Managing Intellectual Capital
and Innovation for Sustainable
and Inclusive Society
27–29 May 2015
Bari • Italy

make learn TIIM

Management,
Knowledge and Learning
Joint International Conference 2015
Technology, Innovation
and Industrial Management

# A NEW METHOD FOR FINDING SIMILAR PATTERNS IN MOVING BODIES

Prateek Kulkarni
Goa College of Engineering, India
kvprateek@gmail.com

## Abstract:

An important consideration in similarity-based retrieval of moving object trajectories is the definition of a distance function. The shape based trajectory retrieval is gaining importance due many real time applications. The existing distance functions consider partial matching of the shape of trajectories while retrieving trajectories from database. In this paper we have proposed Angular distance measure as a new shape based distance functions to retrieve trajectories. All the existing distance functions are based on global similarity and there is need for local similarity since it identities sub-trajectory with maximum similarity locally. We have proposed T-BLAST, a local similarity algorithm, which takes into account sub-trajectory matching. Further we have pruned T-BLAST technique by reducing the length of trajectories using Minimum Description Length (MDL) principle. Experimental results show that our proposed Angular distance effectively captures the shape of the trajectories while matching trajectories. T-BLAST accurately and efficiently performs sub-trajectory matching with maximum local similarity. Also, the experimental results show that Pruned T-BLAST algorithm is more efficient compared to T-BLAST.

*Keywords: Shape-based, Sub-trajectory matching, T-BLAST*

## 1. INTRODUCTION

Now days it has become easy to track the moving objects due to rapid growth in mobile and communication technologies. Overall effort required to design such a tracking system is very minimal and cost effective. The moving objects are travelling from one place to another, in order to perform their daily routine work, thereby generating huge amount of data and there is need to analyse this data for determining the interesting patterns.

The similarity search of moving objects has been the main focus of current research due its vast application in day to day life. Few interesting applications of similarity search in the moving object trajectories are discuss here. Stock exchange keep track of huge amount of data and it is represented in the trajectories form. The user might be interested in searching for particular stock whose shape is same as that of query trajectory stock. In the football game the trajectories of the ball is stored in the database. The football coach can use the some of the best goal trajectories as query trajectories and retrieve similar trajectories from the database trajectories. This information is an important information for coach to understand the players skill and accordingly prepare his tack ticks.

Simple Example - Users concerned about being spied upon can use similarity search to identify the two similar trajectories in order to know whether or not they are being spied.

The tracked moving objects are represented in the trajectory form with its Global Positioning System (GPS) coordinates, i.e. latitude, longitude and altitude. The Trajectory of moving object is the path travelled by an object and it is either in 2D(x,y,t) or 3D(x,y,z,t) space where t is the time axis of moving object. The time attribute is continuously updated even though the object is in the stationary state.

We have restricted our dataset to two dimensional space without time axis i.e. (x,y) , since time axis is not playing important role in discovering interesting patterns in our problem statement.

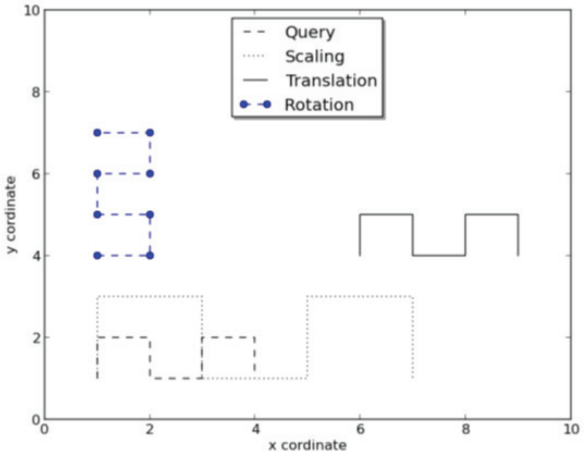
Considerable research has been conducted on similarity search of moving object trajectories (Chen, Zsu and Orias, 2005 and Vlachos, Kollios, and Gunopulos, 2002)... Unfortunately the existing distance function cannot consider following aspects of the trajectories while matching.

Case 1 : Consider two similar trajectories A and B, where shape of B is same as that of A but size of B may be bigger or smaller than that of A (See Figure1). Two scaled trajectories are compared and should return maximum similarity since both are same.

