

Review Article

An Overview of Moving Object Trajectory Compression Algorithms

Penghui Sun, Shixiong Xia, Guan Yuan, and Daxing Li

School of Computer Science and Technology, China University of Mining and Technology, No. 1, College Road, Xuzhou, Jiangsu 221116, China

Correspondence should be addressed to Shixiong Xia; xiasx@cumt.edu.cn and Guan Yuan; yuanguan@cumt.edu.cn

Received 28 December 2015; Revised 23 February 2016; Accepted 3 April 2016

Academic Editor: Vlado Papic

Copyright © 2016 Penghui Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Compression technology is an efficient way to reserve useful and valuable data as well as remove redundant and inessential data from datasets. With the development of RFID and GPS devices, more and more moving objects can be traced and their trajectories can be recorded. However, the exponential increase in the amount of such trajectory data has caused a series of problems in the storage, processing, and analysis of data. Therefore, moving object trajectory compression undoubtedly becomes one of the hotspots in moving object data mining. To provide an overview, we survey and summarize the development and trend of moving object compression and analyze typical moving object compression algorithms presented in recent years. In this paper, we firstly summarize the strategies and implementation processes of classical moving object compression algorithms. Secondly, the related definitions about moving objects and their trajectories are discussed. Thirdly, the validation criteria are introduced for evaluating the performance and efficiency of compression algorithms. Finally, some application scenarios are also summarized to point out the potential application in the future. It is hoped that this research will serve as the steppingstone for those interested in advancing moving objects mining.

1. Introduction

In recent years, with the rapid development and extensive use of GPS devices, RFID sensors, satellites, and wireless communication technologies, it is possible to track various kinds of moving objects all over the world and collect a myriad of trajectory data with respect to the mobility of various moving objects (such as people, vehicles, and animals) containing a great deal of knowledge. These data need an urgent and effective analysis. A moving object spatial-temporal trajectory is a sequence of position, attribute, and time [1], which are three basic characteristics of geographic phenomena and three basis data of GIS database [2]. Moving objects move continuously while their locations can only be updated at discrete times, leaving the location of a moving object between two updates uncertain, for the limit of acquisition, storage, and processing technologies [3]. The simplest description of a trajectory is a finite sequence of geolocations with timestamps.

As time goes on, it will lead to a series of difficulties in storing, transmitting, and analyzing data, for the size of trajectory data is sharply increasing and the scale of data is growing huge and complex. First of all, the sheer volumes of data can quickly overwhelm available data storage, which will make it difficult to store the data. For instance, if data is collected at 2 second intervals, 1 GB of storage capacity is required to store just over 800 objects for a single day. Therefore, the storage of data will result in an enormous cost. The cost of transmitting a large amount of trajectory data, which may be expensive and problematic, is the second major problem highlighting the need of compressing data. According to [4], the cost of sending a volume of data over remote networks can be prohibitively expensive, typically ranging from \$5 to \$7 per Mb. Thus, tracking a fleet of 800 vehicles for a single day would incur a cost of \$5,000 to \$7,000, or approximately \$1,825,000 to \$2,555,000 annually. Finally, along with the increasing of the data scale, it is difficult for us to extract the valuable and useful patterns.

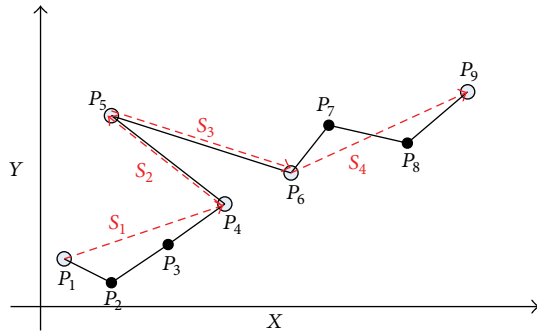


FIGURE 1: A schematic of the trajectory compression.

To address these issues, two categories of trajectory compression strategies have been proposed, aiming to reduce the size of a trajectory while not compromising much precision in its new data representation [5]. One is the offline compression, which reduces the size of trajectory after the trajectory has been fully generated. The other is online compression, compressing a trajectory instantly as an object travels. On the one hand, it can reduce the memory space by compressing, which will make the storage of data easier. On the other hand, it can cut down the size of data, which will be convenient for the transmission of data. What is more, it can reserve the useful information in trajectories and remove redundant data from trajectories, which have the potential to make the thorough analysis of trajectory data easier. The data compression is a method that reduces the size of data to cut down the memory space and improve the efficiency of transmission, storage, and processing without losing information or reorganizes data to reduce the redundancy and memory space according to certain algorithms. Data compression can be classified into two categories, namely, lossless and lossy compression. Moving object trajectory compression aims to reduce the size and memory space of a trajectory on the premise that the information contained in trajectory data is reserved; that is to say, in order to cut down the size of data, it removes redundant location points while ensuring the accuracy of the trajectory [6]. Figure 1 is a schematic diagram of a trajectory compression, where the original trajectory is represented by black lines and the compressed trajectory consists of red lines (namely, S_1 , S_2 , S_3 , and S_4). There are 9 points in the original trajectory, but only 5 points are retained to approximately represent the original trajectory after compressing whose compression ratio is close to 50%. Thus, it can be seen that trajectory compressions play an important role in the storage and analysis of data. But trajectory compression tends to cause a certain loss of information, while compressing trajectories. Therefore, various trajectory algorithms existing in literature balance the tradeoff between accuracy and storage size.

