

# Scaling and Time Warping in Time Series Querying

Ada Wai-chee Fu<sup>1</sup> Eamonn Keogh<sup>2</sup> Leo Yung Hang Lau<sup>1</sup> Chotirat Ann Ratanamahatana<sup>2</sup>

Department of Computer Science and Engineering<sup>1</sup>  
The Chinese University of Hong Kong  
Shatin, Hong Kong  
adafu,yhlau@cse.cuhk.edu.hk

Department of Computer Science and Engineering<sup>2</sup>  
University of California  
Riverside, USA  
eamonn,ratana@cs.ucr.edu

## Abstract

The last few years have seen an increasing understanding that Dynamic Time Warping (DTW), a technique that allows local flexibility in aligning time series, is superior to the ubiquitous Euclidean Distance for time series classification, clustering, and indexing. More recently, it has been shown that for some problems, Uniform Scaling (US), a technique that allows global scaling of time series, may just be as important for some problems. In this work, we note that for many real world problems, it is necessary to combine both DTW and US to achieve meaningful results. This is particularly true in domains where we must account for the natural variability of human action, including biometrics, query by humming, motion-capture/animation, and handwriting recognition. We introduce the first technique which can handle both DTW and US simultaneously, and demonstrate its utility and effectiveness on a wide range of problems in industry, medicine, and entertainment.

## 1 Introduction

We propose to query time series with both the accommodation of a scaling factor and the consideration of time warping effects. In this section we justify our proposal with some examples.

---

*Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Data Base Endowment. To copy otherwise, or to republish, requires a fee and/or special permission from the Endowment.*

**Proceedings of the 31st VLDB Conference,  
Trondheim, Norway, 2005**

## 1.1 Justifying the Need for Uniform Scaling and DTW

Because time series are near ubiquitous, and are becoming increasingly prevalent as our ability to capture and store them improves, there is increasing interest in the database community in techniques for efficiently indexing large time series collections [7, 19]. It is found that in most domains, it is necessary to match sequences with tolerance of small local misalignments, and Dynamic Time Warping has been shown to be the right tool for this [5, 12, 21, 24]. For example, in speech comparison, small fluctuation of the tempo of the speakers should be allowed in order to identify similar contents. More recently, it has been shown that in many domains it is also necessary to match sequences with the allowance of a global scaling factor [14]. In this work, we argue that for most real world problems, it is necessary to be able to handle both types of distortions simultaneously. In fact, even a casual glance of existing literature confirms this. For example, in query-by-humming systems, it is well understood that we must allow for uniform scaling in addition to DTW. The current solution is to simply do DTW at many resolutions that span the possible range of tempos. For example, Meek and Birmingham [17] reports “*We account for the phenomenon of persons reproducing the same tune at different speeds ... allow(ing) for nine tempo mappings.*” However, repeating the query nine times clearly slows the system down. Furthermore, it is possible that the best match occurs somewhere in-between the nine discrete scalings. In [15], the authors also note that in addition to the local problems handled by DTW, “*(people can) perform faster or slower than usual.*” They again deal with this with multiple scaled queries, achieving reasonable performance only by the use of parallel processing.

Dynamic Time Warping is frequently used as the basis of gait recognition algorithms, but even in this highly structured domain, it is recognized that uniform scaling is also needed. For example, [11] notes “*different gait cycles tend to have unequal lengths.*” In fact, even if we discount human variability, it is well

understood that parallax effects from cameras (static or pan-and-tilt) can produce apparent changes in uniform scaling [10].

The need for uniform scaling has been noted in bioinformatics. Moeller-Levet et al. [18] noted that previous work that addressed *only* local scaling (with DTW) is inadequate, and they stressed that “(uniform) scaling factors in the expression level hide similar expressions and have to be eliminated or not considered when assessing the similarity between expression profiles” [11].

Finally, the simultaneous need for both uniform scaling and DTW is well understood in the motion-capture and animation community. For example, Pullen and Bregler [20], explaining their motion-capture editing system, noted “we stretch or compress the real data fragments in time by linearly resampling them to make them the same length as the keyframed fragment . . . (then do DTW).” The computational difficulty of dealing with both uniform scaling and DTW at the same time has even led to practitioners abandoning temporal information altogether! Campbell and Bobick [3] used a phase space representation in which the velocity dimensions are removed, thus completely disregarding the time component of the data. This makes the learning and matching of motion patterns simpler and faster, but only at the cost of a massive increase in false positives.

Let us call “DTW with Uniform Scaling” **SWM**, which stands for *Scaled and Warped Matching*. In this paper, we study the combined effects of scaling and time warping in time series querying.

## 1.2 Motivating Examples

Below, we present two concrete examples that require SWM to produce meaningful and intuitive results.

**Example 1 (Indexing video).** *There is increasing interest in indexing sports data, both from sports fans who may wish to find particular types of shots or moves, and from coaches who are interested in analyzing their athletes performance over time. As a concrete example, we consider high jump. We can automatically collect the athlete’s center of mass information from video and convert the data into a time series (It is possible to correct for the cameras pan and tilt; see [6]). We found that when we issued queries to a database of high jumps, we got intuitive answers only when doing SWM. It is easy to see why if we look at two particular examples from the same athlete and consider all possible matching options, as shown in Figure 1. In this figure, we show four different ways to match two time series, the horizontal axis is the time axis. In each case, we have shifted one of the two series upward to show the way the points in the two series are matched. Each vertical line in the diagrams shows the*

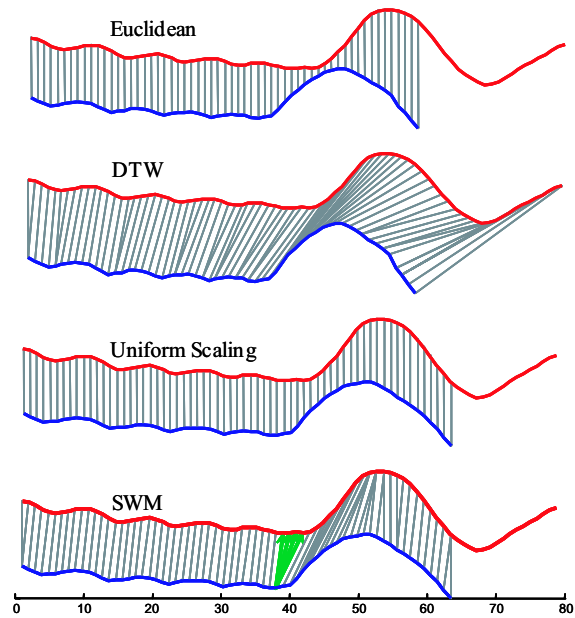


Figure 1: Two examples of an athlete’s trajectories aligned with various measures

*matching of two points. Visually we can say that the two time series are similar, and hence the distance between them should be small. We want to see which of the four measurement can generate a result that gives a small distance as expected. From top to bottom:*

- If we attempt simple Euclidean matching (after truncating the longer sequence), we get a large distance (which we can consider as error) because we are mapping part of the *flight* of one sequence to the *takeoff drive* in the other.
- If we simply use DTW to match the entire sequences, we get a large error because we are trying to explain part of the sequence in one attempt (the bounce from the mat) that simply does not exist in the other sequence. This problem can be corrected by constraints such as the Sakoe-Chiba Band, but without scaling, the matching will be poor.
- If we attempt just uniform scaling, we get the best match when we stretch the shorter sequence by 112%. However the local alignment, particularly of the *takeoff drive* and *up-flight*, is quite poor.
- Finally, when we match the two sequences with SWM, we get an intuitive alignment between the two sequences. The global stretching (once again at 112%) allows DTW to align the small local differences. In this case, the fact that DTW needed to map a single point in a time series onto 4 points in the other time series suggests an important local difference in one of these sequences. Inspection of the original videos by a professional coach

suggests that the athlete misjudged his approach and attempted a clumsy correction just before his takeoff drive.

