

Rotation Invariant Distance Measures for Trajectories

Michail Vlachos
UCR
mvlachos@cs.ucr.edu

D. Gunopulos
UCR
dg@cs.ucr.edu

Gautam Das
Microsoft Research
gautamd@microsoft.com

ABSTRACT

For the discovery of similar patterns in 1D time-series, it is very typical to perform a normalization of the data (for example a transformation so that the data follow a zero mean and unit standard deviation). Such transformations can reveal latent patterns and are very commonly used in datamining applications. However, when dealing with multidimensional time-series, which appear naturally in applications such as video-tracking, motion-capture etc, similar motion patterns can also be expressed at different orientations. It is therefore imperative to provide support for additional transformations, such as rotation. In this work, we transform the positional information of moving data, into a space that is translation, scale and rotation invariant. Our distance measure in the new space is able to detect elastic matches and can be efficiently lower bounded, thus being computationally tractable. The proposed methods are easy to implement, fast to compute and can have many applications for real world problems, in areas such as handwriting recognition and posture estimation in motion-capture data. Finally, we empirically demonstrate the accuracy and the efficiency of the technique, using real and synthetic handwriting data.

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications, *Data Mining*

General Terms: Algorithms

Keywords: Trajectories, Time Warping, Rotation Invariance

1. INTRODUCTION

Advances in wireless communication, sensor devices and GPS technology make it possible nowadays to collect large amounts of *trajectory* data. A trajectory is a set of positional information of a moving object, ordered by time. Examples include tracking animals, gathering human motion data by tracking various body joints, or tracing the evolution of migrating particles in biological sciences. In fact, multi-dimensional trajectory data are prevalent in many fields such as environmental information systems, meteorology, wireless technology, video tracking, or video motion capture.

An important problem in such applications is designing techniques for identifying trajectories that are similar. Such techniques can be used to cluster sets of trajectories, to in-

dex sets of trajectories so that nearest neighbor queries can be executed efficiently, and to classify new trajectories given a set of examples.

While the design of distance functions between trajectories has been considered in recent work [15, 14, 7], none of these techniques consider similarity measures that are *rotation invariant*. In many applications, a similarity measure that is immune to rotation effects is desirable. To illustrate this, consider a hand-writing recognition application, where each letter is represented by a two-dimensional trajectory. In this application, we are interested in classifying a new trajectory, given a set of letter examples. For example, all the trajectories in Figure 1 represent the same letter, so they should be considered similar to one another. However, none of the currently proposed techniques can tackle the problem effectively, primarily because they are not capable of eliminating the rotation component from the similarity calculations.

In this paper we seek distance measures for trajectories that are invariant under rotations. Furthermore, we seek distance functions that are efficient to compute, so that operations such as clustering and indexing for nearest neighbor calculations do not suffer from performance penalties. Our main contributions are the following:

First, we present a distance measure that allows us to find similar trajectories under translation, scaling and rotational transformations. We adapt techniques developed in [4] to first map each trajectory to a trajectory in a rotation invariant space. We extend the robustness of this work by introducing a novel *iterative modulo normalization* technique. The distance of two trajectories in the new rotation invariant space, is computed using Dynamic Time Warping (DTW). We show that the resulting distance measure is robust under the desired family of transformations (translation, rotation, scaling), and moreover, is very efficient to compute. We provide experimental evidence to demonstrate the accuracy of our distance measure.

Second, we give a new technique for *lower-bounding* our distance measure. Our lower-bounding is a general technique for lower-bounding the DTW distance, and is therefore of independent interest. We compare our technique with recent efforts on this problem, and show that our approach is efficient and more accurate than previously known techniques.

2. RELATED WORK

The problem that we examine in this paper is related to problems studied in the areas of pattern recognition and computer vision, such as testing for congruence of point sets [1, 5], which is a fundamental (and still open) problem in computer vision. Also related to this general area are prob-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

Copyright 2004 ACM 1-58113-888-1/04/0008 ...\$5.00.