The trajectory compression technology derives from the topographic cartography and computer graphics. The most native and simplest compression method is uniform sampling algorithm which simply takes every i th point in the trajectory [7]. In 1961, Bellman put forward a new algorithm called

Bellman algorithm, which solves linear generalization problems by dynamic programming methods [8]. This method will guarantee that the segments connecting the specific number of points selected from the curve are closest to the original curve, but the time overhead of it is giant which is up to $O(n^3)$. One of the most classical trajectory compression algorithms called Douglas-Peucker algorithm was presented in 1973 by Douglas and Peucker [9]. In 2001, Keogh et al. put forward the Opening Window algorithm to compress trajectory data online. However, traditional error metrics (such as perpendicular distance) are not suitable for moving object trajectories, whose spatial characteristics and temporal characteristics need to be simultaneously considered, due to their internal features. In 2004, Meratnia and Rolf put forward a top-down speed-based algorithm and a top-down time-ratio algorithm [3]. The former improves the existing compression techniques by exploiting the spatiotemporal information hiding in the time series, while the latter is a transformation of DP algorithm which took a full consideration of spatiotemporal characteristics by replacing spatial error with SED. In 2006, Potamias et al. put forward STTrace algorithm estimating the safe zone of successor point by location, velocity, and direction. Meanwhile, it is suitable for small memory devices [10]. Gudmundsson et al. developed an implementation of the Douglas-Peucker path-simplification algorithm in 2009, which works efficiently even in the case where the polygonal path given as input is allowed to self-intersect [11]. In 2009, Schmid et al. proposed that trajectories stored in the form of trajectory points can be instead of semantic information of road networks [12]. Since then, many researchers have been doing a great deal of studies about the semantic information of road networks [13–18]. With the increasing of the data, traditional compression algorithms are quite limited for online trajectory data. Therefore, online trajectory compression becomes one of the hot topics [19–21]. For example, Opening Window Time-Ratio algorithm was put forward by Meratnia, which is an extension to Opening Window using SED instead of spatial error, to take temporal features into account [2]. And Trajcewski et al. put forward another online algorithm called Dead Reckoning algorithm, which estimates the successor point through the current point and its velocity [22]. Out of traditional position preserved trajectory compression algorithms, many scholars have focused on different perspectives. For example, Birnbaum et al. proposed a trajectory simplicity algorithm based on subtrajectories and their similarity [23]. Long et al. proposed a polynomial-time algorithm for optimal direction-preserving simplification, which supports a border application range than position-preserving simplification [24]. Nibali and He proposed an effective compression system for trajectory data called Trajic, which can fill the gap of good compression ratio and small error margin [25]. Similar to STTrace, Muckell et al. put forward the Spatial QUality Simplification Heuristic method [26]. In 2012, Chen et al. proposed a Multiresolution Polygonal Approximation algorithm, which compressed trajectories by a joint optimization on both the LSSD and the ISSD criteria [27]. In 2014, Muckell et al. proposed a new algorithm, SQUISH-E, which compresses

TABLE 1: Summary of trajectory compression algorithms (n is the size of trajectories).

| Algorithm | Computational complexity | Batch or online | Error criteria |
|---|--------------------------|-----------------|--|
| Uniform sampling algorithm | $O(n)$ | Batch/online | Target compression ratio (λ) |
| Bellman algorithm | $O(n^3)$ | Batch | Target compression ratio (λ) |
| Douglas-Peucker algorithm | $O(n \log n)$ | Batch | Spatial distance (ϵ) |
| Opening Window algorithm | $O(n^2)$ | Online | Spatial distance (ϵ) |
| Top-down time-ratio algorithm | $O(n \log n)$ | Batch | SED distance (ϵ) |
| Top-down speed-based algorithm | $O(n^2)$ | Batch/online | Speed difference (s_d) |
| Opening Window Time-Ratio algorithm | $O(n \log n)$ | Online | SED distance (ϵ) |
| Dead Reckoning | $O(n)$ | Online | SED distance (ϵ), velocity (v) |
| Semantic trajectory compression | $O(n^2)$ | Batch | Spatial distance (ϵ) |
| Paralleled road-network-based trajectory compression | $O(n)$ | Batch | TSND, NSTD |
| Similarity based compression of GPS trajectory data | $O(n \log n)$ | Batch | Similarity (μ) |
| Spatial QUality Simplification Heuristic algorithm | $O(\log n)$ | Batch/online | SED distance (ϵ) |
| Spatial QUality Simplification Heuristic-Extended algorithm | $O(\log n)$ | Batch/online | SED distance (ϵ), compression ratio (λ) |

trajectories with provable guarantees on errors [28]. This algorithm has the flexibility of tuning compression with respect to compression ratio and error. Algorithms involved in this paper are summarized in Table 1 from the complexity, application scope, and error metric of them.

In literature [29], the traditional trajectory compression algorithms were classified into the following 4 categories, which are now unable to contain all of the compression algorithms.

(1) *Top-Down*. The data series is recursively partitioned until some halting condition is met. The popular top-down compression methods include Douglas-Peucker algorithm, top-down speed-based algorithm, and top-down time-ratio algorithm.

(2) *Bottom-Up*. Starting from the finest possible representation, successive data points are merged until some halting condition is met. The algorithm may not visit all data points in sequence.

(3) *Sliding Window*. Starting from one to the end of the data series, a window of fixed size is moved over the data points and compression takes place only on the data points inside the window. Spatial QUality Simplification Heuristic method and SQUISH-E algorithm are the popular sliding window methods.

(4) *Opening Window*. Starting from one to the end of the data series, a compression takes place on the data points inside the window whose size is decided by the number of points to be processed. Its process will not end until some halting

condition is met. The window size is not constant while compressing. The famous Opening Window methods are Opening Window algorithm and Opening Window Time-Ratio algorithm.

The organization of this paper is as follows: the basic ideal of compression and typical algorithms of compression are introduced and discussed in Section 1. The related definitions about moving objects and their trajectories are summarized in Section 2. The survey of moving object compression algorithms is given in Section 3. Some validation criteria of compression performance are discussed in Section 4 to reveal their benefit for moving object compression. In Section 5, some public trajectory datasets are described. Some typical application scenarios are listed to show the application of moving object compression in Section 6. In Section 7, some disadvantages and future works are summarized.

2. Related Definitions about Moving Objects and Their Trajectories

2.1. Trajectory Data. A spatial-temporal trajectory of a moving object is defined as a sequence of position, attribute, and time in literature [1]. It is necessary for a formal description of a trajectory and its correlation attributes to describe the methods in this paper. A trajectory formally defined in literature [30] is also suitable in this paper. Giving TD as Trajectory Database denotes trajectory sets, and $TD = \{TR_1, TR_2, \dots, TR_n\}$. A trajectory (TR) is a chronological sequence consisting of multidimensional locations, which is denoted by $TR_i = \{P_1, P_2, \dots, P_m\}$ ($1 \leq i \leq n$). P_j ($1 \leq j \leq m$), a sampling point in TR_i , is represented as $\langle \text{Location}_j, T_j \rangle$,

which means that the position of the moving object is Location_j at time T_j . Location_j is a multidimensional location point. A trajectory $P_{c_1}, P_{c_2}, \dots, P_{c_m}$ ($1 \leq c_1 < c_2 < \dots < m$) represents a trajectory segment or subtrajectory of a trajectory TR_i , denoted as TS (Trajectory Segment), $\text{TS}_i = \{L_{i_1}, L_{i_2}, \dots, L_{i_{\text{num}}}\}$.