**Example 2 (Query by Humming).** *The need for both local and global alignment when working with music has been extensively demonstrated [4, 16, 17, 24]. For completeness, we will briefly review it here. Finding similar sequences of music has applications in copyright infringement detection, analyzing the evolution of music styles [4], automatic annotation, etc. (It is interesting to note that these studies are not confined to human endeavors; similar techniques have been used in animal “music”, especially in humpback whales and songbirds [16]). However, the vast majority of research in this area is used to support query by humming.*

The basic idea of query by humming is to allow users to search large music collections by providing an example of the desired content, by humming (or singing, or tapping) a snippet. Clearly, humans cannot be expected to reproduce an exact fragment of a song, so the system must be invariant to certain distortions. Some of these are trivial to deal with. For example, the query can be made invariant to key by normalizing both the query and the database to a standard key. However, two types of errors are more difficult to deal with; users may perform the query at the wrong tempo, and users may insert or delete notes. The former corresponds with uniform scaling, the latter with DTW. The music retrieval community has traditionally dealt with these two problems in two ways. The first is to do DTW multiple times, at different scalings [17]. However, this clearly produces scalability problems. The other common approach is to only do DTW with relatively short song snippets as queries believing that short sequences are less sensitive to uniform scaling problems than long sequences. While this is undoubtedly true, short snippets also have less discriminating power.

In Figure 2, we demonstrate the problems with the universally familiar piece of music, *Happy Birthday to You*. For clarity of illustration, the music was produced by the fourth author on a keyboard and converted into a pitch contour, however, similar remarks apply to other music representations. From top to bottom:

- Because the query sequence was performed at a much faster tempo, direct application of DTW fails to produce an intuitive alignment.
- Rescaling the shorter performance by a scaling factor of 1.54 seems to improve the alignment, but note for example that the higher pitched note produced on the third “*birth...*” of the candidate is forced to align with the lower note of the third “*happy...*” in the query.

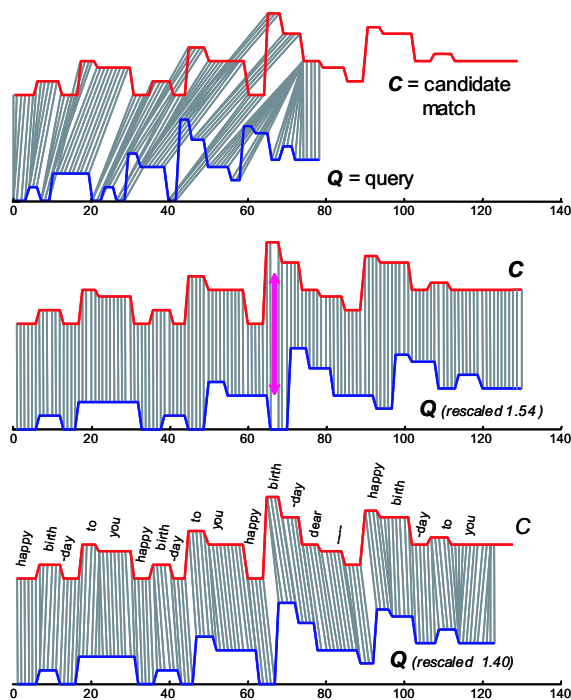


Figure 2: Two performances of *Happy Birthday to You* aligned with different metrics. Both performances were performed in the same key, but are shifted in the Y-axis for visual clarity.

- Only the application of *both* uniform scaling and DTW produces the correct alignment.

Having developed the intuition for DTW and US, and having demonstrated the need to handle both types of distortions simultaneously, we will next define the problem of similarity measurement under SWM more formally.

## 2 Problem Definition

Assume that we are given a database  $D$  which contains  $M$  time series (note that this formulation does not preclude the subsequence matching case, since it may be trivially transformed into this formulation). Further, assume that we are given a query  $Q$ , and a scaling factor  $l, l \geq 1$ , which represents the maximum allowable stretching of the time series. The maximum allowable shrinking is implicitly set to  $1/l$ .<sup>1</sup> Hence the query can be shortened by a factor of up to  $1/l$  or lengthen by a factor of up to  $l$ . Note that while  $1/l$  is bounded below by zero, and  $l$  is bounded from above by infinity, such loose bounds would allow pathological solutions to certain problems, and in any case are surely impossible to support efficiently. We therefore

<sup>1</sup>Such formulation assumes the maximum allowable stretching and shrinking is symmetric. If this is not the case for a specific application, it is trivial to add as an extra parameter: the maximum allowable shrinking  $s$ .

restrict our attention to scaling factors in the range  $0.5 \leq 1/l \leq 1 \leq l \leq 2$ . Note that this range encompasses the necessary flexibility documented in virtually every domain we are aware of. For instance, in [17], the authors reported excellent results with a query-by-humming system that allows “a (maximum) tempo scaling of 1.25.” [17] notes that in their experience, amateur singers can speed up their rendition of a song by as much as 200% or slow down to as little as 50%.

Recently it has been shown that for nearly all types of time series data, using an appropriate global constraints always improves the classification or clustering accuracy and the precision and recall of indexing [21]. Therefore a global constraint is typically enforced to limit the warping path to a roughly *diagonal portion* of the warping matrix.

Given  $N$  variable-length data sequences and a query sequence  $Q$ , we would like to find all data sequences that are “similar” to  $Q$ . Suppose the query sequence is  $Q = Q_1, Q_2, \dots, Q_m$ , where  $Q_i$  is a numerical value. We are interested in tackling the following problem.

**Problem:** Assume the data sequences can be longer than the query sequence  $Q$ . Find the best match to  $Q$  in database, for any rescaling in a given range, under the Dynamic Time Warping distance with a global constraint. By best match we mean the data sequence with the smallest distance from  $Q$ .

This problem has never been considered in the literature before. This problem is realistic in applications such as query by humming.

### 3 Preliminaries

In this section, we review separately time series querying with time warping distance and also querying with the scaling effect. For each case, we can apply a lower bounding technique for pruning the search space.

#### 3.1 Time Warping Distance

Intuitively, *dynamic time warping* is a distance measure that allows time series to be *locally* stretched or shrunk before applying the base distance measure. Definition 1 formally defines time warping distance.

**Definition 1 (Time Warping Distance (DTW)).** Given two sequences  $C = C_1, C_2, \dots, C_n$  and  $Q = Q_1, Q_2, \dots, Q_m$ , the time warping distance DTW is defined recursively as follows:

$$\begin{aligned} \text{DTW}(\phi, \phi) &= 0 \\ \text{DTW}(C, \phi) &= \text{DTW}(\phi, Q) = \infty \\ \text{DTW}(C, Q) &= D_{\text{base}}(\text{First}(C), \text{First}(Q)) + \\ &\min \begin{cases} \text{DTW}(C, \text{Rest}(Q)) \\ \text{DTW}(\text{Rest}(C), Q) \\ \text{DTW}(\text{Rest}(C), \text{Rest}(Q)) \end{cases} \end{aligned}$$

where  $\phi$  is the empty sequence,  $\text{First}(C) = C_1$ ,  $\text{Rest}(C) = C_2, C_3, \dots, C_n$ , and  $D_{\text{base}}$  denotes the distance between two entries.

Several metrics were used as the  $D_{\text{base}}$  distance in previous literature, such as Manhattan Distance [23] and squared Euclidean Distance [12, 22]. We will use squared Euclidean Distance as the  $D_{\text{base}}$  measure. That is,

$$D_{\text{base}}(C_i, Q_j) = (C_i - Q_j)^2$$

Note we deliberately omit the final square root function in our distance definitions. Such optimization speeds up computations without altering the relative ranking given by these distances, which is more important than the actual value in most applications. The same optimization has been used before in [14]. However, if such optimization is not desired, we can also consistently insert the final square root function without altering the essence of this work.