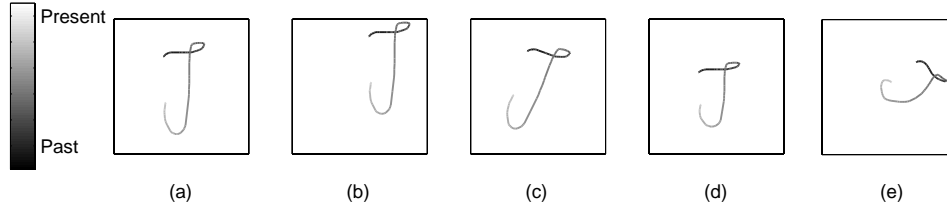


Figure 1: Transformations supported by our representation. Trajectories plotted on 2D for clarity. Time is depicted by the intensity of the color. (a) Original sequence, (b) Translation, (c) Rotation, (d) Scaling, (e) Transformation including (b),(c),(d) and deformation

lems in medical image registration [10], and geometric hashing [6].

Inspired by the work of Cohen and Guibas [4], we use the turning angle transformation as our invariant matching signature. The work of [4] deals with planar lines (and not trajectories), where the x-y position of two dimensional line shapes are transformed into an *angle/arc-length* space. The partial (subsequence) matching problem, is solved by a line sweep algorithm in a scale-shift plane and guarantees scale, translation and rotation invariance. This comes at a high cost of $O(m^2n^2)$ time complexity for lines of length m and n respectively. Moreover, while this method can handle global scaling, there is no support for local scaling. Compared to other methods based on algebraic moments, curvature [11] and Hausdorff distance [3], the turning angle has been shown to be the most robust for the retrieval of two dimensional images and shapes [12].

3. ROTATION INVARIANT TRAJECTORY DISTANCE FUNCTIONS

Let P be a trajectory of a moving object. P is then a sequence of 2-dimensional vectors¹, each describing the position of the object at time instants 0 to $n - 1$:

$$P = [P_1, \dots, P_n] \\ = [(p_{x,0}, p_{y,0}), \dots, (p_{x,n-1}, p_{y,n-1})]$$

Let F be a distance measure for trajectories. F can be, for example, the Euclidean metric, or the Dynamic Time Warping distance. Let \mathcal{R} be the set of two dimensional rotations around the axis origin. Let also \mathcal{T} be the set of two-dimensional translations and \mathcal{S} be the set of all scaling operations by a global factor.

Figure 1 describes the effects of different transformations on a trajectory.

Let $\mathcal{T}\nabla$ be the set of transformations that are composed by rotations, translations and scaling.

Given two 2-dimensional trajectories T, Q , and a distance measure F , we are interested in computing the smallest possible distance of T and Q if we allow transformations in $\mathcal{T}\nabla$ to be applied to one of the trajectories.

$$F_{inv}(T, Q) = \min_{r \in \mathcal{T}\nabla} F(r(T), Q)$$

This new distance measure, F_{inv} can be expensive to compute, therefore here we transform the trajectories into a new space with invariant characteristics where the matching can be more efficiently performed.

¹Although this paper considers mainly 2-dimensional applications, the techniques can be extended to higher dimensions.

3.1 Rotation Invariant Transformations

It is difficult to design distance measures that are invariant to rotation using positional information directly. Instead, we use a modification of a technique by [4] to map the trajectories to a space that is more amenable to this purpose. We briefly describe the technique below:

We define the *movement vector* V at time t as:

$$V_t = P_t - P_{t-1}, \quad t = 1, \dots, n - 1$$

We extract the angles of each movement vector relative to a reference movement vector V_{ref} . This reference vector is the positive x-axis or vector $[1, 0]$. In other words, we calculate how much is the acute rotation angle that is required to align the vector V_{ref} to the direction of the current movement vector V_t .

The acute angle α between these two vectors is computed utilizing the dot product as follows:

$$\alpha_t = \hat{V}_t = \text{sign} \cdot \cos^{-1} \left(\frac{\langle V_t, V_{ref} \rangle}{\|V_t\| \cdot \|V_{ref}\|} \right)$$

and $\langle \cdot, \cdot \rangle$ signifies the vector dot product.

