Robust Similarity Measures for Mobile Object Trajectories

Michail Vlachos
UC Riverside
mvlachos@cs.ucr.edu

Dimitrios Gunopulos *
UC Riverside
dg@cs.ucr.edu

George Kollios †
Boston University
gkollios@cs.bu.edu

Abstract

We investigate techniques for similarity analysis of spatio-temporal trajectories for mobile objects. Such kind of data may contain a great amount of outliers, which degrades the performance of Euclidean and Time Warping Distance. Therefore, here we propose the use of non-metric distance functions based on the Longest Common Subsequence (LCSS), in conjunction with a sigmoidal matching function. Finally, we compare these new methods to various L_p Norms and also to Time Warping distance (for real and synthetic data) and we present experimental results that validate the accuracy and efficiency of our approach, especially under the strong presence of noise.

1 Introduction

In this work we investigate the problem of discovering similar trajectories of mobile objects, especially under the presence of noise. In the last few years, the advances in mobile computing, sensor and GPS technology have made it possible to collect large amounts of spatiotemporal data and there is increasing interest to perform data analysis tasks over this data [5]. For example, in mobile computing, users equipped with mobile devices move in space and register their location at different time instances to spatiotemporal databases via wireless links. In environmental information systems, tracking animals and weather conditions is very common and large datasets can be created by storing locations of observed objects over time.

One very common application of such kind of data would be the discovery of objects that moved in a similar way or followed a certain motion pattern. Therefore, the objective is to cluster different objects into similar groups, or to classify an object based on a set of known examples. The problem is hard, because the similarity model should allow for imprecise matches. Moreover, a common obstacle that one confronts when analyzing these types of data, is the strong presence of noise, usually attributed to a tranceiver and reception problems (figure 1).

The performance of previously used metrics is generally degraded under noisy conditions, so here we formalize non-metric similarity functions, which are very robust to noise and furthermore provide an intuitive notion of similarity between trajectories by giving more weight to the similar portions of the sequences. Stretching of sequences in time is allowed, as well as global translation of the sequences in space. Efficient approximate algorithms that compute these similarity measures are also provided.

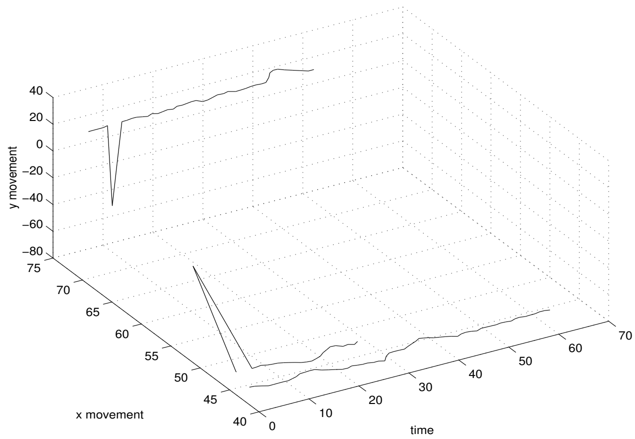

Figure 1. Examples of marine mammal tracking data, representing the movements of dolphins. The peaks represent tranceiver malfunctions.

The rest of the paper is organized as follows. In section 2 we present related work. In section 3 we formalize the new similarity functions by using a sigmoidal matching function and extending the $LCSS$ model. Section 4 demonstrates efficient algorithms to compute these functions. Section 5 provides the experimental validation of the accuracy and efficiency of the proposed approach. Finally, section 6 concludes the paper.

*Supported by NSF CAREER Award 9984729, & NSF IIS-9907477, and the DoD.