In this section, we classify the derivation of trajectories into 4 major categories, briefly introducing a few application scenarios in each category [31].

(1) *Mobility of People*. Real-world movements of people are recorded in the form of spatial-temporal trajectories passively and actively. Such records can be translated into a great amount of spatial-temporal trajectories that can be used in human behavior analysis and inferring social ties.

Active Recording. Travelers actively log their travel routes for the purpose of memorizing a journey and sharing experience with friends. In Flickr, a series of geotagged photos can formulate a spatial-temporal trajectory as each photo has a location tag and a time stamp corresponding to where and when the photo was taken. Likewise, the “check-ins” of a user in a location-based social network can be also regarded as a spatial-temporal trajectory, when sorted chronologically.

Passive Recording. A user carrying a mobile phone unintentionally generates many spatial-temporal trajectories represented by a sequence of cell tower IDs with corresponding transition times. In addition, transaction records of a credit card also indicate the spatial-temporal trajectory of the cardholder, as each transaction contains a time stamp and a merchant ID denoting the location where the transaction occurred.

(2) *Mobility of Transportation Vehicles*. A great number of vehicles (such as taxis, buses, vessels, and aircrafts) have been equipped with a GPS device. For instance, many taxis have been equipped with a GPS sensor, which enables them to report a time-stamped location with a certain frequency. Such reports formulate a large amount of spatial-temporal trajectories that can be used for resource allocation, traffic analysis, and improving transportation networks.

(3) *Mobility of Animals*. Biologists are collecting the moving trajectories of animals like tigers and birds, for the purpose of studying animals' migratory traces, behavior, and living situations.

(4) *Mobility of Natural Phenomena*. Meteorologists, environmentalists, climatologists, and oceanographers are busy collecting the trajectories of natural phenomena, such as hurricanes, tornados, and ocean currents. These trajectories capture the change of the environment and climate, helping scientists deal with natural disasters and protect the natural environment we live in.

2.2. Road Network. A road network is defined as a directed graph $G = (V, E)$, where V is a finite vertex set in which every vertex denotes a location point, and E is a finite edge

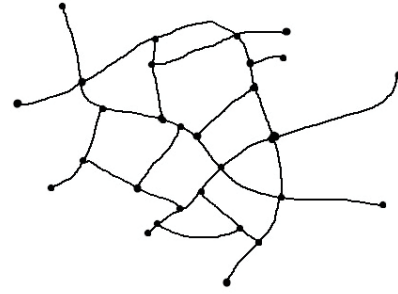


FIGURE 2: A schematic of road network.

set in which every edge denotes a segment connecting 2 vertexes. A road network can also be regarded as a constrained 2-dimensional space, often referred to as 1.5-dimensional space. The road network contains 29 vertexes (V) and 36 edges (E) in Figure 2.

In 2-dimensional space, a point (p) is a two-tuple in the form of (x, y) and a polyline (pl) is a set of points. The distance (d) of the points (p) in the polyline (pl) is the distance along the polyline (pl) from its starting point to point p . The definitions of road network point, distance, polyline, segment, and measurement in road network space are described as follow.

(1) *Road Network Point*. The point p in road network space can be denoted in the form of $p = (e, d)$, where e is the point in a road and d is the measurement of p along the road.

(2) *Road Network Distance*. The distance (d) between 2 random points (p_1 and p_2) in road network space is the shortest path length along the road from p_1 to p_2 .

(3) *Road Network Polyline*. The road network polyline is denoted as $\text{pl} = (p_1, p_2, \dots, p_n)$, where $n > 1$. The length of polyline is the summation of the distance between 2 adjacent vertexes.

(4) *Road Network Segment*. The road network segment in road network space is a road network polyline which owns and only owns 2 vertexes.

(5) *Edge of Road Network*. The edge (e) in road network is the path between 2 adjacent intersections.

(6) *A Trajectory Model Based on Road Network*. The trajectory model based on road network is a new representation of moving object trajectories, which matches GPS points with road network to more accurately describe the spatial motion information of moving objects by their motion laws in road network. Meanwhile, the model introduces a nonlinear interpolation function among sampling points to preferably describe variable motions. The trajectory model based on road network separates the locations from time stamps. In other words, a trajectory is represented by a spatial path and a temporal sequence. The spatial path of a trajectory

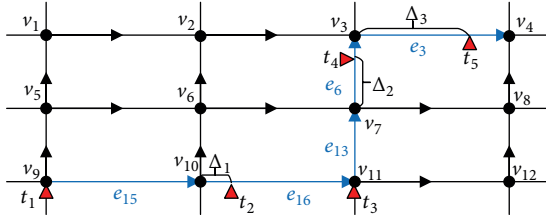


FIGURE 3: A trajectory in road network.

in a road network is a sequence of consecutive edges. As shown in Figure 3, a trajectory sequentially passes edges e_{15} , e_{16} , e_{13} , e_6 , and e_3 . Consequently, it can be represented by a spatial path in the format of $\langle e_{15}, e_{16}, e_{13}, e_6, e_3 \rangle$. Note that a trajectory can start from or end at any point of an edge, not necessarily an endpoint. The temporal information of a trajectory is captured by a two-tuple (d_i, t_i) , where d_i represents the road network distance the object has traveled at the time stamp t_i from the start of the trajectory and t_i represents the time the object has traveled at the location d_i from the start of the trajectory.