Case 2 : Consider two similar trajectories A and B, while trajectory B is translated by some distance with respect to trajectory A (See Figure 1). Two translated trajectories are being compared and should re- turn maximum similarity since both are same.

Case 3 : Consider two similar trajectories A and B, while trajectory B is rotated by some angle with respect to Trajectory A (See Figure 1). Two trajectories with different orientation are being compared and should return maximum similarity since both are same.

**Figure 1:** Trajectories with Scaling, Translation, Rotation



Case 4: Consider two trajectories A and B of length m and n respectively where n _ m and B is query trajectory. The two trajectories A and B may not to be similar over entire length i.e. globally similar. But there might be portions of database trajectory A with maximum similarity with query trajectory B i.e. locally similar. This is also called as partial or sub- trajectory matching.

Case 5: When the object is moving, it might happen that it remains at one place for some time or it might move in straight line or slight deviation in its movement. For the first three cases, the existing distance functions (Chen, Zsu and Orias, 2005; Vlachos, Kollios, and Gunopulos, 2002; Chen and Ng, 2004; Yi, Jagadish and Faloutsos, 1998 and Omnia Ossama and Mokhtar, 2009) fail to generate output since it is based on distance criteria and not on shape of trajectory.

In case 4, the existing distance functions fail to perform partial similarity matching between two trajectories ,since they perform only global matching , while local matching is not supported by those distance functions. The redundant points of the movement of trajectories are removed as discussed in case 5. To best of our knowledge, such a removal of redundant points has not been considered so far.

The main contributions of this paper are the following:-

1. A new distance measure called Angular distance (AD) is defined to solve the problem of scaling, translation and rotation. This new distance measure compares two trajectories by shape.
2. We have developed a technique to search the trajectories based on local and global similarity search.
3. We have developed optimization technique to reduce the length of trajectory using Minimum Description Principle. MDL partitions the trajectory without affecting the shape and uses distance function to calculate cost between the segment (Chen, Zsu and Orias, 2005). This will help in improving performance of the overall proposed technique.

| Symbols | Description |
|---|---|
| Rest(A) | Rest function returns trajectory except first element |
| ang(p1p2p3) | Angle between line segment p1p2 and p2p3 |
| $q_w$ | Query trajectory of window size w |

The rest of paper is arranged as follows: we give brief review of existing distance function in section 2. Section 3 presents new distance function Angular distance, T-BLAST algorithm and pruning of T-BLAST using MDL. Section 4 consists of experimental study of Angular distance, T-BLAST algorithm and pruned T-BLAST technique. Section 5 contain conclusion.

## 2. RELATED WORK

The existing distance measures such as DTW, LCSS, ERP, EDR, Euclidean distance (Chen, Zsu and Orias, 2005; Vlachos, Kollios, and Gunopulos, 2002; Chen and Ng, 2004; Yi, Jagadish and Faloutsos, 1998 and Omnia Ossama and Mokhtar, 2009) purely based on distance based approach. DTW, ERP, LCSS, EDR support local time shifting where as ERP satisfies metric property. Out of all distance function LCSS is consider to be more robust to noise. ERP uses gap penalty to align the two trajectories, while EDR is quantizing the distance between a pair of elements to two values 0 and 1.

Vlachos, Kollios, and Gunopulos (2002) uses distance measure with translation for similar trajectory matching. This distance measure accept translated distance as input from user as one of the parameter. For user it is very difficult to guess the translated distance value since trajectory data is huge and user is having less knowledge about database trajectories.

Altschul et al. (1990) uses the local alignment technique to match two biological sequences. In Altschul et al.(1990) BLAST algorithm is discussed which compares biological sequences for maximum similarity localy. As stated in (Altschul et al., 1990), BLAST is most efficient algorithm.

Lee, Han and Whang (2007) discusses the minimum description principle for partitioning trajectory. The trajectory partitioning is carried out in such a way that, shape of trajectory is unaltered.

# 3. SHAPE BASED SIMILARITY SEARCH OF TRAJECTORIES

## 3.1 Angular distance measure

The trajectory is represented using points in 2D(x,y) space and these points are joined to get the line segments. The angle between each consecutive line segements is computed using vector dot product. For example consider a trajectory T=p1(1,1), p2(2,1), p3(2,2),p4(2,1),p5(3,1). The points p1,p2, p3, p4, and p5 are joined in sequence to get line segment 1,2,3,4. The angle between line segment 1 and 2 is ang(p1p2p3)=90, segment 2 and 3 is ang(p2p3p4)=90 , segment 3 and 4 is ang(p3p4p5)=90. The trajectory T with angle between consecutive line segments in sequence is represented as T1(θ)=90, T2(θ)=90, T3(θ)=90.