†Supported by NSF CAREER Award 0133825

2 Related Work

The simplest approach to define the similarity between two trajectories is to map each one into a vector and then use a p-norm distance to define the similarity measure. The p-norm distance is defined as $L_p(\bar{x}, \bar{y}) = (\sum_{i=1}^{n} \mid x_i - y_i \mid^p)^{\frac{1}{p}}$. For $p = 2$ it is the well known Euclidean distance and for $p = 1$ the Manhattan distance. The advantage of this simple model is that it allows efficient indexing by a dimensionality reduction technique [3, 29, 13]. On the other hand, the model cannot deal well with outliers and is very sensitive to small distortions in the time axis. There are a number of interesting extensions to the above model to support various transformations such as scaling [9, 25], shifting [9, 14], normalization [14] and moving average [25]. Other recent works on indexing time series data for similarity queries assuming the Euclidean model include [17, 16].

Another approach is based on the time warping technique that first has been used to match signals in speech recognition [26]. Berndt and Clifford [6] proposed to use this technique to measure the similarity of time-series data in data mining. The idea is to allow stretching in time in order to get a better distance. Recently, there has been approached to make this measure more scalable [18, 21].

A similar technique is to find the longest common subsequence ($LCSS$) of two sequences and then define the distance using the length of this subsequence [4, 7, 10, 8]. The $LCSS$ shows how well the two sequences can match one another if we are allowed to stretch them but we cannot rearrange the sequence of values.

Other techniques to define time series similarity are based on extracting certain features (Landmarks [22] or signatures [11]) from each time-series and then use these features to define the similarity. An interesting approach to represent a time series using the direction of the sequence at regular time intervals is presented in [24]. A domain independent framework for defining queries in terms of similarity of objects is presented in [15]. In another work, Lee et al. [20] propose methods to index sequences of multidimensional points. They extend the ideas presented by Faloutsos et al. in [12] and the similarity model is based on the Euclidean distance.

Recently, there has been some work on indexing moving objects to answer spatial proximity queries (range and nearest neighbor queries) [19, 1, 27]. Also in [23], Pfoser et al. present index methods to answer topological and navigational queries in a database that stores trajectories of moving objects. These works do not consider a global similarity model between trajectories but they concentrate on finding objects that are close to query locations during a time instant, or time period that also specified by the query.

However most of the above work deals mainly with one dimensional time-series and, moreover, don't address the issue of large amount of outliers.

3 Flexible Similarity Models for Trajectories

We would like to use a distance function that can address the following issues:

- Different sampling Rates or different speeds
- Similar motions in different space regions
- Outliers
- Different trajectory lengths

In previous work we have extended the LCSS model, in order to perform matching of 2D trajectories within a region of δ in time and a region of ϵ in space ([28]). So, whenever the points of two trajectories matched within ϵ we increased the similarity by one. This however penalized the points that were marginally outside the matching region (assigning to them a value of zero) and also made the choice of ϵ an important issue. Therefore, here we propose the use of a weighted matching function according to the distance of the points. So, the experimental results presented later, are going to be *at least* as good as the original $LCSS$ results for some optimal value of ϵ. Now, however, we don't really need to fine-tune the matching parameter. This new approach facilitates, too, the pruning of outliers since they will be given a similarity of zero (unlike the Euclidean or Time Warping Distance where *all* points, including the outliers are going to be matched).

We use the $LCSS$ paradigm in order to allow for time compression and decompression. However, for the matching function between two trajectories S1 and S2 we use a sigmoidal function and we set its matching width equal to $min(std(S1, S2))$ (figure 2). So, all the points that have greater distance than the matching threshold will not be matched (essentially the outliers), but also the weighted approach will help reveal sequences that are originally very similar to each other.

Now we are going to define the sigmoidal matching function. Let $s(x) = 1/(1+e^{-a(x-k-1)})$, $x = 1 \ldots 2k+1$. How large k is depends on the accuracy that we want to achieve. The equation describing the $SigmoidMatch$ function between two 2D points $P_1 = (a_x, a_y)$, $P_2 = (b_x, b_y)$ belonging to sequences S_1 and S_2, respectively, is:

$$\begin{cases} 0 \quad if \;\; \mathrm{L}_1(a_i, b_i) > min(stdS_{1i}, stdS_{2i}), \quad i = x, y \\ \\ \frac{1}{2}\sum_{i=x}^{y} s(\lceil \frac{|L_1(a_i,b_i)|}{min(stdS_{1i}, stdS_{2i})} \cdot (2k+1) \rceil) \quad , otherwise \end{cases}$$

One can use different parameters for the variable a, in order to indicate where more matching weight should be given. In figure 2 we can observe the different similarity assigned for different values of a. In our experiments we used a value of $a = 0.25$.