It is well known that dynamic time warping distance can be computed by filling a *warping matrix* using a *dynamic programming algorithm* directly derived from the definition of time warping distance. A *warping path* can be identified by tracing the elements in the warping matrix that were used to compute the time warping distance. Formally, a warping path  $W$  for two sequences  $Q$  and  $C$  is a sequence of elements  $w_1, w_2, \dots, w_p$  so that  $w_k = (i_k, j_k)$  is an entry in the warping matrix, where  $i_k \geq i_{k-1}$  and  $j_k \geq j_{k-1}$ ,  $\max(|Q|, |C|) \leq |W| \leq |Q| + |C| - 1$ .<sup>2</sup>

#### 3.2 Constraints and Lower Bounding

In the previous section we have explained with examples the importance of having global constraints on time warping in order to give meaningful results. Keogh [12] suggested a lower bounding measure based on such *global constraints* on time warping. Two commonly used global constraints exist. The *Sakoe-Chiba Band* [22] limits the warping path to a band enclosed by two straight lines that are parallel to the diagonal of the warping matrix. The *Itakura Parallelogram* [9] limits the warping path to be within a parallelogram whose major diagonal is the diagonal of the warping matrix.

[12] viewed a global constraint as a constraint on the warping path entry  $w_k = (i, j)_k$  and gave a general form of global constraints in terms of inequalities on the indices to the elements in the warping matrix,

$$j - r \leq i \leq j + r$$

where  $r$  is a constant for the Sakoe-Chiba Band and  $r$  is a function of  $i$  for the Itakura Parallelogram.

<sup>2</sup> $|X|$  denotes the length of a sequence  $X$ .

Incorporating the global constraint into the definition of dynamic time warping distance, Definition 1 can be modified as follows.

**Definition 2 (Constrained DTW (cDTW)).** Given two sequences  $C = C_1, C_2, \dots, C_n$  and  $Q = Q_1, Q_2, \dots, Q_m$ , and the time warping constraint  $r$ , the constrained time warping distance cDTW is defined recursively as follows:

$$\text{Dist}_r(C_i, Q_j) = \begin{cases} D_{\text{base}}(C_i, Q_j) & \text{if } |i - j| \leq r \\ \infty & \text{otherwise} \end{cases}$$

$$\text{cDTW}(\phi, \phi, r) = 0$$

$$\text{cDTW}(C, \phi, r) = \text{cDTW}(\phi, Q, r) = \infty$$

$$\text{cDTW}(C, Q, r) = \text{Dist}_r(\text{First}(C), \text{First}(Q)) + \min \begin{cases} \text{cDTW}(C, \text{Rest}(Q), r) \\ \text{cDTW}(\text{Rest}(C), Q, r) \\ \text{cDTW}(\text{Rest}(C), \text{Rest}(Q), r) \end{cases}$$

where  $\phi$  is the empty sequence,  $\text{First}(C) = C_1$ ,  $\text{Rest}(C) = C_2, C_3, \dots, C_n$ , and  $D_{\text{base}}$  denotes the distance between two entries.

The upper bounding sequence  $UW$  and the lower bounding sequence  $LW$  of a sequence  $C$  are defined using the time warping constraint  $r$  as follows.

**Definition 3 (Enveloping Sequences by Keogh [12]).** Let  $UW = UW_1, UW_2, \dots, UW_m$  and  $LW = LW_1, LW_2, \dots, LW_m$ ,

$$UW_i = \max(C_{i-r}, \dots, C_{i+r}) \text{ and}$$

$$LW_i = \min(C_{i-r}, \dots, C_{i+r})$$

Considering the boundary cases, the above can be rewritten as

$$UW_i = \max(C_{\max(1, i-r)}, \dots, C_{\min(i+r, n)}) \text{ and}$$

$$LW_i = \min(C_{\max(1, i-r)}, \dots, C_{\min(i+r, n)})$$

These two sequences form an envelope which encloses the sequence  $C$ , as shown in Figure 3.

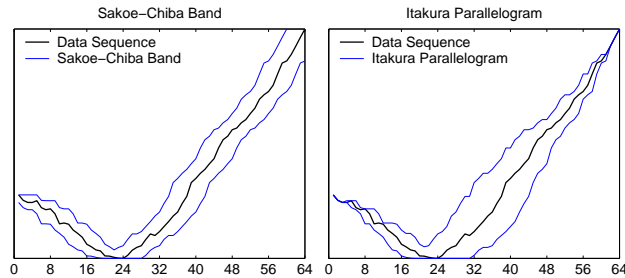


Figure 3: Enveloping sequences derived from two different constraints

The lower bounding measure by Keogh [12] bounds the time warping distance between two sequences  $Q$  and  $C$  by the Euclidean distance between  $Q$  and the envelope of  $C$ . Equation (1) below formally defines the lower bounding distance.

$$LB_W(Q, C) = \sum_{i=1}^m \begin{cases} (Q_i - UW_i)^2 & \text{if } Q_i > UW_i \\ (Q_i - LW_i)^2 & \text{if } Q_i < LW_i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

### 3.3 Uniform Scaling

Consider a query sequence  $Q = Q_1, \dots, Q_m$  and a candidate sequence  $C = C_1, \dots, C_n$ .

We assume that  $m$  is not greater than  $n$  ( $m \leq n$ ); hence, the query is typically shorter than the candidate sequence. We assume that the data can scale up or down by a factor of at most  $l$ ,  $l \geq 1$ . The entry  $Q_m$  may be matched to  $C_{lm}$  when the data is shrunk by a factor of  $l$ . To simplify our discussion we shall assume that  $lm \leq n$ .

In order to scale time series  $C = C_1, \dots, C_q$  to produce a new time series  $C' = C'_1, \dots, C'_m$  of length  $m$ , we use the formula

$$C'_j = C_{\lceil j \cdot q/m \rceil} \text{ where } 1 \leq j \leq m$$

This is similar to the formula used in [14]. We target to find a scaled prefix in  $C$  to compare with  $Q$ . With a scaling factor of  $l$ ,  $q$  can range from  $\lceil m/l \rceil$  to  $lm$ .

**Definition 4 (Uniform Scaling (US)).** Given two sequences  $Q = Q_1, \dots, Q_m$  and  $C = C_1, \dots, C_n$  and a scaling factor bound  $l$ ,  $l \geq 1$ . Let  $C(q)$  be the prefix of  $C$  of length  $q$ , where  $\lceil m/l \rceil \leq q \leq lm$  and  $C(m, q)$  be a rescaled version of  $C(q)$  of length  $m$ ,

$$C(m, q)_i = C(q)_{\lceil i \cdot q/m \rceil} \text{ where } 1 \leq i \leq m$$

$$\text{US}(C, Q, l) = \min_{q=\lceil m/l \rceil}^{\min(lm, n)} D(C(m, q), Q)$$

where  $D(X, Y)$  denotes the Euclidean distance between two sequences  $X$  and  $Y$ .

Note that the ceiling function in the definition of  $C(p, q)$  may be replaced by the floor function. The whole definition of  $C(p, q)$  may also be replaced by some interpolation on the values of  $C(q)_{\lfloor i \cdot q/p \rfloor}$  and  $C(q)_{\lceil i \cdot q/p \rceil}$ .

#### 3.3.1 Lower bounding uniform scaling

We create two sequences  $UC = UC_1, \dots, UC_m$  and  $LC = LC_1, \dots, LC_m$ , such that

$$UC_i = \max(C_{\lceil i/l \rceil}, \dots, C_{\lceil i \rceil})$$

$$LC_i = \min(C_{\lfloor i/l \rfloor}, \dots, C_{\lfloor i \rfloor})$$

These sequences bound the points of the time series  $C$  that can be matched with  $Q$ .

