sequence of linear segments between successive samples. The graphical editor we have imagined allows for the editing of two time series A and B using three elementary edit operations depicted in Figs. 1a, 1b, and 1c.

Instead of the classical *delete*, *insert*, and *match* operations, we introduce $delete_A$, $delete_B$, and *match* operations as follows:

1. The $delete_A$ (deletion into the first time series) operation (Fig. 1b) consists of clicking on the dot that represents the sample in A to delete (a'_i) and of dragging and dropping this dot onto the previous sample dot (a'_{i-1}). We suggest that the editing effort or cost associated with this delete operation is proportional to the length of vector $(a'_i - a'_{i-1})$ to which we add a constant penalty $\lambda \geq 0$.
2. The $delete_B$ (deletion into the second time series) operation (Fig. 1c) consists of clicking on the dot that represents the sample in B to delete (b'_j) and of dragging and dropping this dot onto the previous sample dot (b'_{j-1}). Here again, we suggest that the editing effort or cost associated with this delete operation is proportional to the length of vector $(b'_j - b'_{j-1})$ to which we add a constant penalty $\lambda \geq 0$. Due to sampling rate variations or process variability, one could be faced with a situation where, in time series data, one event is either registered many times or only a few times when recording different utterances; this would justify that the deletion cost be proportional to the distance to the previous sample. Nevertheless, *outliers* (e.g., spurious data points) deletion cannot be covered by

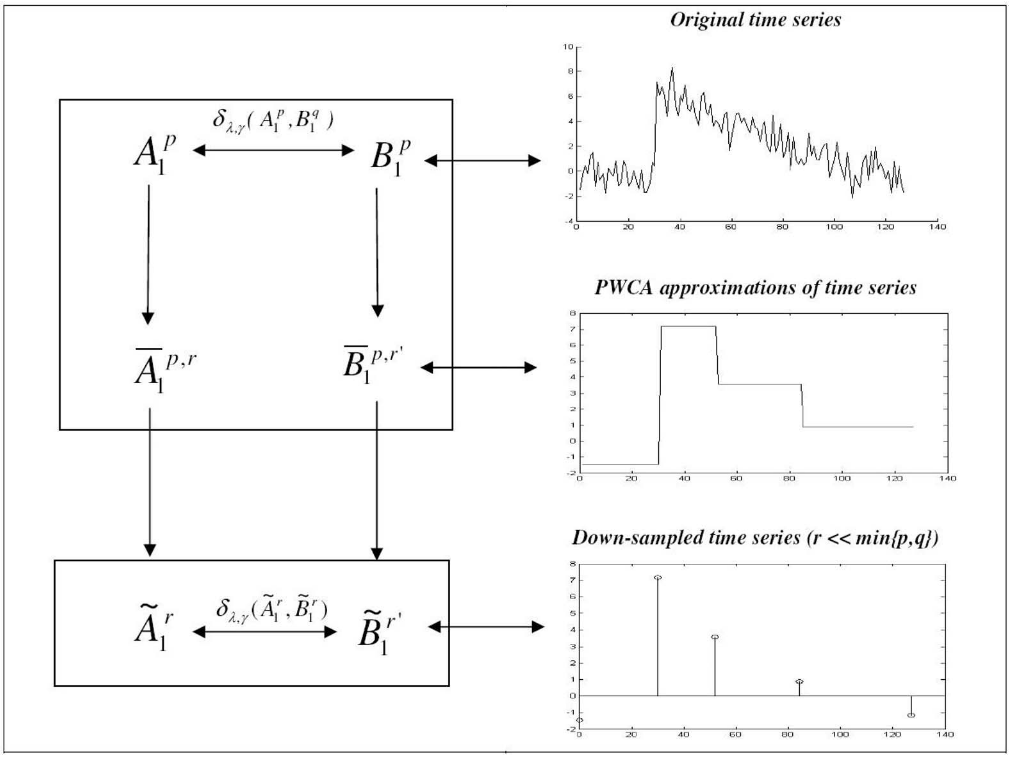

Fig. 2. Linking the matching of time series in the original space to the matching in the downsampled space.

this argument. According to TWED, the deletion cost for such sample depends on the previous sample in the time series, and there is no specific argument to justify it. The other elastic measures (DTW, ERP, LCSS) do not offer better justification for the deletion cost of outliers.

3. The *match* operation (Fig. 1a) consists of clicking on the segment to match in the first time series $(a'_i; a'_{i-1})$ and then of dragging and dropping this segment onto the graphic position corresponding to the matching segment $(b'_j; b'_{j-1})$ in the second time series. We can suggest that the editing effort or cost associated with the match operation is proportional to the sum of the lengths of the two vectors $(a'_i - b'_j)$ and $(a'_{i-1} - b'_{j-1})$.

This provides the basis for the TWED distance we propose.

4.2 Definition of TWED

$$\delta_{\lambda,\nu}(A_1^p, B_1^q) = \\ Min \begin{cases} \delta_{\lambda,\nu}(A_1^{p-1}, B_1^q) + \Gamma(a'_p \to \Lambda) & delete_A, \\ \delta_{\lambda,\nu}(A_1^{p-1}, B_1^{q-1}) + \Gamma(a'_p \to b'_q) & match, \\ \delta_{\lambda,\nu}(A_1^p, B_1^{q-1}) + \Gamma(\Lambda \to b'_q) & delete_B, \end{cases} \quad (8)$$

with

$$\begin{aligned} \Gamma(a'_p \to \Lambda) &= d(a'_p, a'_{p-1}) + \lambda, \\ \Gamma(a'_p \to b'_q) &= d(a'_p, b'_q) + d(a'_{p-1}, b'_{q-1}), \\ \Gamma(\Lambda \to b'_q) &= d(b'_q, b'_{q-1}) + \lambda. \end{aligned}$$

The recursion is initialized as follows:

$$\begin{aligned} &\delta_{\lambda,\nu}(A_1^0, B_1^0) = 0, \\ &\delta_{\lambda,\nu}(A_1^0, B_1^j) = \infty \text{ for } j \geq 1, \\ &\delta_{\lambda,\nu}(A_1^i, B_1^0) = \infty \text{ for } i \geq 1, \\ &\text{with } a'_0 = b'_0 = 0 \text{ by convention.} \end{aligned}$$

