# Mining Trajectory Corridors Using Fréchet Distance and Meshing Grids

Haohan Zhu[1,2], Jun Luo[2], Hang Yin[3], Xiaotao Zhou[2], Joshua Zhexue Huang[2], and F. Benjamin Zhan[4]

[1] Institute of Computing Technology, Chinese Academy of Sciences
[2] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
{hh.zhu,xt.zhou,zx.huang}@siat.ac.cn, {jun.luo,hang.yin}@sub.siat.ac.cn
[3] University of Science and Technology of China
[4] Texas Center for Geographic Information Science, Department of Geography, Texas State University-San Marcos
zhan@txstate.edu

**Abstract.** With technology advancement and increasing popularity of location-aware devices, trajectory data are ubiquitous in the real world. Trajectory corridor, as one of the moving patterns, is composed of concatenated sub-trajectory clusters which help analyze the behaviors of moving objects. In this paper we adopt a three-phase approach to discover trajectory corridors using Fréchet distance as a dissimilarity measurement. First, trajectories are segmented into sub-trajectories using meshing-grids. In the second phase, a hierarchical method is utilized to cluster intra-grid sub-trajectories for each grid cell. Finally, local clusters in each single grid cell are concatenated to construct trajectory corridors. By utilizing a grid structure, the segmentation and concatenation need only single traversing of trajectories or grid cells. Experiments demonstrate that the unsupervised algorithm correctly discovers trajectory corridors from the real trajectory data. The trajectory corridors using Fréchet distance with temporal information are different from those having only spatial information. By choosing an appropriate grid size, the computing time could be reduced significantly because the number of sub-trajectories in a single grid cell is a dominant factor influencing the speed of the algorithms.

**Keywords:** Trajectory, Fréchet Distance, Meshing Grids.

## 1 Introduction

With the technology progress and popularity of location-aware devices, vast data of moving objects have been collected. Trajectory data are ubiquitous in the real world including tropical cyclones data, transportation system data, flocking sheep data, migrating birds data, to name a few. Consequently, trajectory analysis has become a pragmatic tool to discover moving objects patterns. Trajectory corridor, through which the moving objects frequently pass, is one of the

M.J. Zaki et al. (Eds.): PAKDD 2010, Part I, LNAI 6118, pp. 228–237, 2010.

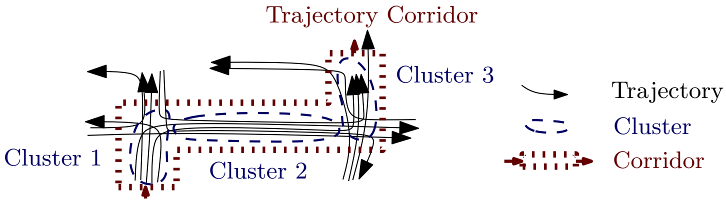


**Fig. 1.** An example of trajectories, clusters and corridors

moving patterns. In this paper, we address the trajectory corridor as concatenated sub-trajectory clusters which could help identify and predict the behaviors of the moving objects. An example to illustrate trajectories, local clusters and trajectory corridors is shown in Fig.1.

Because not only may a trajectory belong to multiple trajectory corridors simultaneously, but trajectory corridors are also composed by different groups of trajectories at different parts. In this paper, we adopt a three-phase approach to mine the trajectory corridors using Fréchet distance and meshing grids. Firstly, trajectories are segmented into sub-trajectories according to the meshing-grids. In the second phase, a hierarchical method is utilized in each grid cell separately to cluster intra-grid sub-trajectories. Finally, the local clusters in each single grid cell are concatenated to construct trajectory corridors.

Summarizing, the contributions presented in this paper are:

– We introduce discrete Fréchet distance as a novel dissimilarity measurement between trajectory clustering because it is generally regarded as a more appropriate distance function for polygonal curves and could easily consider not only shapes, relative positions and orientations, but also velocities of trajectories.
– We propose a meshing grid structure. By utilizing grid structure, the segmentation and concatenation need only single traversing of trajectories or grid cells. When choosing appropriate grid size, the computing time could be reduced significantly since the dominant factor of the computing time is the amount of sub-trajectories in a single grid cell.

## 2 Related Work

Gaffney and Smyth[7] propose a method of trajectory clustering using mixture of regression models. Nanni and Pedreschi[10] propose a density-based trajectory clustering algorithm based on OPTICS[2]. In the above work, the object to be clustered is the whole trajectory, namely, one trajectory can be only in one cluster. Thus, trajectory corridors are not their objective.