The lower bounding function, which lower bounds the distance between  $Q$  and  $C$  for any scaling  $\rho$ ,  $1 \leq \rho \leq l$ , can now be defined as:

$$LB_S(Q, C) = \sum_{i=1}^m \begin{cases} (Q_i - UC_i)^2 & \text{if } Q_i > UC_i \\ (Q_i - LC_i)^2 & \text{if } Q_i < LC_i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

**Lemma 1.** *For any two sequences  $Q$  and  $C$  of length  $m$  and  $n$  respectively, for any scaling constraint on the warping path  $w_k = (i, j)_k$  of the form  $j/l \leq i \leq lj$ , the value of  $LB_S(Q, C)$  lower bounds the distance between  $C$  and  $Q$  under a scaling of  $C$  between  $1/l$  and  $l$ , where  $l \geq 1$ .*

*Proof.* We can assume a matching path  $w_k = (i, j)_k$  which defines a mapping between the indices  $i$  and  $j$ , so that each such mapping constitutes a term  $(Q_i - C_j)^2$  to the required distance. We will show that each term  $t_{lb}$  in the square root of our lower bounding distance  $LB_S(Q, C)$  can be matched with a term  $t$  resulted from the one-to-one mapping, with  $t_{lb} \leq t$ .

Based on the constraints on the scaling factor, we have the constraint  $j/l \leq i \leq lj$  between  $i$  and  $j$  in  $w_k = (i, j)_k$ . From this, we have  $i/l \leq j \leq il$  and by definition

$$\begin{aligned} UC_i &= \max(C_{\lceil i/l \rceil}, \dots, C_{\lceil il \rceil}) \\ LC_i &= \min(C_{\lceil i/l \rceil}, \dots, C_{\lceil il \rceil}) \end{aligned}$$

thus  $UC_i = \max(C_{\lceil i/l \rceil}, \dots, C_j, \dots, C_{\lceil il \rceil}) \geq C_j$ , or

$$Q_i - UC_i \leq Q_i - C_j$$

If  $Q_i > UC_i$  then  $Q_i - UC_i > 0$ , hence

$$(Q_i - UC_i)^2 \leq (Q_i - C_j)^2$$

Similarly we can show that if  $Q_i < LC_i$  then

$$(Q_i - LC_i)^2 \leq (Q_i - C_j)^2 \quad \square$$

## 4 Scaling and Time Warping

Having reviewed time warping, uniform scaling, and lower bounding, this section introduces *scaling and time warping* (SWM).

**Definition 5 (Scaling and Time Warping (SWM)).** *Given two sequences  $Q = Q_1, \dots, Q_m$  and  $C = C_1, \dots, C_n$ , a bound on the scaling factor  $l, l \geq 1$  and the Sakoe-Chiba Band time warping constraint  $r$  which applies to sequence length  $m$ . Let  $C(q)$  be the*

*prefix of  $C$  of length  $q$ , where  $\lceil m/l \rceil \leq q \leq \min(lm, n)$  and  $C(m, q)$  be a rescaled version of  $C(q)$  of length  $m$ ,*

$$\begin{aligned} C(m, q)_i &= C(q)_{\lceil i \cdot q/m \rceil} \text{ where } 1 \leq i \leq m \\ \text{SWM}(C, Q, l, r) &= \min_{q=\lceil m/l \rceil}^{\min(lm, n)} \text{cDTW}(C(m, q), Q, r) \end{aligned}$$

To simplify our discussion we shall assume that  $lm \leq n$ . We are interested in being able to scale the sequence and also to find nearest neighbor or evaluate range query by means of time warping distance. As noted in [14], a naïve search for the uniform scaling problem alone requires  $O(|D| \cdot (a - b))$  time, where  $[b, a]$  is the range of lengths resulting from scaling. Time warping computation alone requires  $O(n^2)$  time for time series length of  $n$ . Hence we need to find a more efficient technique for the SWM problem.

In previous sections, we reviewed the lower bounding technique for each sub-problem. Here, we propose to combine these lower bounds to form overall lower bounds for the querying problem. Figure 4 illustrates this graphically.<sup>3</sup>

We apply time warping on top of scaling, i.e. we scale the sequence first, and then measure the time warping distance of the scaled sequence with the query. Typically, time warping with Sakoe-Chiba Band constrains the warping path by a fraction of the data length, which is translated into a constant  $r$ . Hence, if the fraction is 10%, then  $r = 0.1|C|$ . If the length of  $C$  is changed according to the scaling fraction  $\rho$ , that is,  $C$  is changed to  $\rho C$ , then the Sakoe-Chiba Band time warping constraint is  $r = 0.1|\rho C|$ . Hence, we have  $r = r'\rho$ , where  $r'$  is the Sakoe-Chiba Band time warping constraint on the unscaled sequence, and  $\rho$  is the scaling factor.

The lower envelope  $L_i$  and upper envelope  $U_i$  on  $C$  can be deduced as follows: Recall that the upper and lower bounds for uniform scaling between  $1/l$  and  $l$  is given by the following:

$$\begin{aligned} UC_i &= \max(C_{\lceil i/l \rceil}, \dots, C_{\lceil il \rceil}) \\ LC_i &= \min(C_{\lceil i/l \rceil}, \dots, C_{\lceil il \rceil}) \end{aligned}$$

and the upper and lower bounds for a Sakoe-Chiba Band time warping constraint factor of  $r$  for a point  $C_i$  is given by:

$$\begin{aligned} UW_i &= \max(C_{\max(1, i-r)}, \dots, C_{\min(i+r, n)}) \\ LW_i &= \min(C_{\max(1, i-r)}, \dots, C_{\min(i+r, n)}) \end{aligned}$$

Therefore, when we apply time warping on top of

<sup>3</sup>In this example, the scaling factor is  $l = 1.5$ , the time warping constraint is  $r' = 10\%$  of the length of  $C$ .



scaling the upper and lower bounds will be:

$$\begin{aligned}
 U_i &= \max(UW_{\lceil i/l \rceil}, \dots, UW_{\lceil il \rceil}) \\
 &= \max(C_{\max(1, \lceil i/l \rceil - r')}, \dots, C_{\min(\lceil i/l \rceil + r', n)}, \dots, \\
 &\quad C_{\max(1, \lceil il \rceil - r')}, \dots, C_{\min(\lceil il \rceil + r', n)}) \\
 &= \max(C_{\max(1, \lceil i/l \rceil - r')}, \dots, C_{\min(\lceil il \rceil + r', n)}) \quad (3)
 \end{aligned}$$

$$\begin{aligned}
 L_i &= \min(LW_{\lceil i/l \rceil}, \dots, LW_{\lceil il \rceil}) \\
 &= \min(C_{\max(1, \lceil i/l \rceil - r')}, \dots, C_{\min(\lceil i/l \rceil + r', n)}, \dots, \\
 &\quad C_{\max(1, \lceil il \rceil - r')}, \dots, C_{\min(\lceil il \rceil + r', n)}) \\
 &= \min(C_{\max(1, \lceil i/l \rceil - r')}, \dots, C_{\min(\lceil il \rceil + r', n)}) \quad (4)
 \end{aligned}$$

The lower bound function which lower bounds the distance between  $Q$  and  $C$  for any scaling in the range of  $\{1/l, l\}$  and time warping with the Sakoe-Chiba Band constraint factor of  $r'$  on  $C$  is given by:

$$LB(Q, C) = \sum_{i=1}^m \begin{cases} (Q_i - U_i)^2 & \text{if } Q_i > U_i \\ (Q_i - L_i)^2 & \text{if } Q_i < L_i \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

**Lemma 2.** For any two sequences  $Q$  and  $C$  of length  $m$  and  $n$  respectively, given a scaling constraint of  $\{1/l, l\}$  (see Section 2 on problem definition), where  $l \geq 1$ , and a Sakoe-Chiba Band time warping constraint of  $r'$  on the original (unscaled) sequence  $C$ , the value of  $LB(Q, C)$  lower bounds the distance of  $SWM(C, Q, l, r')$ .