Definition 1: The two trajectories R and S represented as R=r1(θ),r2(θ),r3(θ),...,rn(θ) and S=s1(θ),s2(θ),s3(θ),...,sn(θ) respectively. The two trajectories R and S, are said to match (match($r_i$,$s_j$)=true) if and only if ($r_i$(θ)-$s_j$(θ)) ≤ ε,where ε is the matching threshold.

Definition 2: Given two trajectories R and S of length m and n, respectively, the Angular distance(AD) between R and S is the minimum angle difference needed to change R to S or S to R. The definition of Angular distance(AD) is given below.

$$AD(R,S) = \begin{cases} m \\ n \\ \min\,(AD\big(Rest(R), Rest(S) + subcost, AD(Rest(R),S) + 1, AD\big(R, Rest(S)\big) + 1\big) \\ otherwise \end{cases} \cdots (1)$$

where subcost = 0 if match(r1,s1)=true and subcost=1 otherwise. Here we have assumed that cost of match is 0 and mismatch is 1. Angular distance measure matches the trajectories correctly and accurately even though they are scaled, translated and rotated. This measure is a shape based measure since it considers shape of trajectory using angle. Existing distance measures are basically based on Euclidean distance and hence they cannot capture shape of trajectory while matching.

Example: Given two trajectories A = p1(1,1), p2(2,1), p3(2,2), p4(2,1), p5(3,1) and B = p1(4,3), p2(4,4), p3(5,4), p4(5,3), p5(6,3) and minimum threshold value = 10 respectively. The trajectory A and B are processed to compute angle between consecutive line segments and is represented as A=a1(θ)=90, a2(θ)=90 ,a3(θ)=90 and B=b1(θ)=90, b2(θ)=90, b3(θ)=90. AD(A,B)=min(AD( Rest(A), Rest(B))+subcost, AD( Rest(A), B) +1, AD(A, Rest(B)) +1). First parameter of Min function, compare the angle of a1 and b1 and if it is matched, then subcost=0. Here it is matched and subcost=0. The parameter 2 and 3 make a calls to AD function with 1 is added to distance of mismatch value. The parameter 2 call AD function with trajectory Rest (A) and B , while parameter 3 calls AD function with trajectory A and Rest(B). For trajectory A and B, the first parameter of Min function returns minimum distance and is equal to zero since there is a match for (a1,b1), (a2,b2) ,(a3,b3), whereas argument 2 and 3 returns maximum value due to its mismatch.

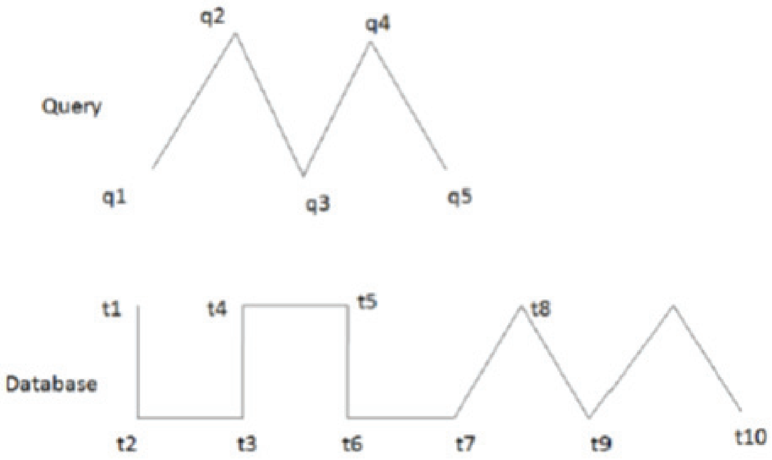
## 3.2 Tblast: trajectory basic local alignment search

Basic Local Alignment Search Tool (BLAST) is used for similar search of biological sequence. BLAST is a local similarity search which identifies sub-sequence with maximum similarity locally. We have applied BLAST basic principle for searching similar trajectories.

T-BLAST is based on identifying local sub-trajectory which has maximum similarity while comparing two trajectories. This method is also known as sub-trajectory Matching.