In the trajectory clustering algorithm proposed by Lee et al.[9], trajectories are partitioned into a set of line segments which are clustered by using DBSCAN[6] algorithm. However, because Fréchet distance could handle polygonal curves well, we need not to simplify the trajectories into line segments but only cut off trajectories into several short polygonal curves. And we adopt a hierarchical method for clustering to avoid distance accumulation discussed in Section 5.2.

## 3 Preliminary Definition

In reality, a moving object trajectory is a finite set of observations at discrete time stamps $t_1$, $t_2$, $\cdots$, $t_n$. It could be assumed that an object moves between two consecutive time steps on a straight line and the velocity is constant along the segment. Hence we define the trajectory $\tau$ as a polygonal line with $n$ vertices having time stamps.

**Definition 1 (Trajectory).** $\tau = \langle (t_1, p_1), (t_2, p_2), \cdots, (t_n, p_n) \rangle$, *where* $p_i \in \mathbb{R}^d$.

The space is partitioned by meshing-grids. Every grid cell $G_j$ has an id $j$ to label it. The sub-trajectory is recorded in $\tau'_{i,j,mark}$ where $i$ represents the original trajectory $\tau_i$ it belongs to, $j$ represents the grid cell $G_j$ it falls into and $mark$ is the mark to differentiate the different part of the same trajectory in the same grid. The definition of sub-trajectories is the same as that of trajectories.

**Definition 2 (Local Cluster in Grid Cell $G_j$).** $C_j = \langle\ \tau'_{i_1,j,mark_1}, \tau'_{i_2,j,mark_2}, \cdots, \tau'_{i_m,j,mark_m}\ \rangle$, *where* $\tau'_{i_k,j,mark_k}$ $(1 \leq k \leq m)$ *is a sub-trajectory in grid cell* $G_j$*.*

The local cluster in a certain grid cell is a set of sub-trajectories in that cell, so the cluster has no shape or range. However, the cluster has its own origin and destination. The position and velocity at origin and destination of a cluster are average values of sub-trajectories in that cluster.

**Definition 3 (Trajectory Corridor).** $TC = \langle\ C_{j_1}, C_{j_2}, \cdots, C_{j_l}\ \rangle$, *where* $C_{j_i}$ $(1 \leq i \leq l)$ *is a local cluster in grid cell* $G_{j_i}$*, and the consecutive local clusters* $C_{j_i}$ *and* $C_{j_{i+1}}$ *need to follow concatenating criteria discussed in Section 5.3.*

The trajectory corridor is a sequence of local clusters and the order of the sequence indicates the direction of the trajectory corridor. Every trajectory corridor is composed of either one local cluster or more. Moreover, different trajectory corridors may share the same local cluster.

## 4 Computing Fréchet Distance

Distance function is the key component of mining trajectory corridors because dissimilarity measurement is necessary to group similar trajectories. The Fréchet distance is a measurement for curves and surfaces. It is defined using reparameterizations of the shapes. The Fréchet distance is generally regarded as being a more appropriate distance function than the Hausdorff distance or other distances for curves[1] because it takes the continuity of the shapes into account.

Due to high complexity of computing Fréchet distance, the discrete Fréchet distance as a natural variant for polygonal curves is more proper for computing. Eiter and Mannila[5] proposed a dynamic programming algorithm to compute discrete Fréchet distance in $O(mn)$ time.

Consider two polygonal curves $P$, $Q$ in $\mathbb{R}^d$ given by the sequences of their vertices $\langle p_1, \cdots, p_n\rangle$ and $\langle q_1, \cdots, q_m\rangle$, respectively. Computing discrete Fréchet distance only uses coupling $C$ which is a sequence of pairs of vertices $C{=}\langle C_1, \cdots, C_k\rangle$ with $C_r{=}(p_i, q_j)$ for all $r{=}1, \ldots, k$ and some $i{\in}\{1, \ldots, n\}$, $j{\in}\{1, \ldots, m\}$, fulfilling

- $C_1{=}(p_1, q_1)$ and $C_k{=}(p_n, q_m)$
- $C_r{=}(p_i, q_j) \Rightarrow C_{r+1}{\in}\{(p_{i+1}, q_j), (p_i, q_{j+1}), (p_{i+1}, q_{j+1})\}$ for $r{=}1, \ldots, k-1$