However, the dot product does not specify the *direction* of the rotation, which is denoted by *sign*. (e.g., clockwise or counterclockwise). This information is obtained using the direction of the cross product between V_t and V_{ref} , by examining whether it spans towards the positive or the negative z-axis. For 2D trajectories the sign is calculated as:

$$\text{sign} = \begin{cases} 1 & \text{if } ([V_t \times V_{ref}] \cdot [0 \ 0 \ 1]^T) > 0 \\ -1 & \text{if } ([V_t \times V_{ref}] \cdot [0 \ 0 \ 1]^T) < 0 \end{cases}$$

In addition to the angle of each movement vector we also record its Euclidean length L (arc length). Therefore, we have transformed the spatial coordinates of a trajectory into a sequence of **Angle/Arc-Length** pairs. We call this new transformed space, **AAL space** and figure 2 shows how the AAL representation of a 2-dimensional trajectory is computed.

$$P_{AAL} = [(\hat{V}_1, \|V_1\|), \dots, (\hat{V}_{n-1}, \|V_{n-1}\|)]$$

We also consider different approaches to define the reference vectors required by the AAL transformation. As a reference vector, instead of the positive x-axis (which we call *exact angles*), one could use other vectors as well. In this work we also consider the following:

Angle Relative to Previous (*relative angles*):

$$V_{ref} = P_{t-1} - P_{t-2}$$

The problem with this approach is that small differences in the angles can be compounded in the original trajectory, resulting to a large deviation between two sequences, while the distance in the AAL space remains small.

Angle Relative to center of Mass (*cMass angles*):

The center of mass is defined as the average of the different locations of the object over time:

$$P_{cMass} = \frac{1}{n} \left[\sum_n p_{x,n}, \sum_n p_{y,n} \right] \text{ and } V_{ref} = P_{t-1} - P_{cMass}$$

In the experimental section we show that both exact and *cMass* angles, have good robustness under real or synthetic datasets. The relative angles perform well for the synthetic ones, but for real datasets their performance degrades rapidly.

Representation of trajectories in the new space offers significant advantages over the spatial representation, because it is translation invariant. Additionally when we are computing exact angles from a reference vector, rotation of a trajectory by θ degrees this will result in a shifted pattern by θ degrees in the AAL space. Finally, scale invariance can be accomplished by dividing the arc length of each movement vector by the total arc length of the trajectory.

$$P_{AAL} = \left[\left(\hat{V}_1, \frac{\|V_1\|}{\sum_i \|V_i\|} \right), \dots, \left(\hat{V}_{n-1}, \frac{\|V_{n-1}\|}{\sum_i \|V_i\|} \right) \right]$$

4. ITERATIVE MODULO NORMALIZATION

After the transformation of the x-y position of the trajectories into the angle/arc-length space (using the *exact angles*), similar but rotated trajectories will depict analogous patterns in the new space. However, the patterns may differ by some amount of vertical shift according to their orientation.

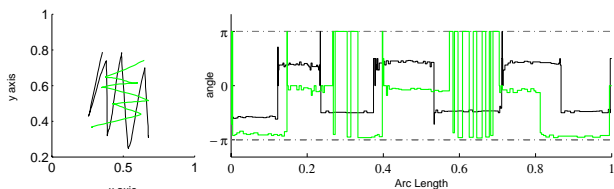


Figure 3: *Left:* Trajectories plotted on 2D. *Right:* Angle/Arc-Length features extracted from the trajectories.

To create a rotation invariant transformation we normalize the transformed sequences. A simple normalization would be to subtract the average angle value. This, nonetheless, is not sufficient because this average value can be significantly distorted. For example, figure 3 shows two similar (but not identical) 2D trajectories. Their similarity is more prevalent if the darker one is rotated clockwise by around 70° . The patterns in the AAL space appear very similar but they are distant by a vertical shift corresponding to 70° in the angle axis. Additionally, one can observe that some sections of the 2nd trajectory appear very noisy in the AAL space. This is attributed to the fact that certain consecutive movement vectors can oscillate around the π angle. Figure 4 demonstrates the two feature patterns when the average angle value is subtracted from both trajectories. Obviously, an even better matching can be found (see Figure 5).