It is interesting to note that the penalties for $delete_A$ or $delete_B$ operations are similar to those proposed in the δ_{ppm} measure if we do not consider the time stamps coordinate and address the matching of 1D monotone increasing time series. Furthermore, using the graphical editor paradigm, we define the time series matching game as follows: Two time series, A and B, are displayed on the graphic. The goal is to edit A and B to completely superimpose the two curves. The editing process is performed from left to right: If i is an index on the segments of A and j on the segments of B, then the process initial setting is $i = j = 1$. A match operation will increment i and j simultaneously: $i \leftarrow i + 1$ and $j \leftarrow j + 1$. A $delete_A$ operation will increment i only: $i \leftarrow i + 1$. A $delete_B$ operation will increment j only: $j \leftarrow j + 1$. According to the above-mentioned constraint, once segment i in A has been processed using either a *match* or a $delete_A$ operation, it is impossible to edit it again: This rule applies for all previous segments r in $\{1, \ldots, i-1\}$. Similarly, once a segment j in B has been used either in a *match* or in a $delete_B$ operation, it is impossible to use former samples r in $\{1..j-1\}$ for future match or deletion operations. Therefore, according to this game, the editing process provides a sequence of edit operations, as well as ordered pairs of indices (i, j), where i is an index in the sequence of segments of A, and j is an index in the sequence of segments of B. In other words, the process provides an ordered sequence of triplets (op_k, i_k, j_k), where op_k is the kth edit operation selected, and i_k and j_k are the values of the index in A and B, respectively, when the edit operation is performed. A partial order can be defined on the triplets as follows: $(op_{k1}, i_{k1}, j_{k1}) < op_{k2}, i_{k2}, j_{k2})$ if and only if (iff) $i_{k1} \leq i_{k2}$ and $j_{k1} \leq j_{k2}$ and either $i_{k1} \neq i_{k2}$ or $j_{k1} \neq j_{k2}$. Since, for each step of the editing game, one of the indices is increased by one while the other is either incremented by one or remains unchanged, all the triplets in the output editing sequence are ordered in increasing order. Supposing that the game editing process has provided a sequence of edit operations up to i_k and j_k index values, if the sub-$\widehat{A}_1^{i_{k-1}}$ ($\widehat{A}_1^{i_{k-1}}$ refers to the sequence obtained from A after the first $k-1$ edit operations) and $\widehat{B}_1^{j_{k-1}}$ are not superimposed, then, as there is no possibility to process the former samples so that they may be superimposed, the game process cannot be successful. It is easy to show that $\delta_{\lambda,\nu}$, as defined in (8), provides a successful sequence of editing operations at a minimal global cost for all pairs of time series in U^2.

4.3 Some Properties of TWED

Proposition 4.1. *$\delta_{\lambda,\nu}$ is a distance on the set of finite discrete time series U:*

- *P1. $\delta_{\lambda,\nu}(A, B) \geq 0$ for any finite discrete time series A and B.*
- *P2. $\delta_{\lambda,\nu}(A, B) = 0$ iff $A = B$ for any finite discrete time series A and B.*
- *P3. $\delta_{\lambda,\nu}(A, B) = \delta_{\lambda,\nu}(B, A)$ for any finite discrete time series A and B.*
- *P4. $\delta_{\lambda,\nu}(A, B) \leq \delta_{\lambda,\nu}(A, C) + \delta_{\lambda,\nu}(C, B)$ for any finite discrete time series A, B, and C.*

The proof of Proposition 4.1 is given in [22]. The proofs for P1 and P3 are straightforward. P2 and P4 are proved inductively. The induction relies on the sum of the lengths of time series A, B, and C.

Proposition 4.2. *$\delta_{\lambda,\nu}$ is upper bounded by twice the Minkowski's distance D_{LP}. $\forall \lambda \geq 0, \forall X, Y \in U^2$, $\delta_{\lambda,\nu}(X, Y) \leq 2 \cdot D_{LP}(X, Y)$ whenever X and Y have the same length.*

The proof of Proposition 4.2 is given in [22]. Basically, if X and Y have the same length, the direct sequence of match operations (without any deletion or insertion) applying on samples pairs (x'_i, y'_i) has a global cost equal to twice the Minkowski's distance between X and Y. The result follows:

Proposition 4.3. *$\delta_{\lambda,\nu}$ is an increasing function of λ and ν:*

$$\forall \lambda \geq 0,\ \forall \nu > 0,\ \forall \lambda' \geq \lambda,\ \forall \nu' > \nu,\ \forall X,\ Y \in U^2 \\ \delta_{\lambda,\nu}(X, Y) \leq \delta_{\lambda',\nu'}(X, Y).$$

The proof, given in [22], is a straightforward consequence of the definition of $\delta_{\lambda,\nu}$.

4.4 Providing Stiffness into $\delta_{\lambda,\nu}$

Going back to the graphical editor game, we have envisaged that the penalty or cost associated with each edit operation should be proportional to the mouse pointer displacement involved during the edition. If we separate the spatial displacement in S from the temporal displacement in T, then we have to consider a spatial penalty that could be handled by a distance measured in S and a temporal penalty more or less proportional to some distance measured in T. By doing so, we could parameterize a distance in between the Minkowski's Distance, which is characterized by a kind of "*infinite stiffness*," and DTW, which is characterized by a "*null stiffness*."

In practice, we choose $d(a', b') = d_{LP}(a, b) + \nu \cdot d_{LP}(t_a, t_b)$, where ν is a non-negative constant, which characterizes the *stiffness* of the $\delta_{\lambda,\nu}$ elastic measures. Notice that $\nu > 0$ is required for $\delta_{\lambda,\nu}$ to be a distance. If $\nu = 0$, then $\delta_{\lambda,\nu}$ will be a distance on S but not on $S \times T$. The final formulation of $\delta_{\lambda,\nu}$ is shown as follows:

$$\delta_{\lambda,\nu}(A_1^p, B_1^q) = Min \begin{cases} \delta_{\lambda,\nu}(A_1^{p-1}, B_1^q) + \Gamma(a'_p \to \Lambda) & delete_A, \\ \delta_{\lambda,\nu}(A_1^{p-1}, B_1^{q-1}) + \Gamma(a'_p \to b'_q) & match, \\ \delta_{\lambda,\nu}(A_1^p, B_1^{q-1}) + \Gamma(\Lambda \to b'_q) & delete_B, \end{cases} \tag{9}$$

with

$$\begin{aligned} \Gamma(a'_p \to \Lambda) &= d_{LP}(a_p, a_{p-1}) + \nu \cdot (t_{a_p} - t_{a_{p-1}}) + \lambda, \\ \Gamma(a'_p \to b'_q) &= d_{LP}(a_p, b_q) + d_{LP}(a_{p-1}, b_{q-1}) \\ &\quad + \nu \cdot (|t_{a_p} - t_{b_q}| + |t_{a_{p-1}} - t_{b_{q-1}}|), \\ \Gamma(\Lambda \to b'_q) &= d_{LP}(b_q, b_{q-1}) + \nu \cdot (t_{b_q} - t_{b_{q-1}}) + \lambda. \end{aligned}$$

Some analogy can be found between the parameters λ and ν of $\delta_{\lambda,\nu}$ and parameters e and d of the affine model defined for symbolic sequence matching (see Section 3.6). Nevertheless, some major differences exist: The penalties in $\delta_{\lambda,\nu}$ are, for one part, proportional to the time-stamp difference between matching, deleted, or inserted samples. A constant penalty is added for the two deletion operations that correspond to gaps. Conversely, the affine model proposes a penalty proportional to the gap length corresponding to series of successive insertions or deletions, with a constant penalty for the first operation in the sequence of deletions.

The iterative implementation of $\delta_{\lambda,\nu}$ using the Lp metrics to evaluate the distance between two samples is depicted in Algorithm 1.