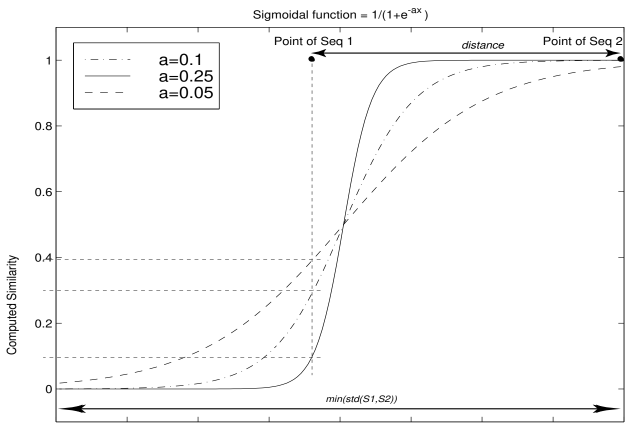

Figure 2. The $SigmoidMatch$ function for different parameters and the weighted matching procedure.

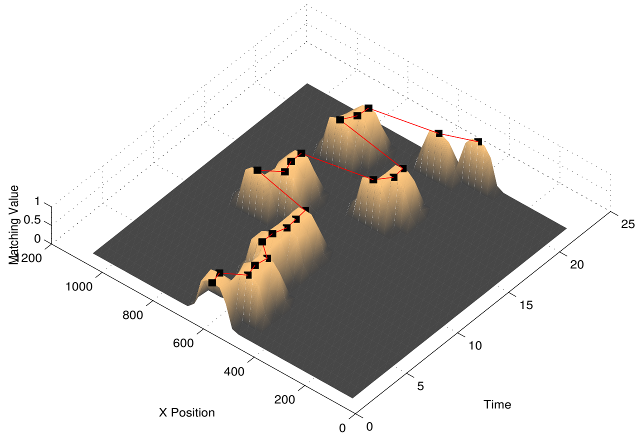

Figure 3. Sigmoidal matching of a sequence (without the time matching within δ). The similarity value at each point of another sequence, declines up to a certain cut-off point.

Let A and B be two trajectories of moving objects with size n and m respectively, where $A = ((a_{x,1}, a_{y,1}), \ldots, (a_{x,n}, a_{y,n}))$ and $B = ((b_{x,1}, b_{y,1}), \ldots, (b_{x,m}, b_{y,m}))$. For a trajectory A, let $Head(A)$ be the sequence $Head(A) = ((a_{x,1}, a_{y,1}), \ldots, (a_{x,n-1}, a_{y,n-1}))$.

Our similarity function is defined as:

Definition 1 *Given an integer δ, we define the $SigmoidSim_\delta(A, B)$ as follows:*

$$\begin{aligned} SigmoidSim(A, B) \quad = \quad & SignoidMatch(a_m, b_n) + \\ & max\{SigmoidSim(Head(A), \\ & \qquad\qquad Head(B)), \\ & \quad SigmoidSim(Head(A), B), \\ & \quad SigmoidSim(A, Head(B))\} \\ & where, \mid n - m \mid \leq \delta \end{aligned}$$

The constant δ controls how far in time we can go in order to match a given point from one sequence to a point in another sequence. An example of the sigmoidal matching (disregarding the matching within δ) is depicted in figure 3.

The first similarity function allows time stretching. So, objects that are close in space at different time instants can be matched if the time instants are also close.