For this purpose we introduce an iterative normalization procedure. The pseudo-code of the algorithm is illustrated in Figure 6 and it performs two basic steps:

- A normalization, by subtracting the average value of the sequence
- A modulo operation that wraps the angle values within the range of $[-\pi, \pi]$

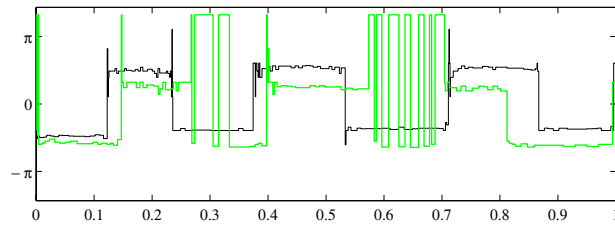


Figure 4: After Regular Normalization (subtraction of mean value)

These phases are repeated a certain number of times or until stability. For simplicity in all our experiments we iterate the normalization steps five times, which proves to be more than adequate for achieving stability.

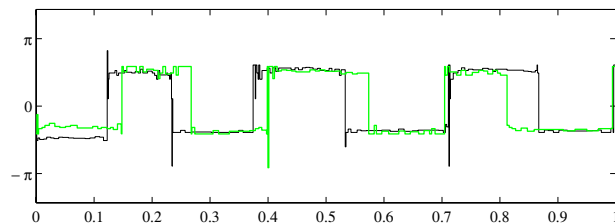


Figure 5: After Iterative Normalization. Now the similarity between the two sequences is much more obvious.

```

1 seqNorm = iterativeNormalization(sequence)
2 {
3   n = sequence.Length;
4   seqNorm = sequence;
5
6   // repeat k times or until stability
7   for i=1 to k
8   {
9     if avgValue != average(seqNorm) break;
10    // subtract mean value
11    avgValue = average(seqNorm);
12    seqNorm = seqNorm - avgValue;
13
14    // wrap points in range [-pi, pi]
15    for j=1 to n
16    {
17      if seqNorm[j]< -pi seqNorm[j]= 2*pi+seqNorm[j];
18      if seqNorm[j]> pi seqNorm[j]= -2*pi+seqNorm[j];
19    }
20 }
21}
    
```

Figure 6: Algorithm Iterative Modulo- π Normalization

5. SUPPORT FOR ELASTIC MATCHING

Using the previous transformations we have addressed the issues of translation, rotation and scaling. However, in order for the two patterns in Figure 5 to be matched effectively it is necessary to provide support for local compressions and decompressions in the arc-length axis. To accomplish that we use a *warping distance measure* in the AAL space.

The most widely used measure that supports local compressions and decompressions is Dynamic Time Warping (DTW) [2, 8, 14]. In order to utilize the DTW one assumes that the measurement values (in this case the angles), are taken at discrete and equi-spaced time instances

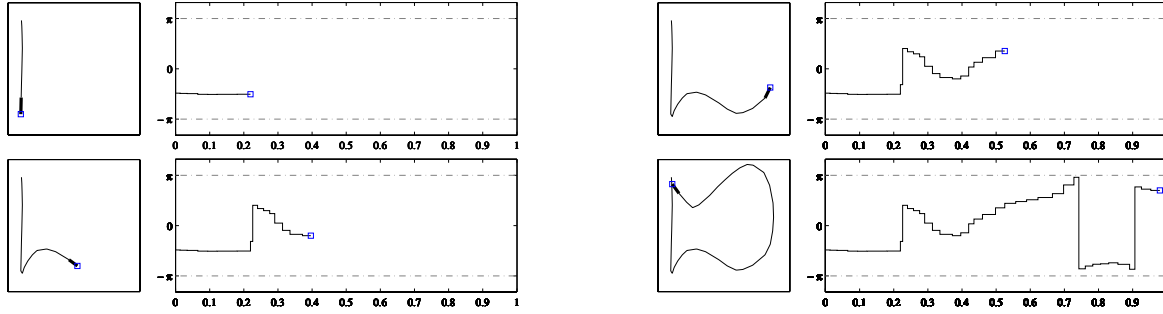


Figure 2: An example of the Angle/Arc-Length features extracted. The figures are read from top to bottom and from left to right.