4.5 Algorithmic Complexity of $\delta_{\lambda,\nu}$

The time complexity of $\delta_{\lambda,\nu}$ is the same as DTW and ERP, namely, $O(p.q)$, where p and q are the lengths of the two time series being matched. The space complexity is also the same as DTW, i.e., $O(p.q)$, but as for the ERP distance, the costs $\Gamma(a'_p \to \Lambda)$ and $\Gamma(\Lambda \to b'_q)$ can be tabulated to speed up the calculation leading to an extra space complexity of $O(p+q)$ for $\delta_{\lambda,\nu}$.

Algorithm 1. Iterative implementation of TWED

```
float TWED(float A[1..n], float timeSA[1..n],
        float B[1..m], float timeSB[1..m],
        float λ, float ν) {
  DECLARE int DP[0..n,0..m];
  DECLARE int i, j;
  DECLARE float cost;
  DECLARE float A[0] := 0; float timeSA[0] = 0;
  DECLARE float B[0] := 0; float timeSB[0] = 0;
  FOR i := 1 to n
     DP[i, 0] := ∞;
  FOR j := 1 to m
     DP[0, j] := ∞;
  DP[0, 0] := 0;
  FOR i := 1 to n {
     FOR j := 1 to m {
        cost := D_LP(A[i], B[j]);
        DP[i − 1, j] := minimum (
           // Delete_A
           DP[i − 1, j] + D_LP(A[i − 1], A[i]) +
              ν · (timeSA[i] − timeSA[i − 1]) + λ,
           // Delete_B
           DP[i, j − 1] + D_LP(B[j − 1], B[j]) +
              ν · (timeSB[j] − timeSB[j − 1]) + λ,
           // match
           DP[i − 1, j − 1] + D_LP(A[i], B[j]) +
              D_LP(A[i − 1], B[j − 1]) +
              ν · ( = |timeSA[i] − timeSB[j]| +
                 | timeSA[i-1]-timeSB[j-1] | );
     } //END FOR j
  } //END FOR i
  Cost := DP[n, m];
  RETURN Cost;
}
```

4.6 Bounding the TWED Measure

In this section, we get back to our third motivation about defining a measure that exploits time stamps while verifying the triangle inequality. We show how piecewise constant approximations (PWCA) with few segments of time series can be used to improve the efficiency of range queries. Various methods exist to get polygonal curve approximations of time series, in particular, heuristic [7], [9], [14], [13], near optimal [17], [24], or optimal [29] solutions. Most of them can be adapted to provide PCWA approximation of time series.

We define $\overline{A}_1^{p,r}$ as a PWCA of time series A_1^p containing $r-1$ constant segments and p samples. This approximation can be obtained using any kind of solution (from heuristic to optimal solutions), say, the optimal solution similar to the one proposed in [29]. $\overline{A}_1^{p,r}$ and A_1^p have the same number of samples, namely, p. Let $\tilde{A}_1^r$ be the time series composed with the r segment extremities in $\overline{A}_1^{p,r}$. $\tilde{A}_1^r$ contains r samples. Let us similarly define $\overline{B}_1^{q,r'}$ and $\tilde{B}_1^{r'}$ from B_1^q time series.

Proposition 4.4. $\forall\lambda \geq 0$, $\forall\nu > 0$, $\forall X_1^p \in U$, $\forall r \in [1; p[$, *$\delta_{\lambda,\nu}(\overline{X}_1^{p,r}, \tilde{X}_1^r) \leq \lambda \cdot (p - r) + \nu \cdot \Delta T \cdot (2p - r)$, where ΔT is the time difference average between two successive samples inside the piecewise constant segments of the approximation.*

The proof of this proposition is given in [22]. The matching consisting of $(p-r)$ deletions in $\overline{X}_1^{p,r}$ and the matching of the r corresponding samples in $\overline{X}_1^{p,r}$ and $\tilde{X}_1^r$ has a global cost equal to $\lambda \cdot (p-r) + \nu \cdot \Delta T \cdot (2p-r)$. The result follows.

From Propositions 4.2, 4.3, 4.4, and the triangular inequality property, we get an upper bound for $|\delta_{\lambda,\nu}(A_1^p, B_1^q) - \delta_{\lambda,\nu}(\tilde{A}_1^r, \tilde{B}_1^{r'})|$ that quantifies the difference of the distance of two time series evaluated in the original space with the distance of their approximations evaluated in the downsampled space (see the Appendix for details):

$$\begin{aligned}|\delta_{\lambda,\nu}(A_1^p, B_1^q) - \delta_{\lambda,\nu}(\tilde{A}_1^r, \tilde{B}_1^{r'})| \leq \\ &\lambda \cdot (p+q-r-r') \\ &+ \nu \cdot \Delta T \cdot (2(p+q) - r - r') \\ &+ 2 \cdot D_{LP}(\overline{A}_1^{p,r}, A_1^p) + 2 \cdot D_{LP}(\overline{B}_1^{q,r'}, B_1^q).\end{aligned} \tag{10}$$

This shows that $\delta_{\lambda,\nu}(A_1^p, B_1^q)$ and $\delta_{\lambda,\nu}(\tilde{A}_1^r, \tilde{B}_1^{r'})$ are potentially close when two conditions are satisfied:

1. The PWCA approximations of A and B are close to the original time series in the sense of the LP-$distance$. This should be ensured by the optimal solution of the min-ϵ problem using piecewise constant segments whenever the number of segments r is not too small.
2. λ and $\nu \cdot \Delta T$ are small compared to $2 \cdot (p+q) - r - r'$.

Hence, we get the following lower bounds that can be considered tight if the two previous conditions are satisfied:

$$\begin{aligned}&\forall \lambda \geq 0, \forall \nu > 0, \forall \lambda' > \lambda, \forall \nu' > \nu, \forall A_1^p, B_1^q \in U^2, \\ &\forall r \in [1; p[, \forall r' \in [1; q[\\ &\quad \delta_{\lambda,\nu}(\tilde{A}_1^r, \tilde{B}_1^{r'}) - \Upsilon \leq \delta_{\lambda,\nu}(A_1^p, B_1^q) \\ &\quad\quad \leq \delta_{\lambda',\nu'}(A_1^p, B_1^q),\end{aligned} \tag{11}$$

where $\Upsilon = \lambda \cdot (p+q-r-r') - \nu \cdot \Delta T \cdot (2(p+q) - r - r') - 2 \cdot D_{LP}(\overline{A}_1^{p,r}, A_1^p) - 2 \cdot D_{LP}(\overline{B}_1^{q,r'}, B_1^q)$.

This last inequality is still potentially useful to design fast and dirty filters (FDFs) dedicated to range query searching, typically for applications for which λ' and ν' cannot be small enough, while λ and ν can be set up small. For range query search, if R is the radius of the range query and A_1^p is the center of the query ball, then B_1^q is outside the search range if one of the following conditions is verified:

$$\delta_{\lambda,\nu}(\tilde{A}_1^r, \tilde{B}_1^{r'}) > R + \Upsilon. \tag{12}$$

$$\delta_{\lambda,\nu}(\tilde{A}_1^r, \tilde{B}_1^{r'}) > R + \delta_{\lambda,\nu}(\tilde{A}_1^r, A_1^p) + \delta_{\lambda,\nu}(\tilde{B}_1^{r'}, B_1^q). \tag{13}$$

For time series information retrieval applications, inequalities (12) and (13) are potentially useful. If $\delta_{\lambda,\nu}(\tilde{A}_1^r, A_1^p)$ and $\delta_{\lambda,\nu}(\tilde{B}_1^{r'}, B_1^q)$ are precomputed during the indexing phase, the tighter bound (13) can be used. Otherwise, the second bound (12) can be evaluated during the retrieval phase through the computation of LP-$distances$.