2.3. Perpendicular Distance and Synchronized Euclidean Distance. The perpendicular distance of point p is the shortest

$$d_{\perp} = \frac{|x_i * y_{i+2} + x_{i+2} * y_{i+1} + x_{i+1} * y_i - x_{i+2} * y_i - x_{i+1} * y_{i+2} - x_i * y_{i+1}|}{4 * \sqrt{(x_i - x_{i+2})^2 + (y_i - y_{i+2})^2}}, \quad (2)$$

$$\text{SED} = \sqrt{(x_i - x'_{i+2})^2 + (y_i - y'_{i+2})^2}.$$

2.4. Trajectory Similarity. The trajectory similarity is calculated by measuring the similarity between two trajectories or subtrajectories utilizing Euclidean distance, PCA Plus Euclidean distance, Hausdorff distance, Fréchet distance, and so on. In this section, we introduce 4 classical trajectory similarity measurements.

2.4.1. Euclidean Distance. Let L_i and L_j be p -dimensional trajectory segments with length of n . Their Euclidean distance denoted as D_E is given in

$$D_E(L_i, L_j) = \frac{1}{n} \sum_{k=1}^n \sqrt{\sum_{m=1}^p (a_k^m - b_k^m)^2}. \quad (3)$$

2.4.2. PCA Plus Euclidean Distance. When computing PCA (Principal Components Analysis) Plus Euclidean distance, trajectory is firstly represented as a 1D signal by concatenating the x and the y projections. Then, location signal is converted into the first few PCA coefficients. The trajectory similarity is

distance between the current point and the segment connecting the first and last points of the trajectory, while Synchronized Euclidean Distance of point p is the distance between the currently real point and the synchronized point acquired by interpolating between the precursor point and the successor point of the current point. In Figure 4, the perpendicular distance of point p_{i+1} is denoted as d_{\perp} and the Synchronized Euclidean Distance of point p_{i+1} is denoted as SED.

As shown in Figure 4, p_{i+1} is the current sampling point, $\overline{p_i p_{i+2}}$ is the segment connecting the first and last points of the trajectory, p'_{i+1} is the synchronized point of p_{i+1} in segment $\overline{p_i p_{i+2}}$, and the coordinate of p'_{i+1} is calculated by

$$\begin{aligned} t'_{i+1} &= t_{i+1}, \\ x'_{i+1} &= x_{i+1} + \frac{t_{i+1} - t_i}{t_{i+2} - t_i} (x_{i+2} - x_i), \\ y'_{i+1} &= y_{i+1} + \frac{t_{i+1} - t_i}{t_{i+2} - t_i} (y_{i+2} - y_i). \end{aligned} \quad (1)$$

The perpendicular distance and Synchronized Euclidean Distance between p_{i+1} and $\overline{p_i p_{i+2}}$ are calculated by formula (2), according to formula (1):

the Euclidean distance computed with the PCA coefficients, as shown in

$$D_{\text{PCA}_E}(L_i, L_j) = \sqrt{\sum_{k=1}^K (a_k^c - b_k^c)^2}. \quad (4)$$

Here, a_k^c and b_k^c are, respectively, the k th PCA coefficient in two-dimensional space trajectory segments L_i and L_j , whose length is n , and $K \ll 2n$.

2.4.3. Hausdorff Distance. Given 2 trajectory segments L_i and L_j , their Hausdorff distance denoted as $D_H(L_i, L_j)$ is given in

$$\begin{aligned} D_H(L_i, L_j) &= \max(h(L_i, L_j), h(L_j, L_i)), \\ \text{where } h(L_i, L_j) &= \max_{a \in L_i} \left(\min_{b \in L_j} (\text{dist}(a, b)) \right). \end{aligned} \quad (5)$$

In the formula, $h(L_i, L_j)$ is the direct Hausdorff distance of L_i and L_j , and $\text{dist}(a, b)$ is the Euclidean distance between sampling points a and b in L_i and L_j , respectively.

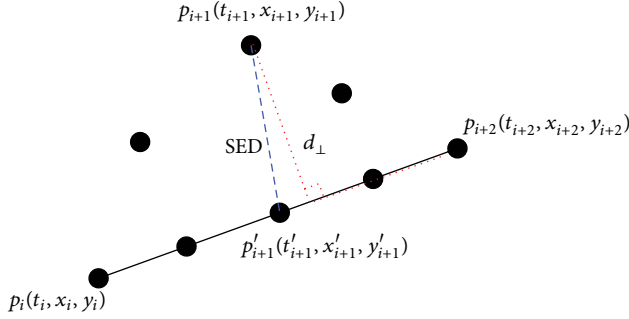


FIGURE 4: A schematic of perpendicular distance and Synchronized Euclidean Distance.

2.4.4. Discrete Fréchet Distance. Discrete Fréchet distance fully considers the location and sequential relationship of the point in trajectories while measuring their similarity. It scans the points on two trajectories and calculates its Euclidean distance point by point. The maximum Euclidean distance is the Discrete Fréchet distance between two trajectories. The calculating formula is shown as

$$D_F(L_i, L_j) = \min \{ \|C\|, C \text{ is the coupling between } L_i \text{ and } L_j \}, \quad (6)$$

$$\text{where } \|C\| = \max_{k=1}^K \text{dist}(a_i^k, b_j^k).$$

Here, L_i and L_j are the trajectory segments whose lengths are m and n , respectively. Consider $K = \min(m, n)$. a_i^k and b_j^k are the k th points on trajectory segments L_i and L_j , respectively. $\text{dist}(a_i^k, b_j^k)$ is the Euclidean distance between a_i^k and b_j^k .

2.4.5. Others. In addition to the 4 trajectory similarity measures discussed above, Vlachos et al. put forward longest common subsequence which is different from distance calculation and is used to obtain the longest common subsequence existing in two trajectory sequences [32]. Chen et al. proposed Dynamic Time Warping method which is a well-known technique to find an optimal alignment between two given (time-dependent) sequences under certain restrictions [33]. Lee et al. put forward a comprehensive distance function which is composed of three components: the angle distance, the parallel distance, and the perpendicular distance [1]. The method overcomes the limitations of the trajectory similarity measure by the length of trajectory segments. It can more comprehensively measure the similarity between trajectory segments. Yuan et al. extract trajectory structure and propose a structure similarity measurement for comparing trajectories in microlevel [34].