*Proof.* The matching warping path  $w_k = (i, j)_k$  defines a mapping between the indices  $i$  and  $j$ . Each such mapping constitutes a term  $t = (Q_i - C_j)^2$  to the required distance. We will show that the  $i$ -th term  $t_{lb}$  in our lower bounding distance  $LB(Q, C)$  can be matched with the term  $t$  resulting in a one-to-one mapping, with  $t_{lb} \leq t$ . For the  $i$ -th term  $t_{lb}$ , if  $Q_i > U_i$ , then  $t_{lb} = (Q_i - U_i)^2$ ; if  $Q_i < L_i$ , then  $t_{lb} = (Q_i - L_i)^2$ , otherwise  $t_{lb} = 0$ , which is always  $\leq t$ .

For scaling plus time warping, as illustrated in Figure 5, the effective constraint on the range of  $j$  is given by:  $\lceil i/l \rceil - r' \leq j \leq \lceil il \rceil + r'$

By Equations 3 and 4

$$U_i = \max(C_{\max(1, \lceil i/l \rceil - r')}, \dots, C_{\min(\lceil il \rceil + r', n)})$$

$$L_i = \min(C_{\max(1, \lceil i/l \rceil - r')}, \dots, C_{\min(\lceil il \rceil + r', n)})$$

$$\text{thus } U_i \geq C_j, \text{ or } Q_i - U_i \leq Q_i - C_j$$

Hence if  $(Q_i > U_i)$  then  $Q_i - U_i > 0$  and we have

$$(Q_i - U_i)^2 \leq (Q_i - C_j)^2$$

Similarly we can show that when  $(Q_i < L_i)$

$$(Q_i - L_i)^2 \leq (Q_i - C_j)^2 \quad \square$$

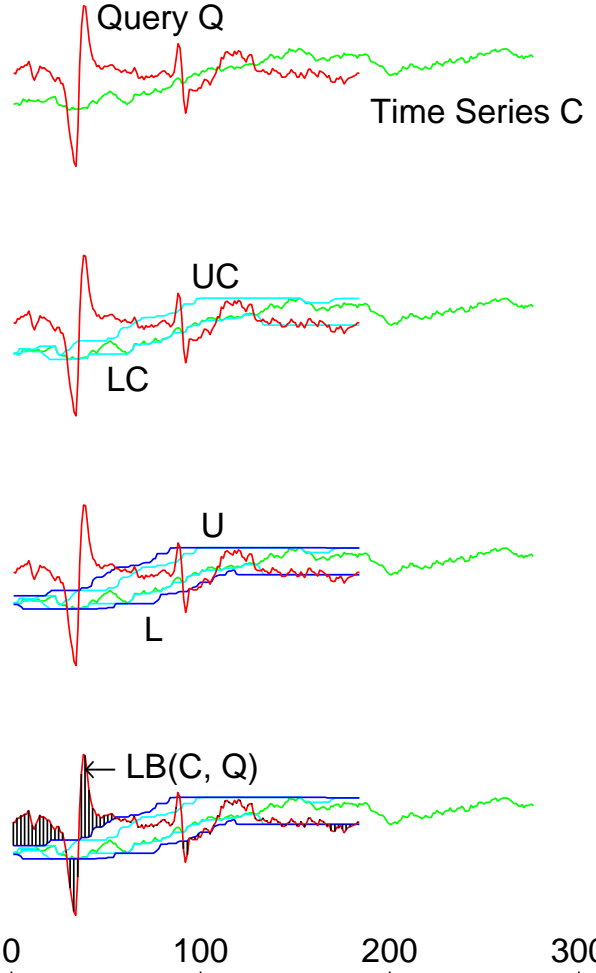


Figure 4: An illustration of the SWM envelopes. From top to bottom: A time series  $C$  and a query  $Q$ ; The series  $C$  bounded from above and below respectively by  $UC$  and  $LC$ , the envelope for scaling; The series  $UC$  bounded above by  $U$  and  $LC$  bounded below by  $L$ , forming the overall envelope for scaling and time warping; and the lower bounding distance  $LB$  derived from the overall envelope.

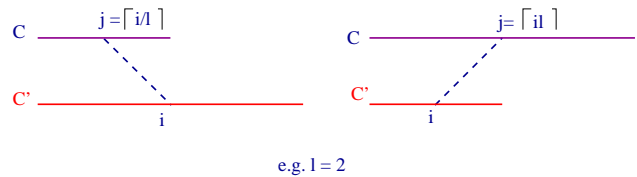


Figure 5: An illustration of the scaling effect, given a sequence  $C$ ,  $C'$  is the result after scaling. Note that the Sakoe-Chiba Band time warping constraint  $r'$  applies to  $C$ . Hence the range of  $j$  is given by  $[\lceil i/l \rceil - r', \lceil il \rceil + r']$ .

## 5 Tightness of the lower bounds

In this section, we show that the lower bounds we have described are tight. In general, to show that a lower bound is tight, we need only find a case where the exact distance is equal to the lower bound distance. However this is not exactly applicable in our scenario. For the lower bounds  $LB_W(Q, C)$ ,  $LB_S(Q, C)$ , and  $LB(Q, C)$  we have discussed so far, the formulae have a similar pattern:

$$LB(Q, C) = \sum_{i=1}^m \begin{cases} (Q_i - U_i)^2 & \text{if } Q_i > U_i \\ (Q_i - L_i)^2 & \text{if } Q_i < L_i \\ 0 & \text{otherwise} \end{cases}$$

For each point in the query sequence, we have a lower envelope value e.g.  $L_i$  and an upper envelope value, e.g.  $U_i$ , so that the sequence  $Q$  can compare in order to calculate the lower bounds. The values of  $L_i$  and  $U_i$  determine the lower bound value. We want to show that both  $L_i$  and  $U_i$  are “tight”. It can happen that for certain pairs of  $Q, C$ , the exact distance is equal to  $LB(Q, C)$  but in the computation of  $LB(Q, C)$  not both of  $L_i$  and  $U_i$  are used, and hence we cannot be sure that both  $L_i$  and  $U_i$  are set as tight as possible. Hence we have the following definition for tightness.

**Definition 6.** Consider a lower bound  $LB(Q, C)$  for a distance  $D(Q, C)$  of the form

$$LB(Q, C) = \sum_{i=1}^m \begin{cases} (Q_i - U_i)^2 & \text{if } Q_i > U_i \\ (Q_i - L_i)^2 & \text{if } Q_i < L_i \\ 0 & \text{otherwise} \end{cases}$$

We say that the lower bound is tight, if there exists a set of sequence pairs so that for each pair  $\{Q, C\}$  in the set,  $D(Q, C) = LB(Q, C)$ , and the  $U_i$  and  $L_j$  values for some  $i, j$  are used (in the  $(Q_i - U_i)^2$  or  $(Q_j - L_j)^2$  term) at least once in computing the lower bounds in the set.

**Lemma 3.** The lower bound  $LB_W(Q, C)$  for the DTW distance with the Sakoe-Chiba Band constraint is tight.

*Proof.* Consider DTW with a Sakoe-Chiba Band constraint of  $r = 1$ . Hence in the warping path entry  $(i, j)$ ,  $j - 1 \leq i \leq j + 1$ .

Select two pairs of  $\{Q, C\}$  as follows (illustrated in Figure 6):

$$Q = \{1, 0.9, 2, 3, 4\}, C = \{1, 2, 2, 3, 4\}; \\ Q' = \{1, 2, 3, 4.1, 4\}, C' = \{1, 2, 3, 3, 4\}$$

It is easy to see that  $D(Q, C) = LB_W(Q, C)$ , and  $D(Q', C') = LB_W(Q', C')$ .