Note that if $r = r' = (1/K) \cdot min(p,q)$, the complexity for evaluating $\delta_{\lambda,\nu}(\tilde{A}_1^r, \tilde{B}_1^{r'})$ is lower than $O(p.q/K^2)$.

5 Experimentations

5.1 Classification Experiment

To empirically evaluate the effectiveness of the TWED distance compared to other metrics or similarity measures, we address a simple classification task experiment. The classification task we have considered consists of assigning one of the possible categories to an unknown time series for the 20 data sets available at the UCR repository [16]. For each data set, a training subset is defined, as well as a testing subset. The classification is based on the simple nearest neighbor decision rule: First, we select a training data set containing time series for which the correct category is known. To assign a category to an unknown time series selected from a testing data set (different from the train set), we select its nearest neighbor (in the sense of a distance or similarity measure) within the training data set and then assign the associated category to its nearest neighbor.

Given a data set, we adapt the stiffness parameter as follows: We use the training data set to select the best *stiffness* (ν) value, as well as the best λ value, namely, the ones leading to the minimal error rate on the training data, according to a leave-one-out procedure (that consists of iteratively selecting one time series from the training set and then in considering it as a test against the remaining time series within the training set itself).

Finally, the testing data set is used to evaluate the final error rate (reported in Tables 1 and 2) with the best λ and ν values estimated on the training set. This leads to OTWED, the optimized versions of TWED. The same procedure is used to set up the parameters defined for the other parametric measures, i.e., optimized DTW (ODTW) and LCSS.

Tables 1 and 2 show the results obtained for the tested methods, e.g., ED on the original time series, ODTW with best warping windows, as defined in [30], classical DTW (DTW) with no warping window, LCSS, as defined in [11], ERP, as defined in [4] and OTWED. In Table 1 and Fig. 3, the time series are not preprocessed, while in Table 2 and Fig. 3, time series are downsampled using an optimal PCA procedure similar as the one described in [29] for polygonal approximation. In this last experiment, each down sampled time series has exactly 50 percent less samples than the original time series. The sampling rate for the down sampled time series is indeed varied since the size of the constant segments used to approximate the time series is not generally constant.

For parameterized measures, best values are selected from the training data in order to minimize the error rate estimated for the training data. More precisely, the settings are listed as follows:

- *ODTW*. The best corridor value is selected for each data set among the set $\{0, max(p,q)\}$ so as to minimize the classification errors estimated for the training data. If different corridor values lead to the minimal error rate estimated for the training data, then the lowest corridor value is selected.
- *LCSS*. The best δ and ϵ values are selected for each data set, respectively, among the sets $\{n, n/2, n/4, \ldots, n/2^k\}$, with $n = max(p,q)$ and $n/2^k \leq 1/2 \leq n/2^{k+1}$, and $\{20, 20/2, 20/4, \ldots, 20/2^k\}$, with $20/2^k \leq 1e^{-2} < 20/2^{k+1}$ so as to minimize the classification errors estimated for the training data. If different (δ, ϵ) values lead to the minimal error

TABLE 1
Comparative Study Using the UCR Data Sets VII: Classification Error Rate Obtained Using the First Near Neighbor Classification Rule for ED, DTW, ODTW, LCSS, ERP, and OTWED Distance

DATASET	#class|#train|#test	1NN ED	1NN ODTW	1NN DTW	1NN LCSS	1NN ERP	1NN OTWED
Synthetic control	6|300|300	0.12	0.017	**0.007**	0.047	0.036	0.023
Gun-Point	2|50|150	0.087	0.087	0.093	**0.013**	0.04	**0.013**
CBF	3|30|900	0.148	0.004	**0.003**	0.009	0.003	0.007
Face (all)	14|560|1690	0.286	0.192	0.192	0.201	0.202	**0.189**
OSU Leaf	6|200|242	0.483	0.384	0.409	**0.202**	0.397	0.248
Swedish Leaf	15|500|625	0.213	0.157	0.210	0.117	0.12	**0.102**
50 Words	50|450|455	0.369	0.242	0.310	0.213	0.281	**0.187**
Trace	4|100|100	0.24	0.010	**0.0**	0.020	0.170	0.050
Two Patterns	4|1000|4000	0.090	0.0015	**0.00**	**0.000**	**0.000**	0.001
Wafer	2|1000|6174	0.005	0.005	0.020	**0.000**	0.009	0.004
face (four)	4|24|88	0.216	0.114	0.170	0.068	0.102	**0.034**
Ligthing2	2|60|61	0.246	**0.131**	**0.131**	0.180	0.148	0.213
Ligthing7	7|70|73	0.425	0.288	0.274	0.452	0.301	**0.247**
ECG	2|100|100	0.120	0.120	0.230	**0.100**	0.130	**0.100**
Adiac	37|390|391	0.389	0.391	0.396	0.425	0.378	**0.376**
Yoga	2|300|300	0.170	0.155	0.164	0.137	0.147	**0.130**
Fish	7|175|175	0.267	0.233	0.267	0.091	0.120	**0.051**
Coffee	2|28|28	0.250	**0.179**	**0.179**	0.214	0.250	0.214
OliveOil	4|30|30	**0.133**	0.167	**0.133**	0.800	0.167	0.167
Beef	5|30|30	**0.467**	**0.467**	0.5	0.533	0.5	0.533

TABLE 2
Comparative Study Using the UCR Data Sets VII: Classification Error Rate Obtained Using the First Near Neighbor Classification Rule on Downsampled Time Series for ED, DTW, ODTW, LCSS, ERP, and OTWED Distance

DATASET	#class|#train|#test	1NN ED	1NN ODTW	1NN DTW	1NN LCSS	1NN ERP	1NN OTWED
Synthetic control	6|300|300	0.233	0.173	0.177	0.243	0.220	**0.000**
Gun-Point	2|50|150	0.140	0.113	0.067	0.027	0.047	**0.020**
CBF	3|30|900	0.240	0.027	**0.017**	0.030	0.028	0.067
Face (all)	14|560|1690	0.482	0.273	0.292	0.336	0.346	**0.236**
OSU Leaf	6|200|242	0.541	0.455	0.430	0.393	0.475	**0.281**
Swedish Leaf	15|500|625	0.932	0.323	0.322	0.288	0.291	**0.146**
50 Words	50|450|455	0.327	0.303	0.369	0.251	0.323	**0.189**
Trace	4|100|100	0.070	**0.000**	**0.000**	**0.000**	0.060	0.090
Two Patterns	4|1000|4000	0.593	**0.000**	**0.000**	0.104	0.013	0.001
Wafer	2|1000|6174	0.025	0.014	0.022	0.018	0.013	**0.010**
face (four)	4|24|88	0.432	0.239	0.216	0.295	0.261	**0.159**
Ligthing2	2|60|61	0.263	**0.098**	0.115	0.148	0.115	0.197
Ligthing7	7|70|73	0.521	0.315	0.342	0.427	**0.260**	0,370
ECG	2|100|100	0.180	0.250	0.260	0.320	0.200	**0.110**
Adiac	37|390|391	0.527	0.486	0.483	0.448	0.496	**0,417**
Yoga	2|300|300	0.204	00.166	0.171	0.188	0.189	**0.140**
Fish	7|175|175	0.371	0.354	0.354	0.189	0.280	**0.086**
Coffee	2|28|28	0.179	**0.143**	0.179	0.214	0.250	0.285
OliveOil	4|30|30	0.567	**0.167**	**0.167**	0.333	0.333	**0.167**
Beef	5|30|30	0.533	0.533	0.500	0.500	0.500	**0.333**

rate estimated for the training data, then the pairs having the highest δ value are selected first, and the pair with the highest ϵ value is finally selected.