(arc lengths for our case), which we achieve this by resampling/interpolating the sequence of angle/arc-length pairs.

If $Q = (Q_1, Q_2, \dots, Q_n)$ and $Head(Q) = (Q_1, Q_2, \dots, Q_{n-1})$ (and similarly for a sequence T) then the recursive equation to provide then warping distance between Q and T is:

$$DTW(Q, T) = D(Q_n, T_n) + \min \begin{cases} DTW(Head(Q), Head(T)) \\ DTW(Head(Q), T) \\ DTW(Q, Head(T)) \end{cases}$$

where $D(\cdot, \cdot)$ is the distance between two points of the sequence. Typically, D is the Euclidean distance, but it can be any distance complying to the requirements of a specific application.

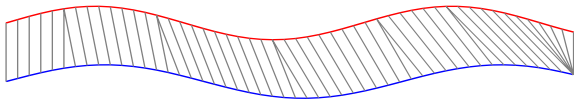


Figure 7: Elastic matching achieved by DTW.

In our setting of this problem we use a different distance function in the warping definition. We call the new distance D_{warp} and we use it to mitigate the potential wrap-around effect of the turning angles:

$$D_{warp}(Q_i, T_i) = \min \begin{cases} |Q_i - T_i| \\ 2\pi - |Q_i - T_i| \end{cases}$$

The DTW distance can be computed using a well known dynamic programming algorithm in $O(n^2)$ time for sequences of length n .

6. APPROXIMATING THE DTW

Since the $O(n^2)$ complexity of DTW quickly becomes a bottleneck for large datamining tasks, one can reduce the warping scope within a matching region of δ^2 effectively reducing the complexity into $O(\delta n)$. This simplification is realistic for most real applications and in many cases it can also improve accuracy by limiting the number of degenerate matchings³.

We revisit some of the most recent DTW lower bounds. The work of Keogh [8] instigated an enhanced interest in the Warping Distance, by introducing a new tight lower bound, called LB-Keogh. The idea is based on the notion of the Minimum Bounding Envelope (MBE), which records the areas of possible matching, when local displacement is constrained within a region δ . The MBE of a sequence Q

²That is, allow matchings between two points if they are δ points apart.

³Typical values for δ are 20 – 30% of n .

consists of the area defined between the upper envelope U and the lower envelope L and their values at position i is given by:

$$L_i = \min(Q_{i-\delta}, \dots, Q_{i+\delta}), \quad U_i = \max(Q_{i-\delta}, \dots, Q_{i+\delta})$$

The author further showed that this lower bounding function can be indexed, by introducing a reduced dimensionality version of it called $LB - PAA$, which is illustrated in figure 8. This is achieved by creating a simplified representation of the query MBE and all database sequences, by converting them into k equi-length Minimum Bounding Rectangles (MBRs)⁴. Each of the k MBRs of Q consists of a tuple, $[\hat{q}_{i,L}, \hat{q}_{i,U}]$ with values:

$$MBR_i(Q) = \begin{cases} \hat{q}_{i,L} = \min(L_{\frac{n}{k}(i-1)+1}, \dots, L_{\frac{n}{k}i}) \\ \hat{q}_{i,U} = \max(U_{\frac{n}{k}(i-1)+1}, \dots, U_{\frac{n}{k}i}) \end{cases}$$

When segmenting any database sequence T into MBRs, using the above equation we can essentially consider that $U_i = L_i = T_i$.

The lower bound distance LB-PAA is the distance between the MBRs of the query envelope ($MBR(MBE(Q))$) and the MBRs of T ($MBR(T)$), which is shown in gray vertical lines in Figure 8).