Definition 2 *We define the similarity $S1$ between two trajectories A and B, given δ, as follows:*

$$S1(\delta, A, B) = \frac{SigmoidSim_\delta(A, B)}{\min(n, m)}$$

We use function $S1$ to define another, more flexible, similarity measure that will be able to detect parallel movements. First, we consider the set of translations. A translation in 1D simply causes a vertical shift either up or down. Let $\mathcal{F}$ be the family of translations in 2D. Then a function $f_{c,d}$ belongs to $\mathcal{F}$ if $f_{c,d}(A) = ((a_{x,1} + c, a_{y,1} + d), \ldots, (a_{x,n} + c, a_{y,n} + d))$. Next, we define a second notion of the similarity based on the above family of functions.

Definition 3 *Given δ, and the family $\mathcal{F}$ of translations, we define the similarity function $S2$ between two trajectories A and B, as follows:*

$$S2(\delta, A, B) = \max_{f_{c,d} \in \mathcal{F}} S1(\delta, A, f_{c,d}(B))$$

The similarity functions $S1$ and $S2$ range from 0 to 1. Therefore we can define the distance function between two sequences by subtracting 1 from the similarity. The new distance function can detect similarities between parallel movements (figure 4) and also allow time-stretching.

4 Efficient Algorithms to Compute the Similarity

4.1 Computing the similarity function $S1$

The $SigmoidSim$ can be computed by a dynamic programming algorithm in $O(n^2)$ time. However we only allow matchings when the difference in the indices is at most δ, and this allows the use of a faster algorithm. We can show that:

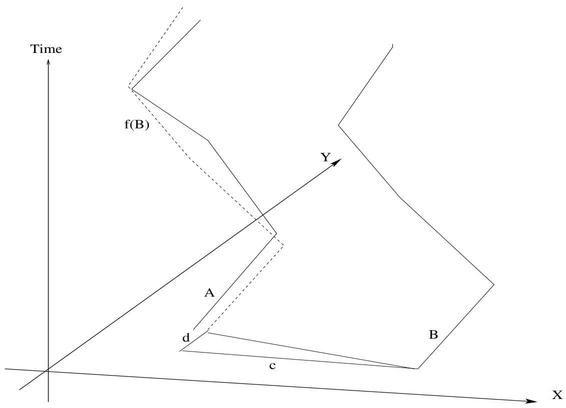

Figure 4. A trajectory B matches another trajectory A after a translation is applied.

Lemma 1 *Given two sequences A and B, with* $|A| = n$ *and* $|B| = m$*, we can find the* $SigmoidSim_{\delta,\epsilon}(A, B)$ *in* $O(\delta(n + m))$ *time.*

If δ is small, the dynamic programming algorithm is very efficient. This is realistic, since in most real datasets that we had at our disposal setting the value of δ to more than $10\% - 20\%$ of the trajectory's length did not yield any significant improvement.

4.2 Computing the similarity function $S2$

We now consider the more complex similarity function $S2$. Now, we have to find the translation $f_{c,d}$ that maximizes the matching between A and $f_{c,d}(B)$ ($SigmoidSim_\delta(A, f_{c,d}(B))$ over all possible translations.

The key observation we make is that, although there is an infinite number of translations that we can apply to B, we have to consider only a finite number of translations if we want the $SigmoidSim$ to be within β of the optimal matching. Here we will simply give a sketch of the proof. The interested reader can look at [28]. The proof follows these steps:

- The matching problem is transformed into an equivalent stabbing problem, so each translation is transformed into a line of slope 1.
- The Sigmoid function is approximated by a constant number of segments on the stabbing plane. Therefore, using the $SigmoidSim$ we have an increased number of lines of slope 1 (translations) that we have to consider, but still the number of segments is multiplied by a constant factor. In this step we define an approximation factor α.
- The set of all possible translations is discovered.
- The translations for each axis are sorted. However, since adjacent translations only lead to LCSS computations that differ by at most 1, therefore we can efficiently *skip* some translations, according to a user specified error bound.