- *OTWED*. For our experiment, the *stiffness* value (ν) is selected from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ and λ is selected from $\{0, .25, .5, .75, 1.0\}$. The ν and λ parameter values are selected for each data set so as to minimize the classification errors estimated on the training data. If different (ν, λ) values lead to the minimal error rate estimated for the training data, then the pairs containing the highest ν value are selected first, and the pair with the highest λ value is finally selected.

For the ERP and OTWED, we used the L1-norm, while the L2-norm has been implemented in DTW and ODTW, as reported in [30]. The gap value used in the ERP has been set equal to the distance between the deleted or inserted sample and 0, as suggested by Chen and Ng [4].

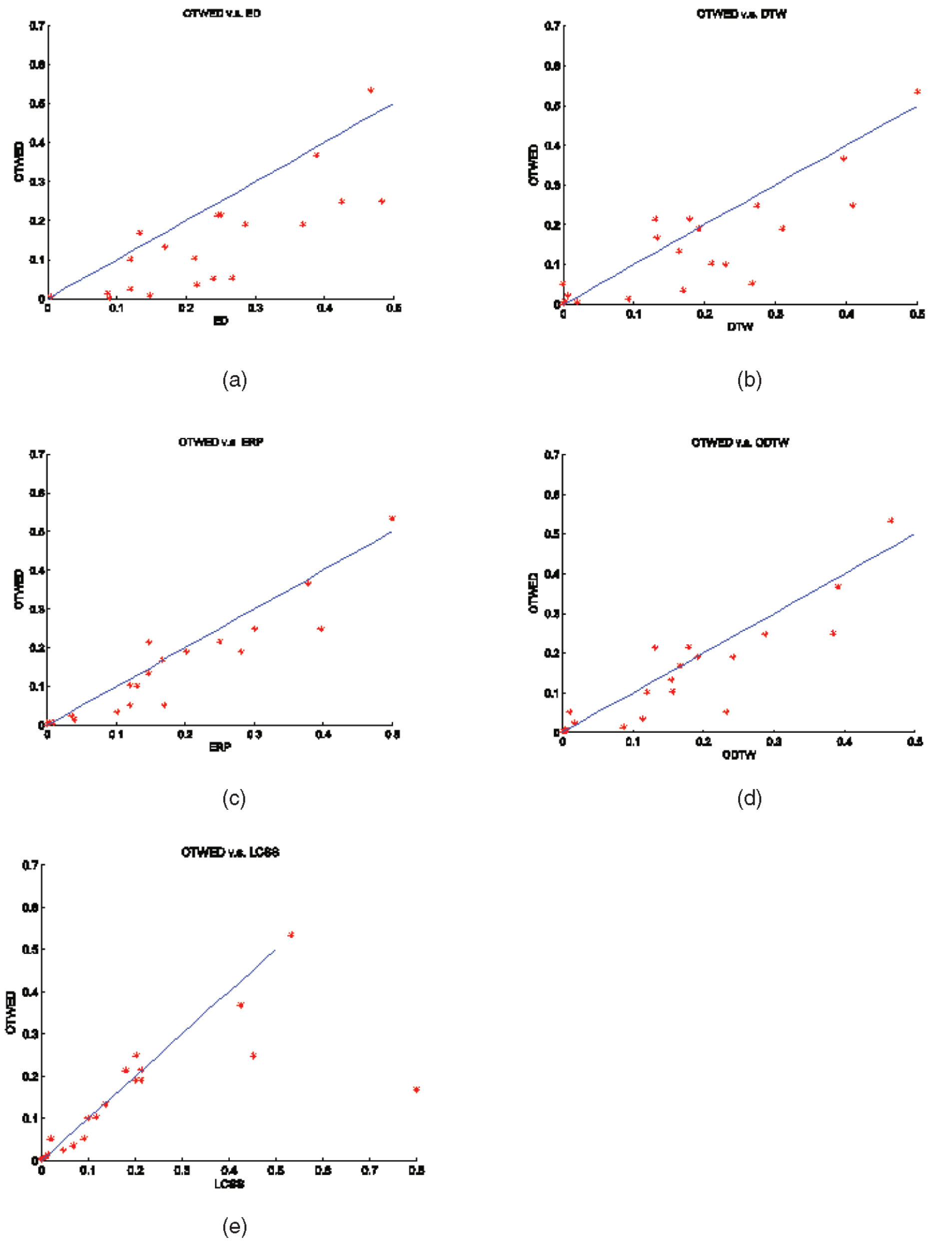

Fig. 3. Comparison of distance pairs (δ_x, δ_y). The x- and y-axes show the error rates for the two compared distances. The straight line has a slope of 1.0, and dots correspond to the error rate for the selected distance pair and tested data sets. A dot below (respectively, above) the straight line indicates that distance δ_y has a lower (respectively, higher) error rate than distance δ_x. (a) OTWED versus ED, (b) OTWED versus DTW, (c) OTWED versus ERP, (d) OTWED versus ODTW, and (e) OTWED versus LCSS.

Finally, as time is not explicitly given for these data sets, we used the index value of the samples as the time stamps for the whole experiment. This experiment shows that the TWED distance is effective for the considered task compared to ED, DTW, ODTW, ERP, and LCSS measures, since it exhibits, on the average, the lowest error rates for the testing data, as shown in Table 2 and Fig. 3. The gain, on the average, is relatively significant: around 2.5 percent against ODTW, 4.7 percent against LCSS, 3 percent against ERP, 9.4 percent against ED, and 8.8 percent against DTW.

The same experiment carried out for downsampled time series (Table 2 and Fig. 4) shows that the error rates drop more than twice as fast for ED, DTW, ERP, LCSS, and ODTW than for TWED. In that experimental context, using time stamps when matching nonuniform downsampled time series seems to be quite effective.