**Definition 4 (Discrete Fréchet Distance[3]).** *Let $P$, $Q$ be two polygonal curves in $\mathbb{R}^d$, and let $|\cdot|$ denote an underlying norm in $\mathbb{R}^d$. Then the Discrete Fréchet Distance $\delta_{dF}(P,Q)$ is defined as*

$$\delta_{dF}(P,Q) = \min_{coupling C} \max_{(p_i,q_j)\in C} |p_i - q_j|$$

*where $C$ ranges over all couplings of the vertices of $P$ and $Q$.*

If the distance computation $|\cdot|$ between vertices ignores velocities of trajectories, the distance function is shape dependent only like DTW[8], LCSS[11], EDR[4] and so on. However, the two trajectories may have similar shapes, but actually they represent totally different moving patterns when considering velocities illustrated in Fig.2. Yanagisawa et al.[12] propose a measurement combined DTW distance with Euclidean distance which considers both velocities and shapes of moving objects, whereas the distance is sensitive to the grouping parameter $\mu$ and they require time duration of different trajectories to be the same length. Hence another merit brought by Fréchet distance is that it could easily take account of velocities. By substituting $|\cdot|$ between vertices, not only spatial information of trajectories but also temporal information of trajectories can be considered. In our paper, a simply solution is provided that the distance $|\cdot|$ between two vertices in two dimensions is defined as $\sqrt{\omega_s(\Delta x^2 + \Delta y^2) + \omega_t \Delta v^2}$, and weights $\omega_s$ and $\omega_t$ differentiate the effects of spatial properties and temporal

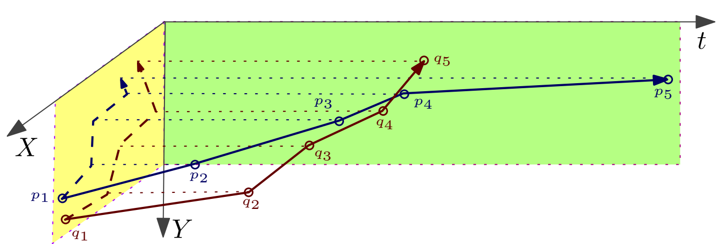


**Fig. 2.** The trajectories with similar shapes but different velocities

properties. $v$ is the instant velocity at the vertex. If $\omega_t = 0$, $|\cdot|$ is translated into Euclidean norm. The experiments in Section 6 illustrates the trajectory corridors considering velocities are quite different from those using $\omega_t = 0$. Actually, adjusting weights to make spatial and temporal properties equally important is empirical and influenced by spatial and temporal units.

## 5 Mining Trajectory Corridors

The procedure of mining trajectory corridors which is composed of three phases is illustrated in Fig.3. In the first phase, trajectories are segmented into sub-trajectories according to the meshing-grids. In the second phase, the hierarchical clustering algorithm based on discrete Fréchet distance is implemented in each grid cell. Finally, the local clusters in abutting grid cells are concatenated according the concatenation criteria to discover trajectory corridors.

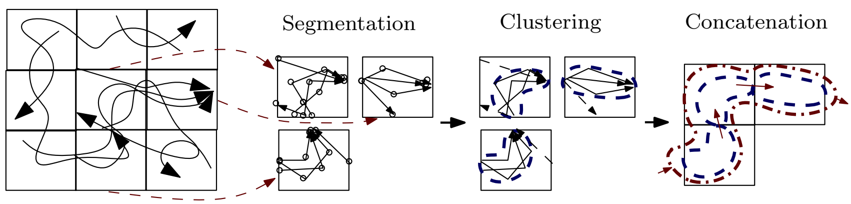


**Fig. 3.** An example of mining trajectory corridors

In this paper, we usee meshing grids structure to segment trajectories and concatenate local clusters because by utilizing grids, the segmentation and concatenation need only single traversing of trajectories or grid cells. Furthermore when changing grid size appropriately, the computing time could be reduced significantly since the dominant factor of the computing time is the amount of sub-trajectories in a single grid cell. This advantage is theoretically and experimentally analyzed and discussed in Section 6.