The running time is summarized in the following theorem:

Theorem 1 *Given two trajectories A and B, with* $|A| = n$ *and* $|B| = m$*, and a constant* $0 < \beta < 1$*, we can find an approximation* $AS2_{\delta,\beta}(A, B)$ *of the similarity* $S2(\delta, A, B)$ *such that* $S2(\delta, A, B) - AS2_{\delta,\beta}(A, B) < \alpha \cdot \beta$ *in* $O((m + n)\delta^3/\beta^2)$ *time, for a constant* α.

Therefore the approximate algorithm offers a dramatic speedup offered by considering a significantly smaller set of possible translations.

5 Experimental Evaluation

We implemented the proposed similarity functions and we compare them to various L_p Norms and also to the Time Warping using as the base distance different L_p Norms, too. We use lower than L_1 Norms, since for them the extreme values do not dominate to the total distance, therefore are more robust to outliers ([2]).

We conducted clustering experiments, using datasets obtained through video tracking, that consisted of various clusters of trajectories in 2D. The datasets are:

- **MobileLong**. This is a real dataset consisting of 5 classes of objects, 3 recordings for each cluster. We have their (x,y) position over time and the lengths of the trajectories vary from 800 points to almost 2000.
- **MobileShort**. Again a real-life spatiotemporal dataset. Now the average length of the sequences is around 100 points and there are 7 classes of objects having 5 recordings per class.
- **MobileShort+Noise**. Constructed from the previous dataset but adding noise at a random position of the trajectory and for 15% of the length of the trajectory. Essentially, the signal is considered non-recoverable during this noisy portion (figure 5).

We used the distance function as follows:

1. For the L_p Norms we slid the shorter of the two trajectories along the longer one and recorded the minimum distance.
2. For DTW we modified the original algorithm in order to match both x and y coordinates. In DTW and pNorms we normalized the data before computing the distances.
3. For our method we used the similarity function $S1$ on the normalized trajectories.

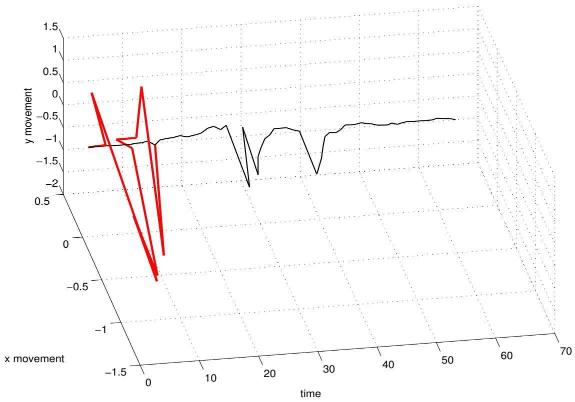

Figure 5. The $MobileShort$ after noise is added.

We performed the following experiment: We clustered hierarchically (using complete linkage) all the recordings for each pair of clusters. Then if at the second level the 2 classes are divided correctly we consider the clustering correct, otherwise not. So, for the $MobileLong$ dataset, we had a total of 5*(5-1)/2 = 10 experiments.

The results clearly depict that the new $SigmoidSim$ model is very robust to noise. In the $MobileLong$ dataset (figure 6) the use of the lower L_p Norms, does not improve the classification accuracy of the other distance functions. So, even though the effect of the outliers is less intense, however these methods have to match all elements (note that the lengths in this dataset very considerably), therefore distorting the total distance. However, our model can efficiently skip the erroneous sequence parts.

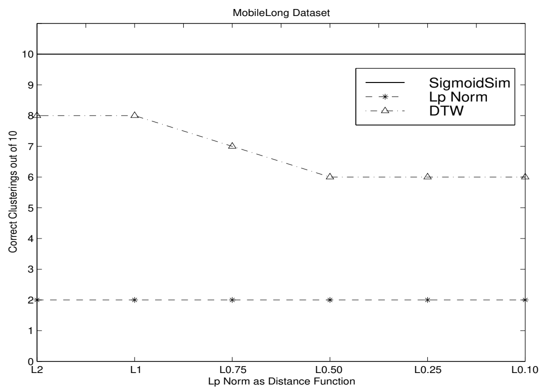

Figure 6. The number of correct clusterings (out of 10) in the $MobileLong$ dataset. The $SigmoidSim$ model has no errors at all.