5.2 Range Query Experiment

Using the multiresolution approach defined in [24] to get nested PCWA approximations of time series in linear time complexity, the FDF that we propose iteratively evaluates the inequality (13) from the crudest level of resolution to the finest (which corresponds to the original time series). Between two successive levels of resolution, halves of the samples are eliminated. Given a radius R and a reference time series A, the FDF rejects candidate time series B as early as possible, i.e., as soon as two approximations of A and B satisfy inequality (13). The experiment consists in evaluating the processing time required to extract all the

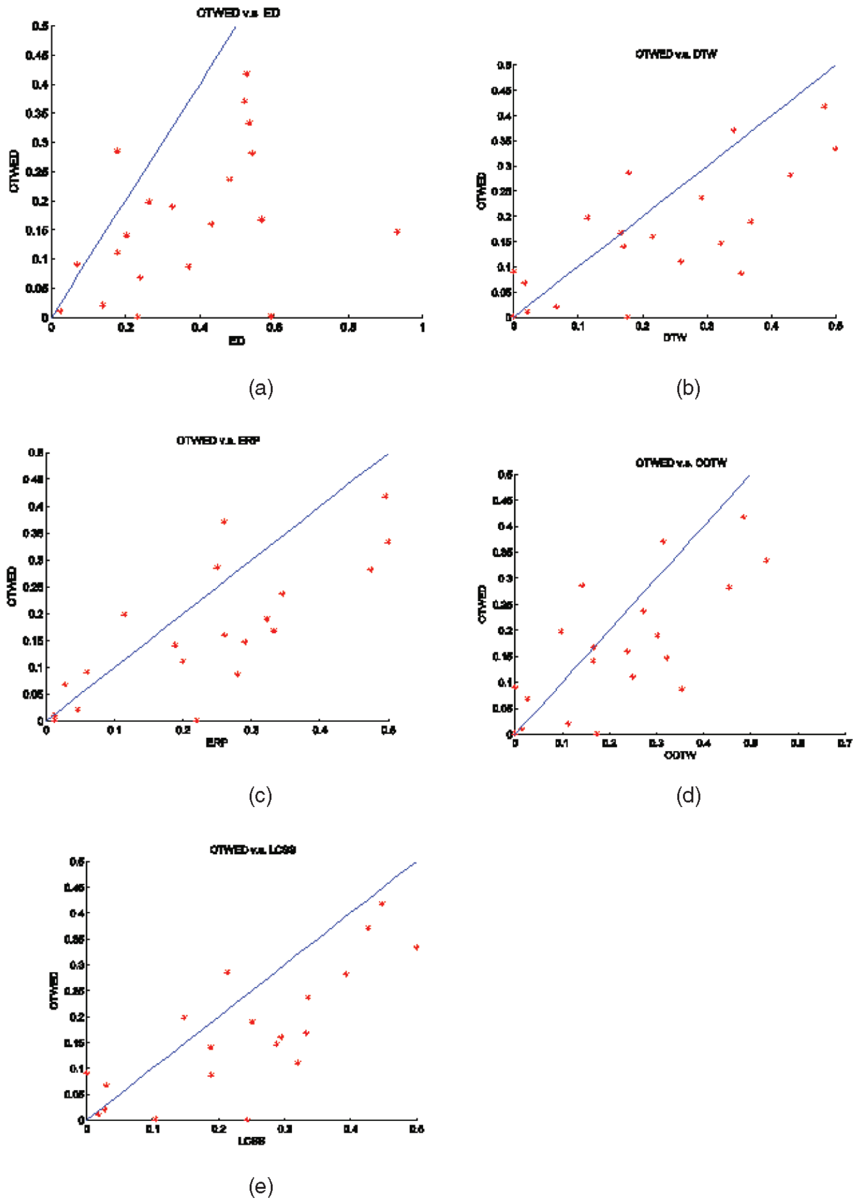

Fig. 4. Comparison of distance pairs (δ_x, δ_y) applied on downsampled times series. The x- and y-axes show the error rates for the two compared distances. The straight line has a slope of 1.0 and dots correspond to the error rate for the selected distance pair and tested data sets. A dot below (respectively, above) the straight line indicates that distance δ_y has a lower (respectively, higher) error rate than distance δ_x. Plot (a) shows OTWED versus ED, plot (b) shows OTWED versus DTW, plot (c) shows OTWED versus ERP, plot (d) shows OTWED versus ODTW, and plot (e) shows OTWED versus LCSS.

times series B located inside a ball of radius R centered on the reference time series A, which is drawn randomly from a database. We compare the FDF against a Linear Scanning (LS) procedure applied in the original time series space.

The first experiment is carried out from a heterogeneous database composed of the 20 data sets available at UCR [16]. This database comprises 23,999 time series. The second experiment is carried out from a homogeneous database, which is composed only of the Two_patterns data set available at UCR [16]. This database comprises 5,000 time series. For both experiments, parameters λ and ν are set constant, equal to the intermediate value 0.01. Fig. 5 shows that the FDF performs well for small radius for both databases. The FDF is by an order of magnitude faster than the LS procedure for radius varying from 1 to 4. The FDF matches LS for a radius R in between values 16 and 32. The FDF isperforming worse than LS for greater radius, mainly because inequalities (12) and (13) no longer efficiently apply.

6 Conclusion

From a graphical curve editing perspective and from earlier work on symbolic edit distance and DTW, we have developed an elastic similarity measure called TWED to match time series with some time shifting tolerance. We have proven that the TWED measure is a metric, and as such, TWED can be used complementarily with methods developed for searching in metric spaces as potential solutions for time series searching and retrieval applications

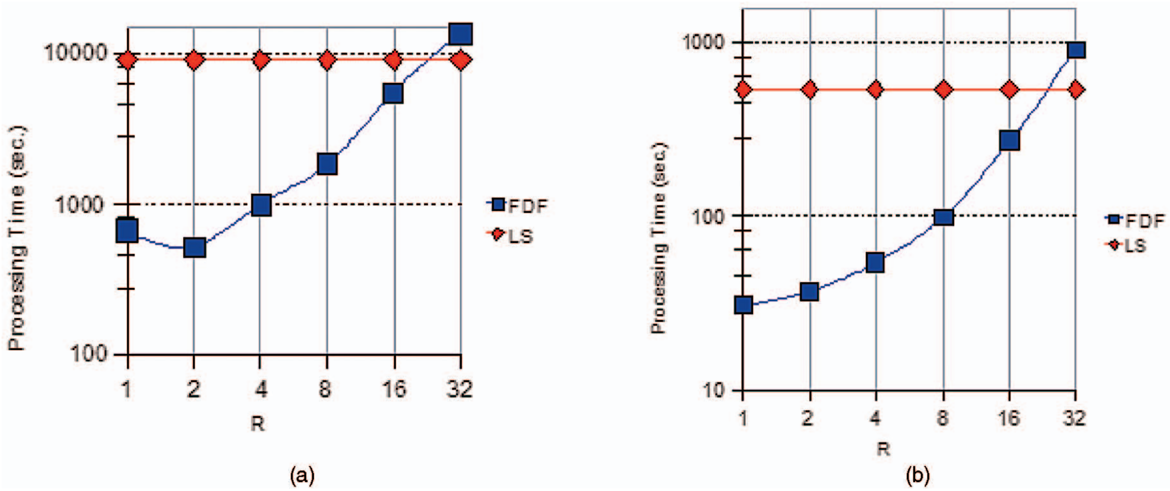

Fig. 5. Processing time (in seconds) required to filter 100 random queries as a function of the radius R for (a) the heterogeneous database and (b) the homogeneous database. The rhombuses constant line refers to the LS procedure, while the square line refers to the FDF procedure.

when time shift tolerance is concerned. The originality of TWED, compared to similar elastic measures, apart from the way insertions and deletions are managed, lies in the introduction of a parameter that controls the *stiffness* of the measure, thus placing TWED in between the EDs (infinite *stiffness*) and the DTW similarity measure (null *stiffness*). Moreover, TWED involved a second parameter that defines a constant penalty for insert or delete operations, similarly to the edit distance defined for string matching. These two parameters can be straightforwardly optimized for each application or data set as far as training data are available.