For  $Q, C$ ,  $Q_2 < LW_2$  and hence  $LW_2$  is used in the computation of  $LB_W(Q, C)$ .

For  $Q', C'$ ,  $Q'_4 > UW'_4$ , hence  $UW'_4$  is used in the computation of  $LB_W(Q', C')$ .  $\square$

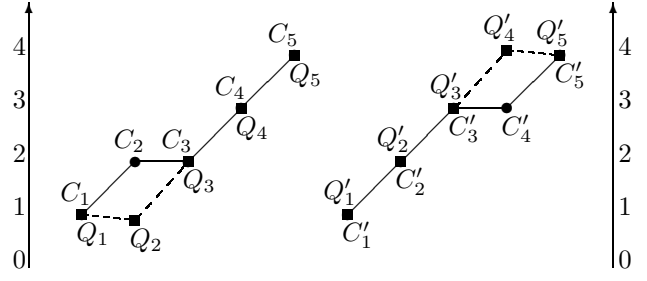


Figure 6: Example sequence pairs  $(Q, C)$  in Lemma 3

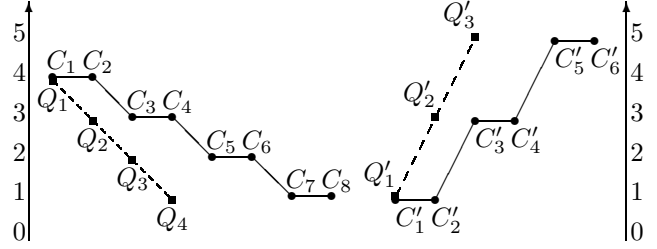


Figure 7: Example sequence pairs  $(Q, C)$  in Lemma 4

**Lemma 4.** The lower bound  $LB_S(Q, C)$  for the distance between  $Q, C$  with a scaling factor between  $1/l$  and  $l$  is tight.

*Proof.* Consider scaling between 0.5 and 2. Hence  $l = 2$ .

Select two pairs of  $\{Q, C\}$  as follows (illustrated in Figure 7):

$$Q = \{4, 3, 2, 1\}, C = \{4.1, 4.1, 3.1, 3.1, 2.1, 2.1, 1.1, 1.1\}; \\ Q' = \{1.1, 3.1, 5.1\}, C' = \{1, 1, 3, 3, 5, 5\}$$

It is easy to see that  $D(Q, C) = LB_S(Q, C)$ , and  $D(Q', C') = LB_S(Q', C')$ .

For  $Q, C$ ,  $LC_i > Q_i$  and all  $LC_i$  are used in the computation of  $LB_S(Q, C)$ .

For  $Q', C'$ ,  $UC'_i < Q'_i$  and all  $UC'_i$  are used in the computation of  $LB_S(Q', C')$ .  $\square$

**Lemma 5.** The lower bound  $LB(Q, C)$  for the distance between  $Q, C$  with a scaling factor bound  $l$  and time warping with the Sakoe-Chiba Band constraint  $r'$  is tight.

*Proof.* Consider a Sakoe-Chiba Band constraint of  $r' = 1$  and a scaling factor between 0.5 and 2. Hence  $l = 2$ .

Select two pairs of  $\{Q, C\}$  as follows (illustrated in Figure 8):

$$Q = \{3, 1.9, 2, 1\}, C = \{3, 3, 3, 3, 2, 2, 1, 1\}; \\ Q' = \{1, 2, 3.1, 3\}, C' = \{1, 1, 2, 2, 2, 3, 3\}$$

It is easy to see that  $SWM(Q, C, l, r') = LB(Q, C)$ , and  $SWM(Q', C', l, r') = LB(Q', C')$ .



For  $Q, C, Q_2 < L_2$  and  $L_2$  is used in the computation of  $LB(Q, C)$ .

For  $Q', C', Q'_3 > U'_3$  and  $U'_3$  is used in the computation of  $LB(Q', C')$ .  $\square$

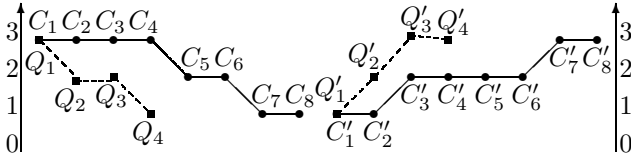


Figure 8: Example sequence pairs  $(Q, C)$  in Lemma 5

## 6 Experimental Evaluation

This section describes the experiments carried out to verify the effectiveness of the proposed lower bounding distance. The experiments were executed on an Intel Xeon 2.2GHz Linux PC with 1GB RAM. The source code for the experiments is written in C Language. MATLAB was also used for pre-processing the raw data.

To evaluate the effectiveness of the proposed lower bounding distance, and thus the proposed solution, an objective measure of the quality of a lower bounding distance is required. The *Pruning Power*  $P$  is defined in [12] as follows,

$$P = \frac{\text{Number of objects that do not require full DTW}}{\text{Number of objects in database}}$$

The Pruning Power is an objective measure because it is free of implementation bias and choice of underlying spatial index. This measure has become a common metric for evaluating the efficiency of lower bounding distances, therefore, it was adopted in evaluating the proposed lower bounding distance.

Extensive experiments were conducted on as many as 41 different datasets. These datasets, which represent time series from different domains, were obtained from “The UCR Time Series Data Mining Archive” [13].

As the datasets came from a wide variety of different domains, they differed significantly in size and in the length of individual data sequences. In order to produce meaningful results, both parameters must be controlled. Thus, from each original dataset, we derived six sets of data, each containing 1024 data sequences, with variable lengths of 32, 64, 128, 256, 512 and 1024, respectively. Short sequences were produced by using only prefixes of the original datasets while long sequences were produced by concatenating original sequences. All experiments were conducted on these derived datasets.

To compute the pruning power of the proposed lower bounding distance, the 1-nearest neighbor search was performed using the linear-scan algorithm. A random subsequence was chosen from the dataset to act as the query, and the remaining 1023 sequences acted as the data. The search was repeated for 50 trials using a different subsequence as query. The actual dynamic time warping distance did not need to be calculated if the lower bounding measure gave a value larger than the time warping distance of the current nearest neighbor. The fraction of sequences that did not require calculation of actual time warping distance became the pruning power of the lower bounding measure in that query. The average of the 50 queries were reported as the pruning power of that particular dataset.

Unless stated otherwise, in all experiments, the length of data was 1024 data points; the scaling factor was between 1.5 and its reciprocal; the length of query was set so that the longest rescaled query is at most as long as the data; and the width of the Sakoe-Chiba Band was set to 10% of the length of the query. In fact, recent evidence suggests that this is a pessimistic setting, and real world problems benefit from even tighter constraints [21].