2.5. Compressive Sensing. Compressive sensing (CS) is an efficient signal processing technique to acquire and reconstruct a signal by finding solutions to underdetermined linear systems. It is also known as compressed sensing,

or sparse sampling. CS is with the principle that, through optimization, the sparsity of a signal can be exploited to recovery from far fewer samples than required by the Shannon-Nyquist sampling theorem. There are two conditions under which recovery is possible. The first one is sparsity which requires the signal to be sparse in some domain. The second one is incoherence which is applied through the isometric property which is sufficient for sparse signals [35].

In this section, we will discuss CS given in literature [16] briefly. Given a vector $x \in \mathbb{R}^n$, the representation $\theta \in \mathbb{R}^n$ can be computed on a basis $\psi \in \mathbb{R}^{n \times n}$ by solving the linear equation $x = \psi\theta$ which is said to be compressible if θ has a large number of elements with small magnitude. If there is a basis on which a given vector x has a compressible representation, then x is also compressible. Compressive sensing considers the problem of recovering an unknown compressible vector x from its projections. Let Φ be an $m \times n$ projection matrix with $m < n$. Consider the equation

$$y = \Phi x + z, \quad (7)$$

where $z \in \mathbb{R}^m$ is a noise vector whose norm is bounded by ϵ . Compressive sensing aims to reconstruct x from y and Φ given the knowledge that x is compressible on the basis ψ . Compressive sensing shows that under certain conditions it is possible to recover x by solving the following ℓ_1 optimization problem:

$$\begin{aligned} \min_{\hat{\theta} \in \mathbb{R}^n} \quad & \|\hat{\theta}\|_1 \\ \text{subject to} \quad & \|y - \Phi\psi\hat{\theta}\|_2 \leq \epsilon. \end{aligned} \quad (8)$$

Given $\hat{\theta}$, x can be estimated from $\hat{x} = \psi\hat{\theta}$.

In the context of trajectory compression, x is the trajectory measured by a Mobile Sensor Networks node. The dimension of x is large. The MSN node computes $y = \Phi x$ and transmits y to the server. The server can compute an estimated trajectory \hat{x} by using y , Φ , and ψ to solve the aforementioned ℓ_1 optimization problem as shown in (8). Note that the compression is lossy with $1 - m/n$ representing both space savings and reduction in wireless transmission requirement.

3. Trajectory Compression Algorithms

In this section, we comprehensively analyze moving object trajectory compression algorithms which have been one of the research hotspots in the moving object data mining field. Existing trajectory compression algorithms include 2 categories: single trajectory compression (STC) and multiple trajectory compression (MTC). The former compresses each trajectory individually ignoring the commonalities among trajectories, and the latter compresses several trajectories or subtrajectories at the same time by the commonalities among trajectories (such as similarity). There are some different classification strategies about compression algorithms, but they are not unable to contain all of the compression algorithms as the rapid development of compression technology. Therefore,

in this paper, we present a new classification strategy to divide trajectory compression algorithms into 5 categories on the basis of compression theories.

3.1. Distance Based Trajectory Compression. Distance (such as perpendicular distance and Synchronized Euclidean Distance) information is one of the most classic and common compression metrics in trajectory compression algorithms. Many researchers have devoted their talent to compress trajectories by deciding whether the sampling point is reserved based on distance, since 1973. The earliest distance based trajectory compression algorithm is Douglas-Peucker algorithm proposed by Douglas and Peucker [9], which recursively selects the point whose perpendicular distance is greater than given threshold until all points reserved meet the condition. Keogh et al. put forward Opening Window algorithm that online compresses trajectory data based on perpendicular distance. A transformation of Douglas-Peucker algorithm called top-down time-ratio algorithm, which takes a full consideration of spatial-temporal characteristics by replacing perpendicular distance with SED, is proposed by Meratnia and Rolf [3]. Then, an extension to Opening Window called Opening Window Time-Ratio algorithm using SED instead of perpendicular distance to take temporal features into account is proposed by Wu and Cao [2]. Gudmundsson et al. developed an implementation of the Douglas-Peucker algorithm which works efficiently even in the case where the polygonal path given as input is allowed to self-intersect [11].

Perpendicular distance based trajectory compression is simple and efficient, but it just considers the spatial features and ignores the temporal features of trajectories. Synchronized Euclidean Distance based trajectory compression not only is simple and efficient but also has a better compression effect than perpendicular distance based trajectory compression for it takes the spatial-temporal features of trajectories into account. Distance based trajectory compression provides an effective way to compress trajectory data and a satisfactory compression result, which has been applied to many fields, such as animal migration, hurricane prediction, and aerospace field. But there are some obvious shortcomings in processing limited trajectories, such as the trajectory of human activities in urban and taxi motion track and keeping the internal features in trajectories for distance based trajectory compression pays more attention to keeping the holistic geometrical characteristics of trajectories.

3.2. Velocity Based Trajectory Compression. Velocity is one of the most basic features of moving objects and it can reflect the motion features of moving objects as well as the internal features in trajectories. The researches on compressing trajectory data based on velocity are not perfect by now. A famous velocity based trajectory compression is top-down speed-based algorithm proposed by Meratnia and Rolf [3] improving the existing compression techniques by exploiting the spatiotemporal information hiding in the time series which can be made by analyzing the derived speeds subsequent to the trajectory. A large difference between the travel speeds of two subsequent segments is a criterion that

can be applied to retain the data point in the middle. An online algorithm called Dead Reckoning algorithm proposed by Trajcevski et al. [22] compresses trajectory by estimating the successor point through the current point and its velocity. A polynomial-time algorithm for optimal direction-preserving trajectory simplification, which supports broader application range than position-preserving simplification, proposed by Long et al. [24] can be also regarded as a velocity based trajectory compression. This method uses the maximum angular difference between the direction of the movement during each time period in original trajectory and the direction of the movement during the same time period in a simplification of original trajectory.

Velocity based trajectory compression not only is simple and efficient but also can keep the internal features in trajectories; however, it is not popular, for the existing velocity based trajectory compression methods only take speed into account which may lead to greater errors and break the holistic geometrical characteristics of trajectories. In the future study, we hope that researchers will pay their attention to compressing trajectory data by various features of velocity (such as velocity direction and accelerated velocity) except for the magnitude of velocity.