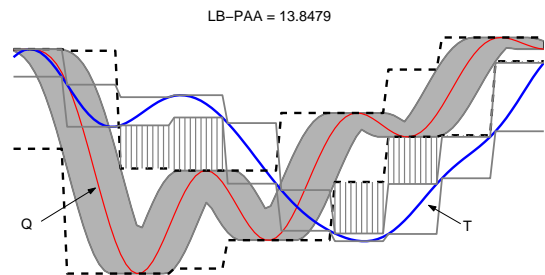


Figure 8: LB-PAA. The total length of the gray vertical lines represents the lower bound distance.

Extensions of the MBE for multidimensional sequences and introduction of new upper bounds have appeared in [14].

In [16] Zhu and Shasha, improved on the previous idea, by leveraging the lower bounding property of the PAA representation [9]. The PAA approximation is a simple and effective dimensionality reduction technique, that converts a sequence of length n into k equi-length segments, that record

⁴ k is a user defined parameter.

average values of the original sequence. That is $PAA(Q) = (\bar{q}_1, \bar{q}_2, \dots, \bar{q}_k)$, where :

$$\bar{q}_i = \frac{k}{n} \sum_{j=\frac{n}{k}(i-1)+1}^{\frac{n}{k}i} Q_j$$

Similarly as before, the MBE of the query Q is constructed, however both the envelope (U & L) and any other sequence T are approximated by their PAA representation. The new lower bound, LB-Zhu, is essentially the distance between $PAA(MBE(Q))$ and $PAA(T)$. A schematic representation is given in Figure 9.

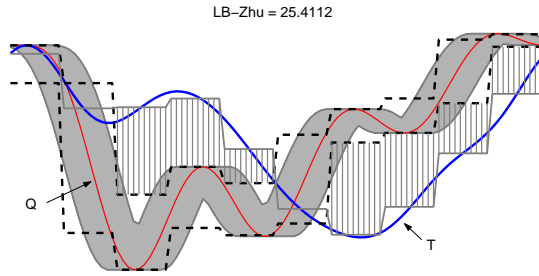


Figure 9: LB-Zhu. The lower bound on the distance is marked by the gray vertical lines.

6.1 A New Lower Bound Technique for DTW

The envelope-based approaches are fast to compute, however the tightness of the lower bound diminishes for large warping lengths (that is, when the parameter δ becomes too large). This happens because the bounding envelope becomes very wide.

We propose a different approach for approximating the DTW. In order to minimize the impact of δ (the width of the bounding envelope), we use only approximations of the query sequence Q ($MBR(Q)$), and the PAA approximation of a sequence T in the database, $PAA(T)$. The distance between the segments $MBR_i(Q) = [\hat{q}_{i,L}, \hat{q}_{i,U}]$ and $PAA_j(T) = \bar{t}_j$ is defined as:

$$D_{seg}(MBR_i(Q), PAA_j(T)) = \begin{cases} k \times \Delta(\hat{q}_{i,L}, \bar{t}_j), & \text{if } \hat{q}_{i,L} > \bar{t}_j \\ k \times \Delta(\hat{q}_{i,U}, \bar{t}_j) & \text{if } \hat{q}_{i,U} < \bar{t}_j \\ 0 & \text{otherwise} \end{cases}$$

Δ can be any distance such as the squared Euclidean or, in our case, the D_{warp} distance which mitigates the wrap-around effects.

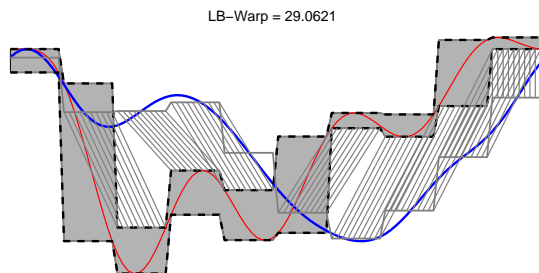


Figure 10: LB-Warp delivers a tighter lower-bound than previously used approaches.