### 5.1 Segmentation

Since the different parts of a certain trajectory may belong to different trajectory corridors, segmenting trajectory into sub-trajectories is indispensable. In the process of segmentation, the size of grid cells are assigned in advance. As illustrated in Fig.4, when traversing each vertex in a trajectory, no segmentation will be executed until consecutive vertices pair $(p_i, p_{i+1})$ are not in the same grid cell. The intersections between the line segment $p_ip_{i+1}$ and edges of grid cells are computed. The trajectory is partitioned at each intersection. After segmentation, only the sub-trajectories in those grid cells which potentially contain local clusters will be preserved for the next phase. This process may reduce computing time dramatically when many grid cells include sparse sub-trajectories. The algorithm of segmentation costs $O(n)$ time, where $n$ is the sum of the vertices of all trajectories.

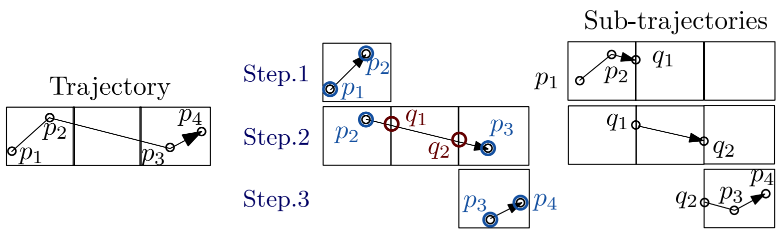


**Fig. 4.** Illustration of Segmentation

### 5.2 Intra-grid Sub-trajectory Hierarchical Clustering

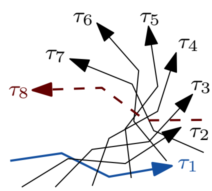

**Fig. 5.** Illustration of distance accumulation

In this paper, we adopt an agglomerative hierarchical clustering method and use two cluster distances $d_{min}$ and $d_{max}$ at the same time to avoid distance accumulation illustrated in Fig.5. In the example, $\tau_1$ and $\tau_8$ are almost in the opposite directions but may be merged into the same cluster because each pair of nearby trajectories $\tau_i$ and $\tau_{i+1}$ has a small distance. Two cluster distances $d_{min}$ and $d_{max}$ are defined as follow: $d_{min}(C_i, C_j) = min_d(p, q)$, $d_{max}(C_i, C_j) = max_d(p, q)$, where $p \in C_i, q \in C_j$, $d(p, q)$ is modified discrete Fréchet distance between $p, q$. In this phase, the computing is only executed in the grid cells that have sufficient sub-trajectories. For each hierarchy, the nearest clusters are merged according to $d_{min}$. Namely, two clusters are merged when they have the minimal $d_{min}$. However, while $d_{max}$ between the nearest clusters exceeds an certain threshold, no merging executes and the algorithm continues to the next hierarchy. Until the minimal $d_{min}$ exceeds an termination condition, clustering ceases. Finally, the local clusters which do not have sufficient sub-trajectories are pruned. The algorithm for each grid can be computed in $O(n^2l^2+n^2 \log n+n^2m^2)$ time, where $n$ is the amount of sub-trajectories, $l$ is the amount of vertices of one sub-trajectory, $m$ is the amount of sub-trajectories in one cluster. An example of intra-grid sub-trajectory clustering using distance matrix and index is illustrated in Fig.6.

### 5.3 Inter-grid Concatenation

In this phase, we propose an algorithm of concatenating local clusters to discover trajectory corridors. The concatenation criteria is defined as follow: When positions and velocities between the origin of one cluster and the destination of the other cluster in the adjacent grid cell are similar, we call the two local clusters are connectable. And we consider concatenation that does not require the adjacent local clusters sharing the sufficient same trajectories or even same amount of trajectories. Trajectory corridors are continuous channels with directions that moving objects frequently visit but enter and leave freely.