Furthermore, a procedure has been drawn up to lower bound the TWED metric. This procedure consists in approximating the time series using polygonal or piecewise constant approximations. It takes benefits from the triangle inequality to link the TWED measure evaluated on the approximated representations of time series to the TWED measure evaluated on the original time series. The computational cost reduction of TWED when evaluated in the approximated representation space is quadratic with the compression rate of the approximation. Nevertheless, this kind of lower bound has no linear complexity. Experimentation shows that one can expect to gain an order of magnitude in processing time using a very simple FDF based on this lower bound. The search for a lower bound whose complexity is effectively linear and that could be efficiently used in conjunction with downsampled approximation of time series is still a perspective.

The empirical quality of the distance has been evaluated through a classification experiment based on the first near neighbor classification rule for 20 different data sets. Globally, for this experiment, TWED performs, on the average, significantly better than the ED and Dynamic Time Warp measure and slightly better than the LCSS measure, the ERP, and the DTW measure with optimized search corridor size. When the classification experiment is applied to downsampled time series, TWED is more robust than the other tested measures. This is mainly because the time series are not uniformly sampled in this experiment, in which case, it is particularly relevant for time stamps.

Appendix

Upper Bound for the TWED Distance between a Time Series and Its DownSampled Approximation

From Propositions 4.2 to 4.4, we can upper-bound the matching of two time series evaluated in the original space with the matching of their approximations evaluated in the downsampled space as follows:

$$\forall \lambda \geq 0, \forall \nu > 0, \forall \lambda' > \lambda, \forall \nu' > \nu, \forall A_1^p, B_1^q \in U^2, \forall r \in [1; p[, \forall r' \in [1; q[$$

$$\begin{aligned}
&\delta_{\lambda,\nu}(\overline{A}_1^{p,r}, A_1^p) \leq 2 \cdot D_{LP}(\overline{A}_1^{p,r}, A_1^p) && \text{Prop. 4.2,}\\
&\delta_{\lambda,\nu}(\overline{A}_1^{p,r}, \tilde{A}_1^r) \leq \lambda \cdot (p - r) + \nu \cdot \Delta T \cdot (2p - r) && \text{Prop. 4.4,}\\
&\delta_{\lambda,\nu}(\overline{B}_1^{q,r'}, B_1^q) \leq 2 \cdot D_{LP}(\overline{B}_1^{q,r'}, B_1^q) && \text{Prop. 4.2,}\\
&\delta_{\lambda,\nu}(\overline{B}_1^{q,r'}, \tilde{B}_1^{r'}) \leq \lambda \cdot (q - r') + \nu \cdot \Delta T \cdot (2q - r') && \text{Prop. 4.4.}
\end{aligned}$$

From these inequalities and the triangle inequality verified by $\delta_{\lambda,\nu}$, we get

$$\delta_{\lambda,\nu}(\tilde{A}_1^r, A_1^p) \leq \lambda \cdot (p - r) + \nu \cdot \Delta T \cdot (2p - r) + 2 \cdot D_{LP}(\overline{A}_1^{p,r}, A_1^p),$$
$$\delta_{\lambda,\nu}(\tilde{B}_1^{r'}, B_1^q) \leq \lambda \cdot (q - r') + \nu \cdot \Delta T \cdot (2q - r') + 2 \cdot D_{LP}(\overline{B}_1^{q,r'}, B_1^q).$$

Moreover,

$$\begin{aligned}
\delta_{\lambda,\nu}(\tilde{A}_1^r, \tilde{B}_1^{r'}) &\leq \delta_{\lambda,\nu}(\tilde{A}_1^r, A_1^p) + \delta_{\lambda,\nu}(A_1^p, B_1^q) + \delta_{\lambda,\nu}(B_1^q, \tilde{B}_1^{r'})\\
&\leq \lambda \cdot (p - r) + \nu \cdot \Delta T \cdot (2p - r)\\
&\quad + \lambda \cdot (q - r') + \nu \cdot \Delta T \cdot (2q - r')\\
&\quad + 2 \cdot D_{LP}(\overline{A}_1^{p,r}, A_1^p) + 2 \cdot D_{LP}(\overline{B}_1^{q,r'}, B_1^q)\\
&\quad + \delta_{\lambda,\nu}(A_1^p, B_1^q),
\end{aligned}$$

$$\begin{aligned}
\delta_{\lambda,\nu}(A_1^p, B_1^q) &\leq \delta_{\lambda,\nu}(A_1^p, \tilde{A}_1^r) + \delta_{\lambda,\nu}(\tilde{A}_1^r, \tilde{B}_1^{r'}) + \delta_{\lambda,\nu}(\tilde{B}_1^{r'}, B_1^q)\\
&\leq \lambda \cdot (p - r) + \nu \cdot \Delta T \cdot (2p - r)\\
&\quad + \lambda \cdot (q - r') + \nu \cdot \Delta T \cdot (2q - r')\\
&\quad + 2 \cdot D_{LP}(\overline{A}_1^{p,r}, A_1^p) + 2 \cdot D_{LP}(\overline{B}_1^{q,r'}, B_1^q)\\
&\quad + \delta_{\lambda,\nu}(\tilde{A}_1^r, \tilde{B}_1^{r'}).
\end{aligned}$$

Leading to

$$|\delta_{\lambda,\nu}(A_1^p, B_1^q) - \delta_{\lambda,\nu}(\tilde{A}_1^r, \tilde{B}_1^{r'})| \leq \lambda \cdot (p + q - r - r') + \nu \cdot \Delta T \cdot (2(p + q - r - r') + 2 \cdot D_{LP}(\overline{A}_1^{p,r}, A_1^p) + 2 \cdot D_{LP}(\overline{B}_1^{q,r'}, B_1^q).$$

Acknowledgments

This work was partially supported by the French *Ministére de l'Enseignement Superieur et de la Recherche*.

References

[1] R. Agrawal and R. Srikant, "Mining Sequential Patterns," *Proc. 11th IEEE Int'l Conf. Data Eng.*, pp. 3-14, 1995.

[2] R. Bellman, *Dynamic Programming.* Princeton Univ. Press, 1957.

[3] E. Chàvez, G. Navarro, R.A. Baeza-Yates, and J.L. Marroquin, "Searching in Metric Spaces," *ACM Computing Surveys,* vol. 33, no. 3, pp. 273-321, 2001.

[4] L. Chen and R. Ng, "On the Marriage of LP-Norm and Edit Distance," *Proc. Int'l Conf. Very Large Data Bases,* pp. 792-801, 2004.

[5] L. Chen, M.T. Ozsu, and V. Oria, "Robust and Fast Similarity Search for Moving Object Trajectories," *Proc. Int'l Conf. Management of Data (SIGMOD '05),* pp. 491-502, 2005.