In order to lower bound the warping distance between Q, T we just need to run a DTW computation on the approximations, using D_{seg} as the base distance function. We

call this lower bound *LB-Warp*. Additionally, the warping length needs to be modified. If the original warping computation allowed matching within δ points (out of n), it has to be modified now to $\lceil \delta(\frac{k}{n}) \rceil$. Therefore the running complexity is reduced from $O(n\delta)$, to $O(k\lceil \delta(\frac{k}{n}) \rceil)$. In Figure 11 we give an example of how the new bound is computed; no envelope is computed around the query sequence (compare with Figure 10), but the approximation segments of the two sequences can be matched with neighboring segments.

We can state the following lemma:

LEMMA 1. (*Lower Bounding Lemma*): Given trajectories Q and T of length n and warping length δ , and their approximations of length k , $MBR(Q)$ and $PAA(T)$ the following inequality holds:

$$LB - Warp_{\lceil \delta(\frac{k}{n}) \rceil}(MBR(Q), PAA(T)) \leq DTW_{\delta}(Q, T)$$

7. EXPERIMENTS

We show the rotation invariance of the new measures, using a real dataset obtained through a graphic tablet, where the x-y position of the stylus is recorded over time (*tablet real dataset*). This dataset contained 90 two-dimensional time-series, a subset of which is shown is Figure 11.

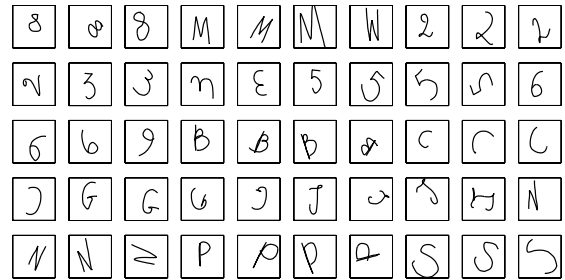


Figure 11: Subset of the dataset used in our experiments, consisting of various strokes (at different angles) written on a graphic tablet.

Based on the *tablet real dataset*, we created also a synthetic dataset, using a single orientation of each tablet stroke and creating three additional copies by rotating the template at random angles between $[-90 \dots +90^\circ]$ (*tablet synthetic*). This dataset is used as a sanity check, to examine possible performance deviations between synthetic and real datasets.

7.1 Classification Accuracy

We have conducted a leave-one-out Nearest Neighbor classification scheme, in order to measure the classification accuracy of the rotation invariant distance measures. We observe that the Exact Angles (all angles are with respect to a reference direction) and the cMass Angles techniques are both very accurate. The performance of the Relative Angles approach is good for synthetic data, however only Exact and cMass Angles give good accuracy for real data. This example indicates (once more) the great discrepancy that can exist in the performance of an algorithm between synthetic and real datasets.

In fig. 12 we show a 2D mapping of 40 handwritten strokes based on their pairwise distances. The position of the symbols on 2D plane, is derived using ISOMAP [13], which represents an improved implementation of Multidimensional Scaling. The spatial proximity in 2D of the rotated versions of the same symbol, indicates that the new representation is very robust to rotation transformations.

Method	Accuracy (Tablet Synthetic)	Accuracy (Tablet Real)
Relative Angles	98.91 %	30 %
Exact Angles	92.39 %	88.89 %
cMass Angles	100 %	92.22 %

Table 1: Accuracy rates for leave-1-out classification experiments using the handwriting dataset.



Figure 12: 2D mapping of handwritten strokes based on the rotation invariant measures.

7.2 Pruning Power of the LB-Warp

Using the sequences of the *tablet real* dataset as seeds, we generated datasets with larger cardinality (1000, 2000, 4000 and 8000) for testing the scalability and pruning power of the new lower bound.

We evaluate the pruning power of the lower bounds, using a scheme not affected by implementation details. We accomplish this by measuring the number of raw sequences retrieved from disk, in order to find the 1-Nearest-Neighbor (1NN) to a given query, averaged over 50 queries.

We observe that the new lower bound represents a significant enhancement over the previous approaches. For certain queries we may examine up to 35 times fewer sequences than LB-Keogh and up to 15 times fewer sequences compared to LB-Zhu (Figure 15).