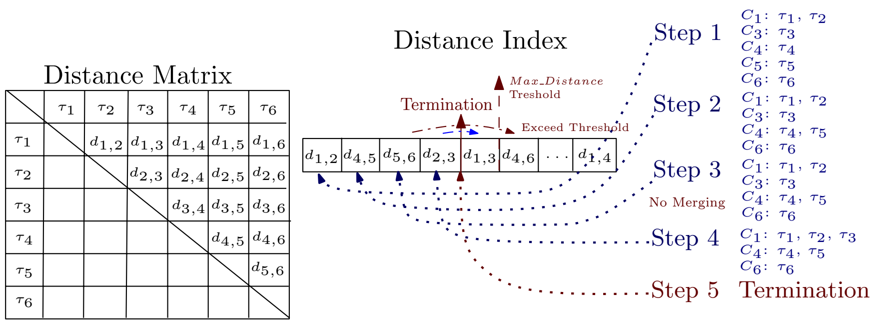


**Fig. 6.** Illustration of intra-grid sub-trajectory clustering using distance matrix and index

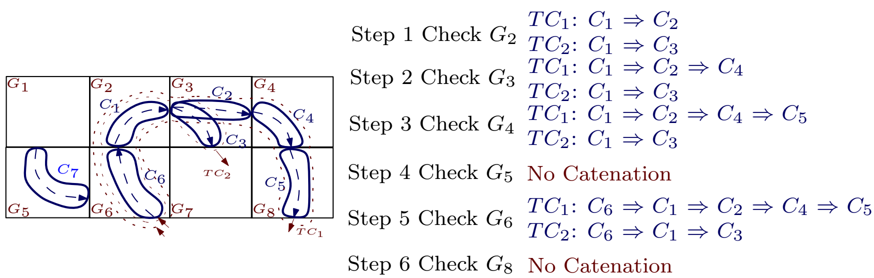


**Fig. 7.** Illustration of inter-grid concatenation

In the phase of local clusters concatenation, traversing all grid cells that have local clusters only once could find all possible trajectory corridors. In each step, we check one grid cell $G_j$ and the local clusters in it. If there exists local clusters $C_q$ in neighbor cells having the origins connectable with the destinations of the local clusters $C_p$ in $G_j$, all trajectory corridors including $C_p$ are catenated to all trajectory corridors including $C_q$. This approach can be computed in $O(nk)$ time, where $n$ is the amount of local clusters and $k$ is the amount of trajectory corridors. An example of concatenation is illustrated in Fig.7 and trajectory corridors with only one cluster are hidden.

## 6 Experiments

In this paper, all experiments were conducted by using the tropical cyclone best track data set[1]. And for all experiments, parameters including pruning criteria, termination condition and concatenation criteria are constant.

Fig.8(a) is the result of clustering which considers both spatial and temporal similarity, whereas, Fig.8(b) is the result of clustering which ignores the velocity.

[1] http://www.jma.go.jp/jma/jma-eng/jma-center/rsmc-hp-pub-eg/besttrack.html

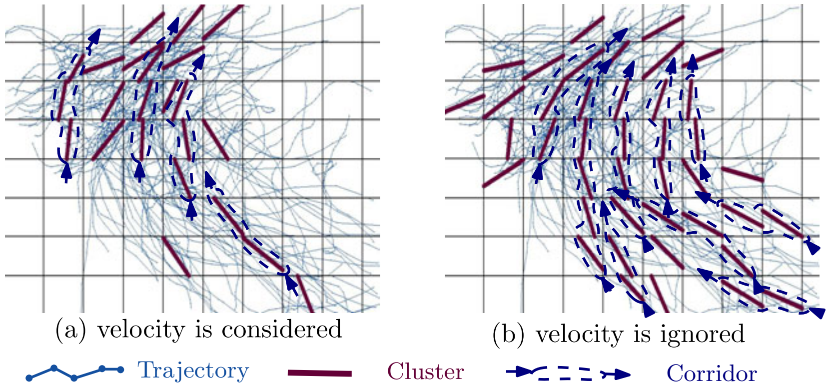


**Fig. 8.** Trajectory corridors considering velocity or not

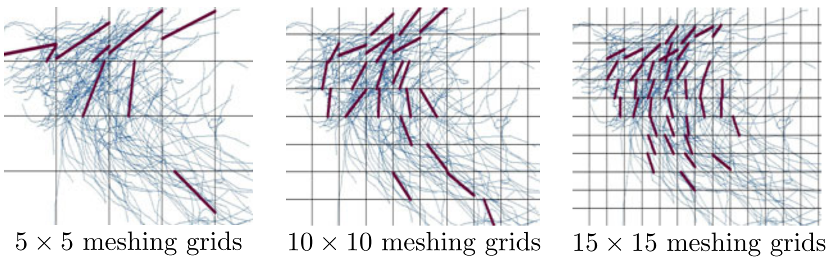


**Fig. 9.** Trajectory corridors in different grid sizes

The experiments successfully demonstrate that by considering temporal properties, the sub-trajectory clusters are different from those that have only spatial properties.