[6] G. Das, D. Gunopulos, and H. Mannila, "Finding Similar Time Series," *Proc. Conf. Principles of Knowledge Discovery and Data Mining,* pp. 454-456, 1997.

[7] D.H. Douglas and T.K. Peucker, "Algorithm for the Reduction of the Number of Points Required to Represent a Line or Its Caricature," *The Canadian Cartographer,* vol. 10, no. 2, pp. 112-122, 1973.

[8] R. Durbin, S. Eddy, A. Krogh, and G. Mitchinson, *Biological Sequence Analysis—Probabilistic Models of Proteins and Nucleic Acids.* Cambridge Univ. Press, 1998.

[9] E. Fink and B. Pratt, "Indexing of Compressed Time Series," *Data Mining in Time Series Databases,* pp. 43-65, World Scientific, 2004.

[10] O. Gotoh, "An Improved Algorithm for Matching Biological Sequences," *J. Molecular Biology,* vol. 162, pp. 705-708, 1982.

[11] D.S. Hirschberg, "A Linear Space Algorithm for Computing Maximal Common Subsequences," *Comm. ACM,* vol. 18, no. 6, pp. 341-343, 1975.

[12] H. Imai and M. Iri, "Polygonal Approximations of a Curve—Formulations and Algorithms," *Computational Morphology,* pp. 71-86, 1988.

[13] E.J. Keogh, K. Chakrabarti, M.J. Pazzani, and S. Mehrotra, "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases," *J. Knowledge and Information Systems,* vol. 3, pp. 263-286, 2001.

[14] E.J. Keogh and M.J. Pazzani, "A Simple Dimensionality Reduction Technique for Fast Similarity Search in Large Time Series Databases," *Proc. Fourth Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD '00),* pp. 122-133, 2000.

[15] E.J. Keogh, L. Wei, X. Xi, S.-H. Lee, and M. Vlachos, "Lb_keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures," *Proc. ACM Int'l Conf. Very Large Databases,* pp. 882-893, 2006.

[16] E.J. Keogh, X. Xi, L. Wei, and C.A. Ratanamahatana, *The UCR Time Series Classification-Clustering Datasets,* http://wwwcs.ucr.edu/eamonn/time_series_data/, 2006.

[17] A. Kolesnikov and P. Fränti, "Reduced-Search Dynamic Programming for Approximation of Polygonal Curves," *Pattern Recognition Letters,* vol. 24, pp. 2243-2254, 2003.

[18] V.I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," *Doklady Akademii Nauk SSSR,* vol. 163, no. 4, pp. 707-710, 1966 (in Russian), english translation in Soviet Physics Doklady, vol. 10, no. 8, pp. 845-848, 1965.

[19] J. Lin, E.J. Keogh, and A.F. van Herie, "Approximations to Magic: Finding Unusual Medical Time Series," *Proc. 18th IEEE Int'l Symp. Computer-Based Medical Systems,* pp. 329-334, 2005.

[20] V. Mäkinen, "Using Edit Distance in Point-Pattern Matching," *Proc. Eighth Symp. String Processing and Information Retrieval (SPIRE '01),* pp. 153-161, 2001.

[21] V. Mäkinen, "Parameterized Approximate String Matching and Local-Similarity-Based Point-Pattern Matching," PhD dissertation, Report A-2003, Dept. of Computer Science, Univ. of Helsinki, 2003.

[22] P.F. Marteau, "Time Warp Edit Distance," Technical Report valoria/UBS-2008-3v, VALORIA, Univ. de Bretagne Sud, http://hal.archives-ouvertes.fr/hal-00258669/fr/, 2008.

[23] P.F. Marteau and S. Gibet, "Adaptive Sampling of Motion Trajectories for Discrete Task-Based Analysis and Synthesis of Gesture," *Proc. Int'l Gesture Workshop,* pp. 224-235, 2005.

[24] P.F. Marteau and G. Ménier, "Speeding up Simplification of Polygonal Curves Using Nested Approximations," *J. Pattern Analysis and Application,* pp. 1-8, Springer, June 2008, http://dx.doi.org.10.1007/s10044-008-0133-y.

[25] P. Moore and D. Molloy, "A Survey of Computer-Based Deformable Models," *Proc. Int'l Machine Vision and Image Processing Conf. (IMVIP '07),* pp. 55-64, 2007.

[26] S.B. Needleman and C.D. Wunsch, "A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins," *J. Molecular Biology,* vol. 48, pp. 443-453, 1970.

[27] M.K. Ng and Z. Huang, *Temporal Data Mining with a Case Study as Astronomical Data Analysis,* LNCS, pp. 2-18, 1997.

[28] S. Park, J.I. Won, J.H. Yoonc, and S.W. Kimb, "A Multi-Dimensional Indexing Approach for Timestamped Event Sequence Matching," *Information Sciences,* vol. 177, pp. 4859-4876, 2007.

[29] J.C. Perez and E. Vidal, "Optimum Polygonal Approximation of Digitized Curves," *Pattern Recognition Letters,* vol. 15, pp. 743-750, 1994.

[30] C.A. Ratanamahatana and E.J. Keogh, "Making Time-Series Classification More Accurate Using Learned Constraints," *Proc. Fourth SIAM Int'l Conf. Data Mining (SDM '04),* pp. 11-22, 2004.

[31] H. Sakoe and S. Chiba, "A Dynamic Programming Approach to Continuous Speech Recognition," *Proc. Seventh Int'l Congress of Acoustic,* pp. 65-68, 1971.

[32] A. Singh, D. Terzopoulos, and D.B. Goldgof, *Deformable Models in Medical Image Analysis.* IEEE CS Press, 1998.

[33] V.M. Velichko and N.G. Zagoruyko, "Automatic Recognition of 200 Words," *Int'l J. Man-Machine Studies,* vol. 2, pp. 223-234, 1970.

[34] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E.J. Keogh, "Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures," *Proc. ACM Special Interest Group Knowledge Discovery and Data Mining (SIGKDD '03),* pp. 216-225, 2003.

[35] R.A. Wagner and M.J. Fischer, "The String-to-String Correction Problem," *J. ACM,* vol. 21, pp. 168-173, 1973.

[36] H. Wang, C. Perng, W. Fan, S. Park, and P. Yu, "Indexing Weighted Sequences in Large Databases," *Proc. IEEE Int'l Conf. Data Eng.,* pp. 63-74, 2003.

[37] H. Wu, B. Salzberg, and D. Zhang, "Online Event-Driven Subsequence Matching over Financial Data Streams," *Proc. Int'l Conf. Management of Data (SIGMOD '04),* pp. 23-34, 2004.

Pierre-François Marteau received the master of engineering degree in computer science from the Ecole Nationale Supérieure d'Electronique et d'Informatique de Bordeaux in 1984, and the PhD degree in computer science from the Institut National Polytechnique de Grenoble (INPG) in 1988. He held a postdoctorate position at the Institute for Nonlinear Sciences, University of California, San Diego, and then worked for eight years as an IT consultant in a private company in Paris before joining the Computer Science Laboratory, Université de Bretagne Sud, in 1999, where he is a professor. He is a member of the IEEE Computer Society.