3.3. Semantic Trajectory Compression. Semantic information in road network has more practical significance in representing moving object trajectories that are collected from limited moving objects. Semantic trajectory compression stores trajectories in the form of semantic information in road network instead of trajectory points, compresses spatial information in trajectory data by spatial compression methods, and compresses temporal information in trajectory data by temporal compression techniques, until some halting condition is met. The new and novel representation for trajectories that replaces trajectory data by the form of semantic information in road network was proposed by Schmid et al. [12] in 2009. Many researchers have paid their attention to semantic trajectory compression since then. Semantic trajectory compression was applied to human motion dataset in urban area by Richter et al. [13] which identifies the relevant reference points along the trajectory, determines all possible descriptions of how movement continues from here, and exploits motion feature description of reference points to compress trajectory data. Song et al. [14] proposed a new framework, namely, paralleled road-network-based trajectory compression, to effectively compress trajectory data under road network constraints. Different from existing works, PRESS proposed a novel representation for trajectories to separate the spatial representation of a trajectory from the temporal representation and proposes a Hybrid Spatial Compression (HSC) algorithm and error Bounded Temporal Compression (BTC) algorithm to compress the spatial and temporal information of trajectories, respectively.

Semantic trajectory compression is only suitable for limited moving objects, such as movement in road network, urban movement, and orbital trajectory, which will get a more realistic significance result in compressing trajectories of limited moving objects.

3.4. Similarity Based Trajectory Compression. Similarity based trajectory compression splits original trajectories into subtrajectories and then clusters subtrajectories with high similarity into the same group and clusters subtrajectories with low similarity into the different groups. And then it unifies spatial information of trajectory data in the same group by a certain strategy which will keep a set of spatial information and all temporal information in every group, until some halting condition is met. A famous similarity based compression is similarity based compression of GPS trajectory data proposed by Birnbaum et al. [23] which splits trajectories into subtrajectories according to the similarities among them. For each collection of similar subtrajectories, this technique stores only one subtrajectory's spatial data. Each subtrajectory is then expressed as a mapping between itself and a previous subtrajectory.

Similarity based trajectory compression has great advantages in retaining the commonalities among trajectories. It is suitable for trajectory set and may be not suitable for a single trajectory for the error may be large.

3.5. Priority Queue Based Trajectory Compression. Priority queue based trajectory compression selects the best subset of trajectory points and permanently removes redundant and inessential trajectory points from original trajectory by utilizing local optimization strategies, until some halting condition is met. The Spatial QUality Simplification Heuristic (SQUISH) method based on the priority queue data structure proposed by Muckell et al. [26] prioritizes the most important points in a trajectory stream. It uses local optimization to select the best subset of points and permanently removes redundant or insignificant points from the original GPS trajectory. Three years later, Muckell et al. [28] presented a new version of SQUISH, called SQUISH-E (Spatial QUality Simplification Heuristic-Extended) which has the flexibility of tuning compression with respect to compression ratio and error.

Priority queue based trajectory compression is not only an online trajectory compression algorithm but also a trajectory compression algorithm that requires presetting the memory buffer. Hence, it can be well applied to real-time applications and small memory devices. It is suitable for all kinds of trajectories, but the compression effect and matching effect may be a little worse than the other compression methods.

3.6. Others. Considering that if the movement pattern and internal features are neglected, applications, such as trajectory clustering, outlier detection, and activity discovery may be not so accurate as we expected. Therefore, we expect that a new algorithm called structure features based trajectory compression which compresses trajectories based on movement pattern and structure features in trajectories, such as moving direction of objects, internal fluctuation in trajectories, and trajectory velocity or acceleration, will attract more attention of researchers, for instance, a polynomial-time algorithm for optimal direction-preserving simplification proposed by Long et al., which supports border application range than

position-preserving simplification [24]. At present, most of the portable equipment used for data collection is inexpensive, power saving, and of lower computational capability, while the data processing procedure is often performed in supercomputers which have a higher computational capability. In order to effectively reduce the transport cost, we expect that compressive sensing based trajectory compression, which reduces the data scale in the process of acquiring data by combining compressive sensing with trajectory features, will attract more attention of researchers, for instance, Rana et al. present an adaptive algorithm for compressive approximation of trajectory in 2011, which performs trajectory compression, so as to maximize the information about the trajectory subject to limited bandwidth [36]. Four years later, another compression method called adaptive trajectory (lossy) compression algorithm based on compressive sensing has been proposed by Rana et al., which has two innovative elements [16]. First, they propose a method to compute a deterministic projection matrix from a learnt dictionary. Second, they propose a method for the mobile nodes to adaptively predict the number of projections needed based on the speed of the mobile nodes.

4. Validation Criteria of Compression Performance

Compression result validation is very important for compression algorithms and it can measure the level of success and correctness reached by the algorithms. There are many solutions to validate the result, mainly including Analysis, Experience, Evaluation, and Example. The Analysis solution includes rigorous derivation and proof or carefully designed experiment with statistically significant results. Experience solution is applied in real-world scenarios or projects and the evidence of approach's correctness (usefulness or effectiveness) can be obtained from the process of execution. Evaluation uses a set of examples to illustrate the proposed approach, with a nonsystemic analysis of gathered information from the execution of examples. Example uses only one or several small-scale examples to illustrate the proposed approach, without any evaluation or comparison of the execution result. In this section, we mainly discuss 2 kinds of compression validation solutions. The first compression validation solution is performance metrics which are used for comparing the efficiency and performance of trajectory compression algorithms. And the other compression validation solution is accuracy metrics which are used for comparing the accuracy and information loss of trajectory compression algorithms. This section, respectively, denotes the original trajectory as OT whose length is m and the compressed trajectory as RT whose length is n , in order to facilitate the validation of trajectory compression.