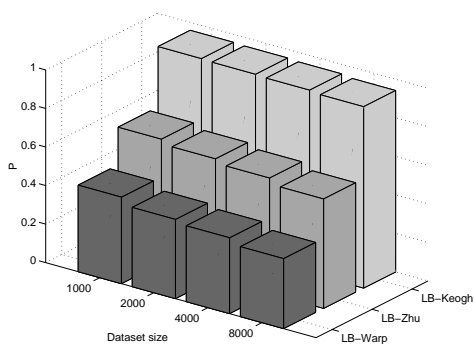


Figure 13: Fraction of the dataset objects examined, averaged over 50 queries. The new lower bound, LB-Warp, examines consistently the fewest number of sequences.

In figure 14 we report the cumulative time for returning the 1NN for 50 queries. This time includes the lower-bound distance calculation as well as the time to find the 1NN and the experiment is conducted for increasing dataset sizes. We observe that LB-Warp outperforms all previous approaches.

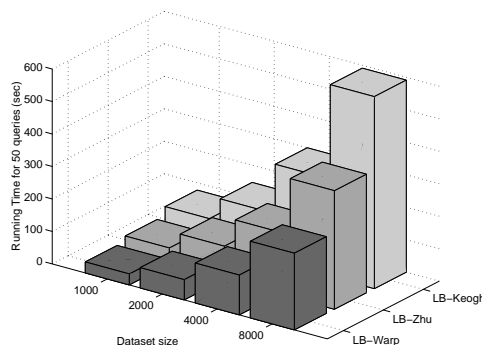


Figure 14: LB-Warp can reduce the running time by 40%, compared to next best method.

8. REFERENCES

- [1] H. Alt, K. Mehlhorn, H. Wagenet, and E. Welzl. Congruence, similarity and symmetries of geometric objects. In *Discrete Computational Geometry*, vol. 3, pages 237–256, 1988.
- [2] D. Berndt and J. Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *Proc. of KDD Workshop*, 1994.
- [3] L. P. Chew, M. T. Goodrich, D. P. Huttenlocher, K. Kedem, J. M. Kleinberg, and D. Kravets. Geometric pattern matching under euclidean motion. In *Computational Geometry*, Vol. 7, pages 113–124, 1997.
- [4] S. Cohen and L. Guibas. Partial matching of planar polylines under similarity transformations. In *Proc. of 8th ACM-SIAM Symposium on Discrete Algorithms*.
- [5] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. In *Pattern Recognition*, vol. 31, no. 8, 1998.
- [6] A. Guezic, X. Pennec, and N. Ayache. Medical image registration using geometric hashing. In *IEEE Comput. Sci. Eng. Spec. Issue Geometric Hashing*, Vol. 4, pages 29–41, 1997.
- [7] T. Kahveci, A. Singh, and A. Gurel. Similarity searching for multi-attribute sequences. In *Proc. of SSDBM*, 2002.
- [8] E. Keogh. Exact indexing of dynamic time warping. In *Proc. of VLDB*, pages 406–417, 2002.
- [9] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. In *Journal of Knowledge and Information Systems*, 2000.
- [10] J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. In *Medical Image Analysis*, 1998.
- [11] W. Rodriguez, M. Last, A. Kandel, and H. Bunke. 3-dimensional curve similarity using string matching. In *3rd International Symposium on Intelligent Manufacturing Systems, Sakarya, Turkey*, 2001.
- [12] B. Scassellati. Retrieving images by 2d shape: a comparison of computation methods with human perceptual judgments. In *Storage and Retrieval for Image and Video Databases (SPIE)*, 1994.
- [13] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science v.290 no.5500*, pages 2319–2323, 2000.
- [14] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proc. of SIGKDD*, pages 216–225, 2003.
- [15] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Proc. of ICDE*, pages 673–684, 2002.
- [16] Y. Zhu and D. Shasha. Warping indexes with envelope transforms for query by humming. In *Proc. of SIGMOD*, pages 181–192, 2003.