For the $MobileShort$ dataset the use of the non-metric Norms improves the performance. Nonetheless, $SigmoidSim$ is consistently better (figure 7). Finally, the $MobileShort$ with the added noise shows the true potential of our approach. Under the strong presence of noise it is almost twice as accurate than the antagonist distance functions (figure 8) and as good as the $L_{0.1}$ Norm on the same dataset without any noise.

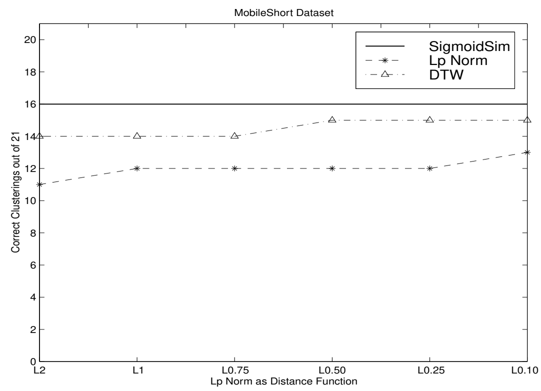

Figure 7. In this dataset the use of Lp Norms for $p \leq 1$ improves the performance.

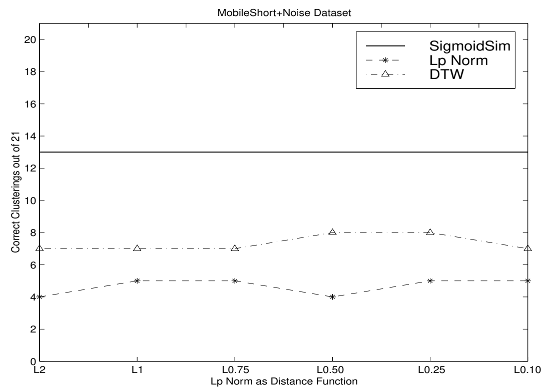

Figure 8. $SigmoidSim$ is obviously superior in the presence of noise.

6 Conclusion

In this paper we presented efficient techniques to accurately capture the similarity between trajectories with significant amount of noise. The similarity measure is based on the LCSS scheme augmented with a flexible sigmoid matching function. Our experiments indicate the clustering superiority of this approach, compared to various Lp-Norms and the Time Warping distance. In the future we plan to extend this model in order to be also rotation invariant. Moreover, of further theoretical importance would be the employment of an indexing structure for the new similarity model, since it is non-metric and does not obey the triangle inequality.

Acknowledgements

We would like to thank Margrit Betke for providing us with the $MobileLong$ video captured datasets, obtained with the Camera Mouse program.

References

[1] P. K. Agarwal, L. Arge, and J. Erickson. Indexing moving points. *In Proc. of the 19th ACM Symp. on Principles of Database Systems (PODS)*, pages 175–186, 2000.

[2] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *ICDT*, pages 420–434, 2001.

[3] R. Agrawal, C. Faloutsos, and A. Swami. Efficient Similarity Search in Sequence Databases. *In Proc. of the 4th FODO*, pages 69–84, Oct. 1993.

[4] R. Agrawal, K. Lin, H. S. Sawhney, and K. Shim. Fast Similarity Search in the Presence of Noise, Scaling and Translation in Time-Series Databases. *In Proc of VLDB*, pages 490–501, Sept. 1995.

[5] D. Barbara. Mobile computing and databases - a survey. *IEEE TKDE*, pages 108–117, Jan. 1999.

[6] D. Berndt and J. Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. *In Proc. of KDD Workshop*, 1994.