4.1. Performance Metrics

4.1.1. Compression Ratio. Compression ratio (\mathbb{R}) is an important index to measure the advantages and disadvantages of

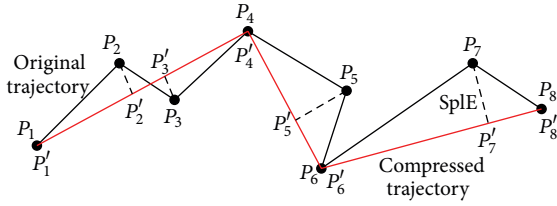


FIGURE 5: A schematic of spatial error.

trajectory compression performance, which is defined as in

$$\mathbb{R} = \left(1 - \frac{n}{m}\right) * 100\%. \quad (9)$$

Compression ratio is the most common compression index which can accurately reflect the change of the size of trajectory data. But \mathbb{R} is influenced by the original signal data sampling rate and quantization accuracy and so on; it is difficult to make an objective measurement. For instance, a compression ratio of 70% indicates that 30% of the original points remained in the compressed representation of the trajectory; namely, if there are 100 points in original trajectory, only 30 points will be reserved in the compressed representation of the trajectory after compressing.

4.1.2. Compression Time. Compression time (T) is an important index to measure the efficiency of trajectory compression performance, which reflects the total time required by the compression. For example, a compression time of 24 indicates that the total time of compressing original trajectory is 24 ms.

4.2. Accuracy Metrics

4.2.1. Spatial Error. Given an original trajectory OT and its compressed representation RT, the spatial error (SplE) of RT with respect to a point p_i in OT is defined as the distance between $p_i(x_i, y_i, t_i)$ and its estimation $p'_i(x'_i, y'_i, t'_i)$. If RT contains p_i , then p'_i is p_i (e.g., $p'_1 = p_1$, $p'_4 = p_4$, $p'_6 = p_6$, and $p'_8 = p_8$ in Figure 5 where there is a trajectory containing p_1, p_2, \dots, p_8). Otherwise, p'_i is defined as the closest point to p_i along the line between precursor point and successor point of p_i in trajectory RT. The precursor point of p_2 is p_1 and the successor point of p_2 is p_4 . Therefore, the spatial error of RT with respect to p_2 is the perpendicular distance from p_2 to line $\overline{p_1 p_4}$.

4.2.2. SED Error. Temporal characteristics of trajectory data are not considered in spatial error, so Synchronized Euclidean Distance (SED) is introduced to overcome this limitation. SED is also the distance between $p_i(x_i, y_i, t_i)$ and its estimation $p'_i(x'_i, y'_i, t'_i)$, which is obtained by linear interpolation method, owning the same time coordinate with p_i . If RT contains p_i , then p'_i is p_i (e.g., $p'_1 = p_1$, $p'_4 = p_4$, $p'_6 = p_6$, and $p'_8 = p_8$ in Figure 6 where there is a trajectory containing p_1, p_2, \dots, p_8). Otherwise, p'_i is defined as the location point owning the same time coordinate with p_i in trajectory RT.

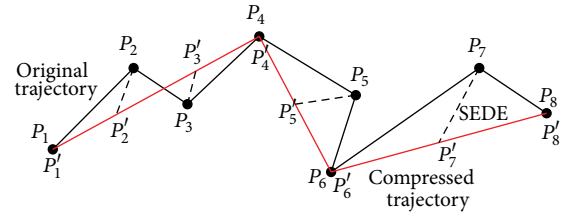


FIGURE 6: A schematic of SED error.

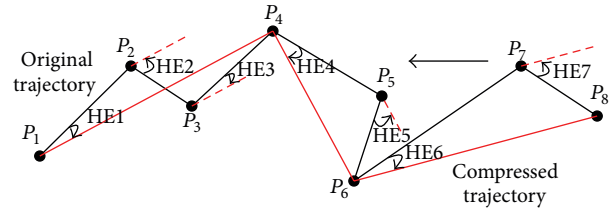


FIGURE 7: A schematic of heading error.

The estimation point of p_2 is p'_2 . Therefore, the SED error of RT with respect to p_2 is the distance between p_2 and p'_2 .

4.2.3. Heading Error. Heading error (HE) is the angular deflection between moving direction from the actual location point $p_{i-1}(x_{i-1}, y_{i-1}, t_{i-1})$ to $p_i(x_i, y_i, t_i)$ along original trajectory and moving direction from the estimation location point $p'_{i-1}(x'_{i-1}, y'_{i-1}, t'_{i-1})$ to $p'_i(x'_i, y'_i, t'_i)$ along compressed trajectory. The estimation p'_i owning the same time coordinate with p_i is obtained by linear interpolation method. As shown in Figure 7, we specify clockwise direction is positive value and anticlockwise direction is negative value, to facilitate the calculation.

4.2.4. Speed Error. Speed error (SpDE) is an important metric for various kinds of transit applications. For instance, velocity measurement system gets overspeed hotspots by velocity information [37], as well as acceleration and deceleration data help to identify all kinds of irregular driving behaviors, which will help police to find vehicle's illegal activities [38]. The computing method of speed error is similar to heading error. It calculates the difference value between actual velocity and estimated velocity instead of calculating angular deflection.

4.2.5. Information Loss Degree. Information Loss Degree (ILD) that can comprehensively analyze the accuracy and error of trajectory compression results is a comprehensive index to measure the advantages and disadvantages of trajectory compression effectiveness. Information Loss Degree can be calculated by the SED distance, Dynamic Time Warping distance, and Speed Corner between original trajectory and compressed trajectory.

Information Loss Degree based on SED (ILDSED) is the mean value of maximum SED distance error (SEDE_{max}(OT, RT)), average SED distance error (SEDE_{avg}(OT, RT)), and minimum SED distance error (SEDE_{min}(OT, RT)) between

original trajectory OT and compressed trajectory RT, which can be calculated as

$$\text{ILDSED}(\text{OT}, \text{RT}) = \frac{\text{SEDE}_{\text{max}}(\text{OT}, \text{RT}) + \text{SEDE}_{\text{avg}}(\text{OT}, \text{RT}) + \text{SEDE}_{\text{min}}(\text{OT}, \text{RT})}{3}. \quad (10)$$

Information Loss Degree based on DTW (ILDdtw) is measured by the time warping distance between original

trajectory OT and compressed trajectory RT, which can be calculated as

$$\text{ILDdtw}(\text{OT}, \text{RT}) = \begin{cases} 0 & m = n = 0 \\ \infty & m = 0 \parallel n = 0 \\ \text{SEDE}(\text{op}_1, \text{rp}_1) + \min \begin{cases} \text{ILDdtw}(\text{Rest}(\text{OT}), \text{Rest}(\text{RT})) \\ \text{ILDdtw}(\text{Rest}(\text{OT}), \text{RT}) \\ \text{ILDdtw}(\text{OT}, \text{Rest}(\text{RT})) \end{cases} & \text{others.} \end{cases} \quad (11)$$

Here, $\text{SEDE}(\text{op}_1, \text{rp}_1)$ is the SED error between point op_1 and rp_1 , which, respectively, are the first point of OT and RT. $\text{Rest}(\text{OT})$ and $\text{Rest}(\text{RT})$ are the remaining trajectory after removing the first sampling point. ILDdtw calculates the Information Loss Degree by DTW error.