We first define was window size for query trajectory. In this paper, we take window size equal to 3, since it is taken in as 3 in Vlachos, Kollios, and Gunopulos (2002). Take window query and match it with window database trajectory from starting position and move from left to right of database trajectory. If window query is matched with database trajectory, then compare full query trajectory with database trajectory. If full query and database trajectory are matched for given minimum threshold value, then output the database trajectory as matched trajectory.
Figure 2 show an example of sub-trajectory matching. T-BLAST detail algorithm is given below.

**Figure 2:** Sub-trajectory matching



T-BLAST Algorithm
Input: Query Trajectory-Q , Database Trajectory - D ,Window size w=3, Minimum Threshold ε
Output: Matched Trajectories – MT

```
$Initialise\ MT = \varphi$
$for\ each\ q_w \in Q\ do$
  $for\ each\ t \in T\ do$
    $for\ each\ t_w \in T\ do$
      $if (Score(q_w, t_w) \leq \epsilon)\ then$
        $if (Score(Q, t) \leq \epsilon)\ then$
          $MT = MT \cup t$
        endif
      endif
```

Score Function (t1,t2)

```
$if (t1.angle - t2.angle) \leq \epsilon$
  $value = 1$
$else$
  $value = 0$
```

### 3.3 Pruning of Tblast: using MDL

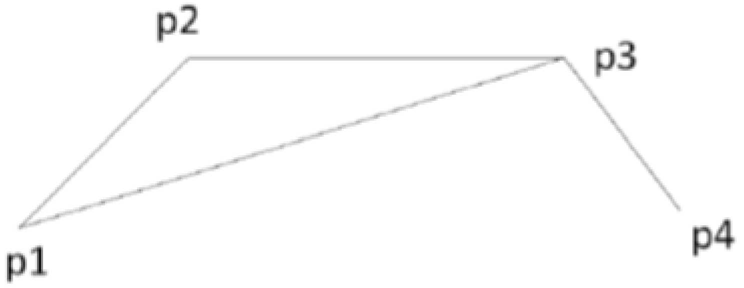
The length of the database trajectory is playing important role while computing the execution time of T-BLAST algorithm. The query trajectory will be always smaller compared to database and there is need to prune the length of database trajectory. The path travelled by the moving object might be straight or object might remain at same position for some time or slight deviation in the normal path. This lead to redundant points in the trajectory and can be removed without affecting the shape of the trajectory. Elimination of such points from trajectory will reduce the length of trajectory, which in turn reduce the execution time of the algorithm.

The trajectory is partitioned based on the cost of partition and non partition. The partition cost is computed using perpendicular distance and angular distance, whereas non partition cost is calculated using Euclidean distance.

Perpendicular and angular distances are calculated using following formulae.

$$d_{\perp}(L_i, L_j) = \frac{l_{\perp 1}^2 + l_{\perp 2}^2}{l_{\perp 1} + l_{\perp 2}} \quad \dots (2)$$

**Figure 3:** Trajectory partition using MDL



$$d_\theta(L_i, L_j) = \begin{cases} \|L_j\| \times \sin\theta & if\ 0 \le \theta \le 90 \\ \|L_j\| & if\ 90 \le \theta \le 180 \end{cases} \qquad (3)$$

The cost of Non-partition is calculated using the Euclidean distance as given below,

$$L1 = \sum_{j=1}^{par_i - 1} \log_2(len(pc_j, pc_{j+1}) \qquad (4)$$

Where len is the function for calculating the distance between two points $c_j$ and $c_{j+1}$
The cost of partition is computed using perpendicular and angular distance as given below:

$$L2 = \sum_{j=1}^{par_i - 1} \sum_{k=c_j}^{c_{j+1}-1} \left(\log_2(d_\perp(pc_jpc_{j+1}, p_kp_{k+1}))\right) + (\log_2(d_\theta(pc_jpc_{j+1}, p_kp_{k+1}))) \qquad (5)$$

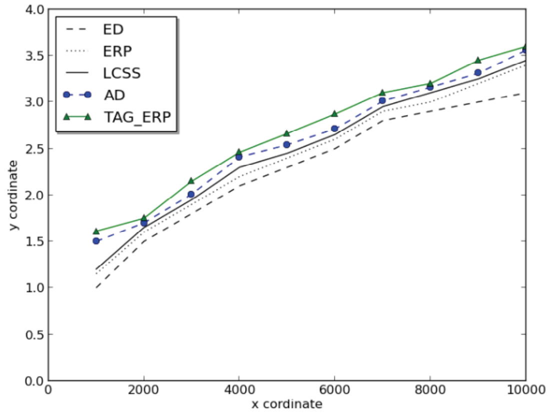
The trajectory is partition when the cost of partition is greater than the cost of non partition. Consider the example shown in Figure 3. The cost of partition of p1p4 is greater than non partition; hence the p3 point is selected as partition point.