[7] B. Bollobas, G. Das, D. Gunopulos, and H. Mannila. Time-Series Similarity Problems and Well-Separated Geometric Sets. *In Proc of the 13th SCG, Nice, France*, 1997.

[8] T. Bozkaya, N. Yazdani, and M. Ozsoyoglu. Matching and Indexing Sequences of Different Lengths. *In Proc.of the CIKM, Las Vegas*, 1997.

[9] K. Chu and M. Wong. Fast Time-Series Searching with Scaling and Shifting. *ACM Principles of Database Systems*, pages 237–248, June 1999.

[10] G. Das, D. Gunopulos, and H. Mannila. Finding Similar Time Series. *In Proc. of the First PKDD Symp.*, pages 88–100, 1997.

[11] C. Faloutsos, H. Jagadish, A. Mendelzon, and T. Milo. Signature technique for similarity-based queries. *In SEQUENCES 97*, 1997.

[12] C. Faloutsos, M. Ranganathan, and I. Manolopoulos. Fast Subsequence Matching in Time Series Databases. *In Proceedings of ACM SIGMOD*, pages 419–429, May 1994.

[13] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. *In Proc. of 25th VLDB*, pages 518–529, 1999.

[14] D. Goldin and P. Kanellakis. On Similarity Queries for Time-Series Data. *In Proceedings of CP '95, Cassis, France*, Sept. 1995.

[15] H. V. Jagadish, A. O. Mendelzon, and T. Milo. Similarity-based queries. *In Proc. of the 14th ACM PODS*, pages 36–45, May 1995.

[16] T. Kahveci and A. K. Singh. Variable length queries for time series data. In *Proc. of IEEE ICDE*, pages 273–282, 2001.

[17] E. Keogh, K. Chakrabarti, S. Mehrotra, and M. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proc. of ACM SIGMOD*, pages 151–162, 2001.

[18] E. Keogh and M. Pazzani. Scaling up Dynamic Time Warping for Datamining Applications. *In Proc. 6th Int. Conf. on Knowledge Discovery and Data Mining, Boston, MA*, 2000.

[19] G. Kollios, D. Gunopulos, and V. Tsotras. On Indexing Mobile Objects. *In Proc. of the 18th ACM Symp. on Principles of Database Systems (PODS)*, pages 261–272, June 1999.

[20] S.-L. Lee, S.-J. Chun, D.-H. Kim, J.-H. Lee, and C.-W. Chung. Similarity Search for Multidimensional Data Sequences. *In Proceedings of ICDE*, pages 599–608, 2000.

[21] S. Park, W. Chu, J. Yoon, and C. Hsu. Efficient Searches for Similar Subsequences of Different Lengths in Sequence Databases. *In Proceedings of ICDE*, pages 23–32, 2000.

[22] S. Perng, H. Wang, S. Zhang, and D. S. Parker. Landmarks: a New Model for Similarity-based Pattern Querying in Time Series Databases. *In Proceedings of ICDE*, pages 33–42, 2000.

[23] D. Pfoser, C. Jensen, and Y. Theodoridis. Novel Approaches in Query Processing for Moving Objects. *In Proceedings of VLDB, Cairo Egypt*, Sept. 2000.

[24] Y. Qu, C. Wang, and X. Wang. Supporting Fast Search in Time Series for Movement Patterns in Multiple Scales. In *Proc of the ACM CIKM*, pages 251–258, 1998.

[25] D. Rafiei and A. Mendelzon. Querying Time Series Data Based on Similarity. *IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No 5.*, pages 675–693, 2000.

[26] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-26(1):43–49, Feb. 1978.

[27] S. Saltenis, C. Jensen, S. Leutenegger, and M. A. Lopez. Indexing the Positions of Continuously Moving Objects. *In Proceedings of the ACM SIGMOD*, pages 331–342, May 2000.

[28] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *ICDE*, 2002.

[29] B.-K. Yi and C. Faloutsos. Fast Time Sequence Indexing for Arbitrary Lp Norms. *In Proceedings of VLDB, Cairo Egypt*, Sept. 2000.