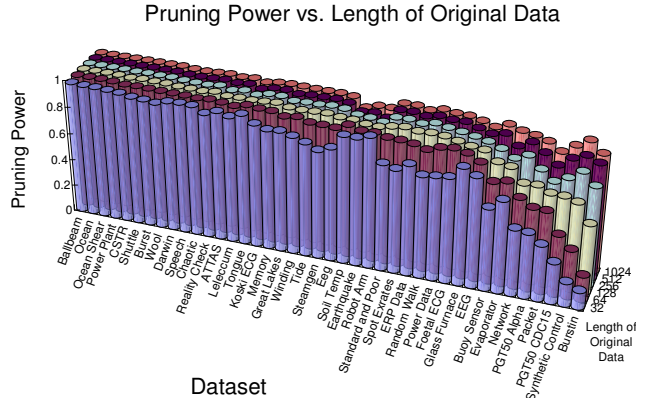


Figure 9: Pruning power vs. length of original data

Figure 9 shows how the pruning power of the proposed lower bounding measure varies as the lengths of data change on different datasets.<sup>4</sup> For a majority of datasets, the pruning power increased with the length of data, suggesting that the proposed algorithm is likely to perform well in real-life environment, in which long sequences of data are collected for a long period of time. More than 78% (32 out of 41) of the datasets obtained a pruning power above 90%. All but three of the datasets exhibited a pruning power of over 90% at length 1024. Even at length 32, over 75% pruning power was achieved in 80% (33 out of 41) of the datasets. Figure 10 shows the pruning power

<sup>4</sup>We note that some of the figures in this section suffer from monochromatic reproduction. We encourage the interested reader to visit <http://www.cs.ucr.edu/~eamonn/VLDB2005/> for large scale color graphics with additional details.

averaged over all datasets; 97% of data sequences of length 1024 and 80% of data sequences of length 32 did not require computation of the actual time warping distances. Figure 9 may contain too much information so we pick 6 of the more significant applications to show the pruning power for them more clearly in Figure 11. The applications include CSTR (speech), ECG, Ocean, Shuttle, Wool and chaotic.

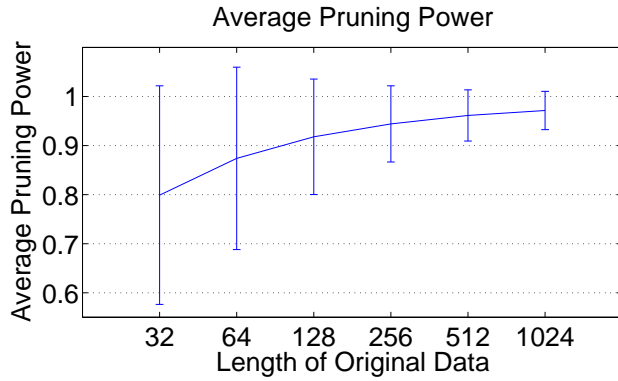


Figure 10: Average pruning power vs. length of original data

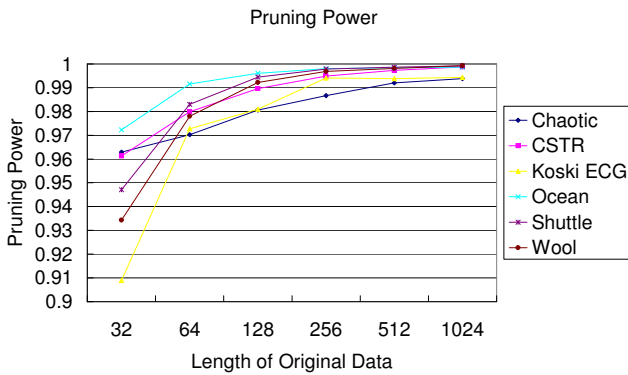


Figure 11: Some significant applications

The promising pruning power will greatly reduce the querying time. We conducted experiments to measure the time required for query evaluation in all the 41 datasets. We compare the brute force approach to the pruning approach. In the timing we included both the time spent on the pruning and the post-processing where the SWM distances for remaining sequences are actually computed. Figure 12 shows the results. The time is consistently reduced, down to about 13% of the time required by brute force search. We have repeated this with some other parameters and the results are similar.

Figure 13 shows the effect of varying the range of allowed scaling factors on pruning power. Note the x-axis indicates the upper bound range of allowed scaling factor. The lower bound range of allowed scaling factor is the reciprocal of the upper bound. For instance,

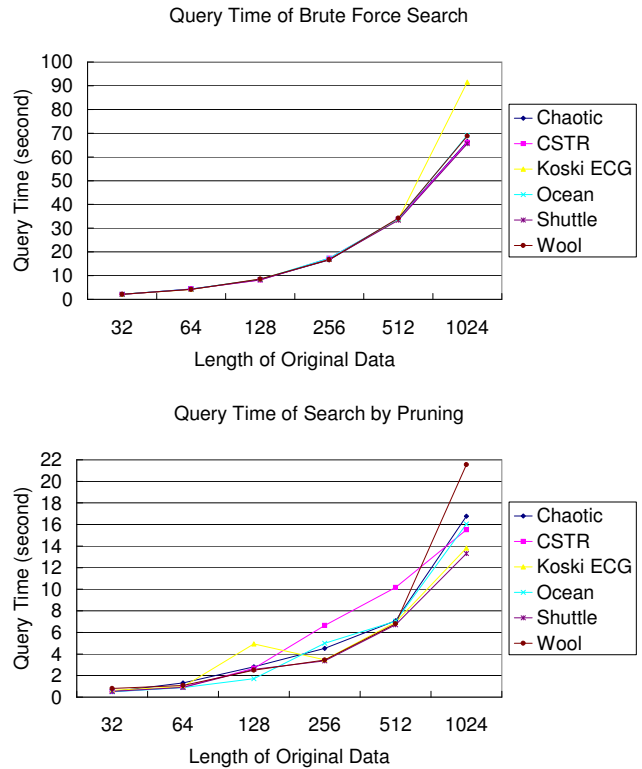


Figure 12: Query time comparison

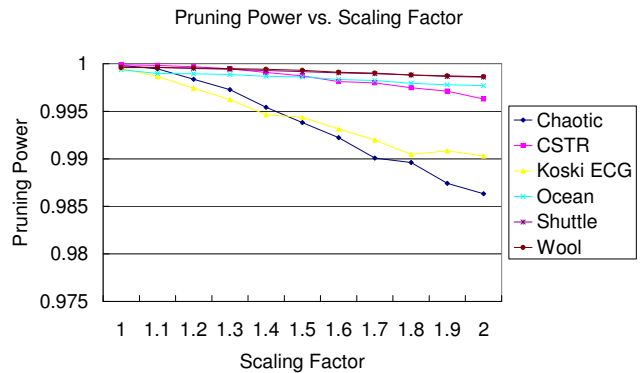


Figure 13: Varying the scaling factor

the label 2 indicates that the range of allowed scaling factor is between  $1/2 = 0.5$  and 2. In particular, the label 1 indicates that the time warping distance was calculated without scaling. It also implied that the size of the range was not increasing linearly. Although we show only 6 of the significant applications, we have experimented on all 41 sets of real data, the important observation is that for all sizes, a pruning power of over 90% was achieved in nearly 83% (34 out of 41) of the datasets. For all datasets (of length 1024), the pruning powers never dropped below 80%.

A more detailed look into the actual data provided some insights as to why most datasets give very high pruning power and why the few other datasets result in less pruning power. Figure 14 shows sample sequences from the two datasets that give the lowest pruning power. And Figure 15 shows the sample sequences from the two datasets that give the highest pruning power. The difference between them is rather obvious visually. The sequences giving the lowest pruning power are those that fluctuate vigorously. The sequences giving the highest pruning power are those that are rather smooth. This is because with vigorous data fluctuation, the lower and upper bound envelope will be loose, and the pruning power will be weakened.

Nevertheless, we note that vigorously fluctuating datasets are far less common than smooth datasets. Figure 16 illustrates this claim by showing the pruning power averaged over all the datasets, as the range of allowed scaling factor changes. For all scaling factors, the average pruning powers are always above 95%. Even if we allow for one standard deviation margin below the average, the pruning power is still above 90% in general.

In conclusion, the result shows that the proposed lower bounding measure effectively speeds up the query evaluation process. It also confirms the applicability of the lower bounding technique, even when a tight lower bound may not be readily obtainable.