From Fig.9, it is obvious that more grid cells produce more sub-trajectories, local clusters and trajectory corridors. However, the computing time is reduced significantly, when the amount of sub-trajectories per cell decreases from 21 to 5.(see Fig.10(a)). Since building index runtime is the dominant factor in the overall runtime illustrated in Fig.10(b), the overall running time could be approximated to $O(kn^2 \log n)$, where $n$ is the amount of sub-trajectories in one grid cell and $k$ is the amount of grid cells. Thus when choosing appropriate grid size, the computing time could be reduced, because the amount of sub-trajectories per cell may decrease. The grid size may affect both computing time and clustering quality. The quality of the results may decrease when grid size is larger due to more noises. So to keep the quality of the clustering results and to reduce computing time requires a trade-off which vary from data to dada.

The experiments successfully validate our algorithm to discover trajectory corridors. The tropical cyclones in Western North Pacific and the South China Sea often have parts of their trajectories in such a corridor: start from the position

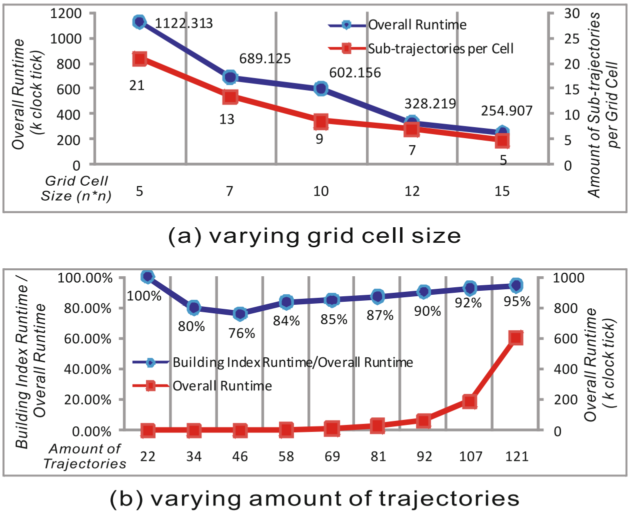


**Fig. 10.** Runtime in different meshing-grid sizes

around 135 °E−150 °E, 10 °N, move towards northwestern to the location about 120 °E−135 °E, 25 °N, then towards northeastern and finally end approximately at 145 °E, 40 °N.

## 7 Conclusions

In this paper, we adopt a three-phase approach to discover trajectory corridors using Fréchet distance as a dissimilarity measurement. The experiments successfully discovered tropical cyclone corridors by segmenting, clustering and concatenating various components of a trajectory. The trajectory corridors discovered by using Fréchet distance with temporal information may be more significant. The use of meshing grid structure could reduce computing time effectively by choosing an appropriate grid size.

## References

1. Alt, H., Knauer, C., Wenk, C.: Comparison of distance measures for planar curves. Algorithmica 38(1), 45–58 (2003)
2. Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J.: Optics: Ordering points to identify the clustering structure. In: SIGMOD Conference, pp. 49–60 (1999)
3. Buchin, K., Buchin, M., Gudmundsson, J., Löffler, M., Luo, J.: Detecting commuting patterns by clustering subtrajectories. In: Hong, S.-H., Nagamochi, H., Fukunaga, T. (eds.) ISAAC 2008. LNCS, vol. 5369, pp. 644–655. Springer, Heidelberg (2008)
4. Chen, L., Tamer Özsu, M., Oria, V.: Robust and fast similarity search for moving object trajectories. In: SIGMOD Conference, pp. 491–502 (2005)

5. Eiter, T., Mannila, H.: Computing discrete fréchet distance. Technical Report CD-TR, 94(64) (1994)
6. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231 (1996)
7. Gaffney, S., Smyth, P.: Trajectory clustering with mixtures of regression models. In: KDD, pp. 63–72 (1999)
8. Keogh, E.J.: Exact indexing of dynamic time warping. In: VLDB, pp. 406–417 (2002)
9. Lee, J.-G., Han, J., Whang, K.-Y.: Trajectory clustering: a partition-and-group framework. In: SIGMOD Conference, pp. 593–604 (2007)
10. Nanni, M., Pedreschi, D.: Time-focused clustering of trajectories of moving objects. J. Intell. Inf. Syst. 27(3), 267–289 (2006)
11. Vlachos, M., Gunopulos, D., Kollios, G.: Discovering similar multidimensional trajectories. In: ICDE, pp. 673–684 (2002)
12. Yanagisawa, Y., Satoh, T.: Clustering multidimensional trajectories based on shape and velocity. In: ICDE Workshops, p. 12 (2006)