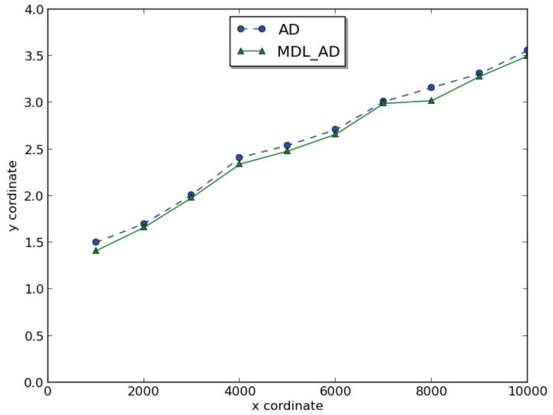
## 4. EXPERIMENTAL STUDY

In the experimental evaluation, we choose Network generator a moving object dataset generator developed by Brinkhoff (2003). Using this generator, we simulate two dimensional trajectories of vehicles on the road network in the city of San-Francisco.

The number of trajectories varies from 1000,2000,: : : ,10,000. When evaluating performance, we use CPU time as the standard because all dataset sizes are small enough to put into memory. All experiments are run on a linux PC with AMD Athlon 900MHz CPU, 2GB memory, and 100GB hard disk. C++ program were written to implement our proposed Angular distance measure and T-BLAST algorithm.

The experimental results shows that T-BLAST outperform TAG-ERP where as Euclidean distance, ERP, EDR, LCSS, DTW are much faster than T-BLAST. T-BLAST is accurate enough to perform sub-trajectory matching accurately whereas Euclidean distance, ERP, EDR, LCSS, DTW fails. The Pruned T-BLAST algorithm is much faster compared to T-BLAST.





In this paper we have proposed new Angular distance measure for matching of similar trajectories based on shape. Also we proposed T-BLAST local similarity search technique for sub-trajectory matching. Further T-BLAST was pruned using MDL principle. Angular distance measure effectively captured shape of trajectories while matching trajectories. T-BLAST accurately and efficiently performs sub-trajectory matching of trajectories and pruned T-BLAST was faster as compared to T-BLAST technique.

## REFERENCE LIST

1. Altschul, S.F., W. Gish, W. Miller, E. W. Myers, D. J. Lipman. (1990). Basic local alignment search tool. *Journal of Molecular Biology*,215(3):403-10.
2. Brinkhoff, T. (2003). Generating traffic data. Bulletin of the Technical Committee on Data Engineering, *IEEE Computer Society*, 26:1925.
3. Chen, L. and R. Ng. (2004). *On the marriage of lp-norms and edit distance*. In VLDB 04: Proceedings of the Thirtieth international conference on Very large data bases, pages 792803, Toronto, Canada.
4. Chen, L., M. T. O Zsu, and V. Oria. (2005). *Robust and fast similarity search for moving object trajectories*. In SIGMOD 05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pages 491502, Baltimore, Maryland.
5. Jae-Gil L., J. Han, K. Y. Whang. (2007). *Trajectory Clustering: A Partition-and-Group Framework*.SIGMOD07, Beijing, China. June 11-14.
6. Omnia Ossama, H., M. O. Mokhtar. (2009). *Similarity Search in Moving Object Trajectories*.15th International Conference on Management of Data COMAD 2009, Mysore, India, December 9-12.
7. Vlachos, M., G. Kollios, and D. Gunopulos. (2002). *Discovering similar multidimensional trajecto*ries. In Proc. 18th Int. Conf. on Data Engineering, pages 673-684.
8. Yi, B. K., H. Jagadish, and C. Faloutsos. (1998*). E_cient retrieval of similar time sequences under time warping.* In Proc. 14th Int. Conf. on Data Engineering, pages 23-27.