## 7 Conclusion

We show the importance of the problem of SWM: handling scaling and time warping distance in time series querying. Since the direct computation of SWM distance is very costly, we propose a lower bound technique for pruning the search space. The strategy is that we first compute a lower bound for the SWM distance, the computation of which is very simple. If the lower bound does not fall within our current search range, we can prune the data sequence since it cannot be in our qualified result. We show by experiment that we can typically prune over 90% of the search space for 1-nearest neighbor search in a large variety of dataset. Our method can easily be extended to cover  $k$ -nearest neighbor search.

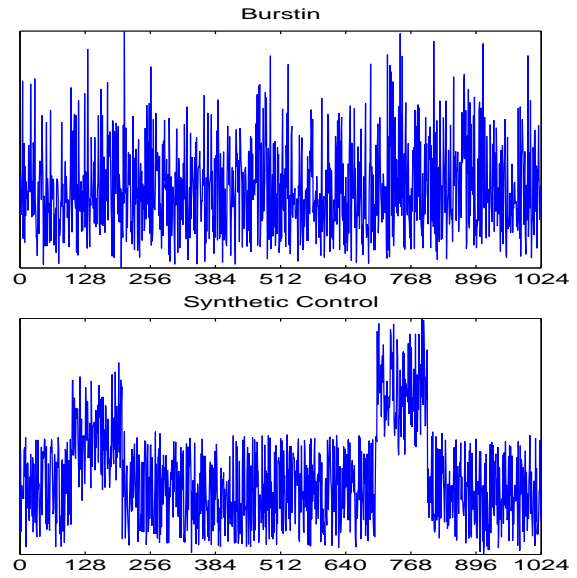


Figure 14: Data giving the lowest pruning power

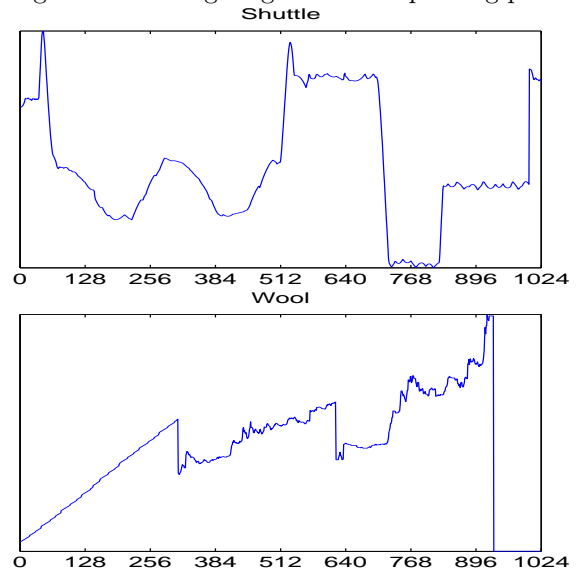


Figure 15: Data giving the highest pruning power

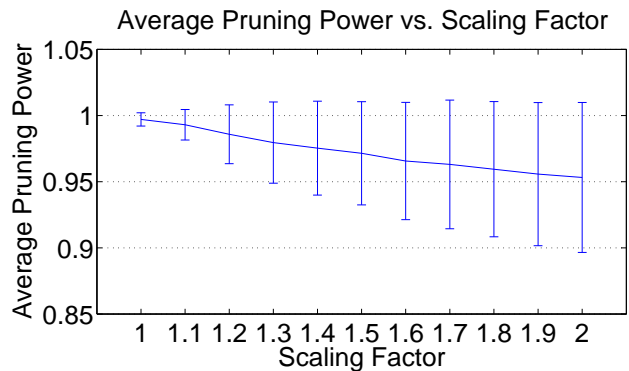


Figure 16: Average pruning power vs. scaling factor

For future work we shall examine indexing techniques for the SWM problem. Standard index structures such as the R-Tree [8], the R\*-Tree [1] or the X-Tree [2] can be used. We also may consider mechanisms to reduce the dimensionality of the time series for better performance. Another important problem is to handle sub-sequence matching for a query sequence that can match any portion of a data sequence. Our work can be extended in this direction along with an indexing method based on previous work such as [7] and [19].

**Acknowledgements:** This research was supported by the RGC Research Direct Grant 03/04, the RGC Earmarked Research Grant of HKSAR CUHK 4179/01E, and 4209/04E, and the Innovation and Technology Fund (ITF) in the HKSAR [ITS/069/03].

## References

- [1] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R\*-tree: An efficient and robust access method for points and rectangles. In *SIGMOD '90: Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, 1990.
- [2] S. Berchtold, D. A. Keim, and H.-P. Kriegel. The X-tree: An index structure for high-dimensional data. In *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases*, 1996.
- [3] L. Campbell and A. Bobick. Recognition of human body motion using phase space constraints. In *Proceedings of International Conference on Computer Vision*, 1995.
- [4] W. Chai and B. Vercoe. Folk music classification using hidden markov models. In *Proceedings of International Conference on Artificial Intelligence*, 2001.
- [5] F. Chan and A. Fu. Efficient time series matching by wavelets. In *Proceedings of the 15th International Conference on Data Engineering*, 1999.
- [6] N. Dalal and R. Horaud. Indexing key positions between multiple videos. In *Proceedings of IEEE Workshop on Motion and Video Computing*, 2002.
- [7] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *SIGMOD '94: Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, 1994.
- [8] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *SIGMOD '84: Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, 1984.
- [9] F. Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1), 1975.
- [10] A. Kale, R. Chowdhury, and R. Chellappa. Towards a view invariant gait recognition algorithm. In *Proceedings of the IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS)*, 2003.
- [11] A. Kale, N. Cuntoor, B. Yegnanarayana, A. Rajagopalan, and R. Chellappa. Gait analysis for human identification. In *Proceedings of the 3rd International Conference on Audio and Video Based Person Authentication*, 2003.
- [12] E. Keogh. Exact indexing of dynamic time warping. In *VLDB 2002, Proceedings of 28th International Conference on Very Large Data Bases*, 2002.
- [13] E. Keogh and T. Folias. The UCR Time Series Data Mining Archive. Available at <http://www.cs.ucr.edu/~eamonn/TSDMA/index.html>, 2002. Riverside CA. University of California - Computer Science & Engineering Department.
- [14] E. Keogh, T. Palpanas, V. B. Zordan, D. Gunopoulos, and M. Cardle. Indexing large human-motion databases. In *VLDB 2004, Proceedings of 30th International Conference on Very Large Data Bases*, 2004.
- [15] N. Kosugi, Y. Sakurai, and M. Morimoto. SoundCompass: A practical query-by-humming system. In *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, 2004.
- [16] G. Matessi, A. Pilastro, and G. Marin. Variation in quantitative properties of song among European populations of the reed bunting (*Emberiza schoeniclus*) with respect to bill morphology. *Can. J. Zool.*, 78, 2000.
- [17] C. Meek and W. Birmingham. The dangers of parsimony in query-by-humming applications. In *Proceedings of International Symposium on Music Information Retrieval*, 2003.
- [18] C. Moeller-Levet, F. Klawonn, K.-H. Cho, and O. Wolkenhauer. Fuzzy clustering of short time series and unevenly distributed sampling points. In *Proceedings of IDA*, 2003.
- [19] Y.-S. Moon, K.-Y. Whang, and W.-K. Loh. Duality-based subsequence matching in time-series databases. In *Proceedings of the 17th International Conference on Data Engineering*, 2001.
- [20] K. Pullen and C. Bregler. Motion capture assisted animation: Texturing and synthesis. *ACM Transactions on Graphics (Proc SIGGRAPH 2002)*, 22(3), 2002.
- [21] C. A. Ratanamahatana and E. Keogh. Everything you know about dynamic time warping is wrong. In *3rd Workshop on Mining Temporal and Sequential Data, in conjunction with the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, 2004.
- [22] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1), 1978.
- [23] B.-K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proceedings of the 14th International Conference on Data Engineering*, 1998.
- [24] Y. Zhu and D. Shasha. Warping indexes with envelope transforms for query by humming. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, 2003.