Information Loss Degree based on Speed Corner (ILDcorner) is measured by the original and compressed Speed Corner of moving objects which can be calculated as

$$\begin{aligned} \text{ILDcorner}(\text{OT}, \text{RT}) \\ = \frac{\sum_1^{\min(m,n)} ((|\theta_i - \theta_j|) / (|\theta_i| + |\theta_j|))}{m + n}. \end{aligned} \quad (12)$$

5. Public Trajectory Datasets

There are quite a few real trajectory datasets that are publicly available. In this section, a detailed description of real trajectory datasets is given from their sources, characteristics, sampling rate, and so on.

5.1. GeoLife Trajectory Dataset. A GPS trajectory dataset from Microsoft Research GeoLife project was collected by 182 users in a period of over 5 years from April 2007 to August 2012. A GPS trajectory of this dataset is represented by a sequence of time-stamped points, each of which contains the information of latitude, longitude, and altitude. This dataset whose size is 1.55 GB contains 17,621 trajectories with a total distance of 1,292,951 kilometers and a total duration of 50,176 hours. These trajectories were recorded by different GPS loggers and GPS-phones and have a variety of sampling rates. 91.5 percent of the trajectories are logged in a dense representation, for example, every 1~5 seconds or every 5~10 meters per point.

5.2. T-Drive Taxi Trajectories. A sample of trajectories from Microsoft Research T-Drive project was generated by over 30,000 taxicabs in a period of 6 months from March 2009 to August 2009. The total distance traveled by the taxis is more than 800 million kilometers and the total number of GPS points is nearly 1.5 billion. The size of the dataset is 756 Mb and the average sampling interval and average distance between two consecutive points are around 3.1 minutes and 300 meters, respectively.

5.3. GPS Trajectory with Transportation Labels. This is a portion of GPS trajectory dataset collected in (Microsoft Research Asia) GeoLife project. Each trajectory has a set of transportation mode labels, such as driving, taking a bus, riding a bike, and walking. There is a label file associated with each folder storing the trajectories of a user. A GPS trajectory of this dataset is represented by a sequence of time-stamped points, each of which contains the information of latitude, longitude, height, speed, heading direction, and so forth. These trajectories were recorded by different GPS loggers or GPS-phones and have a variety of sampling rates. 95 percent of the trajectories are logged in a dense representation, for example, every 2~5 seconds or every 5~10 meters per point, while a few of them do not have such a high density being constrained by the devices. The size of the dataset is 560 Mb.

5.4. Check-in Data from Location-Based Social Networks. The dataset from a LBSN in China whose size is 10.68 Mb consists of 2,756,710 check-in data generated by 10,049 users excluding the timestamp and relationships between users. Each check-in includes the information of ID, latitude, longitude, and timestamp.

5.5. Hurricane Trajectories. This dataset is provided by the National Hurricane Service (NHS) containing 1,740 trajectories of Atlantic Hurricanes from 1851 to 2012. NHS also provides annotations of typical hurricane tracks for each month throughout the annual hurricane season that spans from June to November. The data were collected every 6 hours.

5.6. Movebank Animal Tracking Data. Movebank is a free, online database of animal tracking data helping animal tracking researchers to manage, share, protect, analyze, and archive their data. Movebank is an international project with over 11,000 users, including people from research and conservation groups around the world. A lot of datasets are collected in this database, such as Continental black-tailed godwits (data from Senner et al., 2015) whose size is 5.161 Mb [39], and Navigation experiments in lesser black-backed gulls (data from Wikelski et al., 2015) whose size is 29.09 Mb [40].

6. Application Scenarios of Trajectory Compression

(1) The unrestricted movement of moving objects is a typical application scenario of trajectory compression, such as a bird flying in the sky, a fish swimming in the sea, and a horse running on the grassland. Distance based trajectory compression and velocity based trajectory compression have a high efficiency in this application scenario and a good application prospect in many fields, such as studying animals' migratory traces, behavior, and living situations, as well as animal migration research and hurricanes, tornados, and ocean currents prediction. For instance, animal tracking data helps biologists understand how individuals and populations move within local areas, migrate across oceans and continents, and evolve through millennia. This information is being used to address environmental challenges such as climate and land use change, biodiversity loss, invasive species, and the spread of infectious diseases. However, the data that need to be analyzed always have a large scale which will make them difficult to be analyzed and find the useful information in the data, so it is necessary to compress the data by removing the redundant data and only keeping the valuable data.

(2) The restricted movement of moving objects is another very important application scenario of trajectory compression, such as the motion track of taxis in urban area. Semantic trajectory compression can effectively and efficiently compress the trajectory data in this application scenario with respect to transport analysis, smart city plan, and smart transportation management. For instance, vast amounts of trajectory data can be collected by vehicle positioning equipment and other devices, which can be used to help police to find dangerous driving, predict the stream of people in major festivals in important places of a city, and trace escaping route of criminals. But the large scale of the data will lead to the difficulty of finding dangerous driving, predict the stream of people in major festivals in important places of a city, and trace escaping route of criminals for police, so the data are in

urgent need of compression which can remove the redundant data and only reserve the valuable data in the dataset.

(3) Priority queue based trajectory compression is widely applied to small memory devices and has a high efficiency in this scenario. For instance, most of the portable mobile devices have a small memory. If the data have to be analyzed on portable mobile devices, it is easy to meet the breakdown (out of memory) that will lead to the device not working. Therefore, it is necessary for portable mobile devices with a compress application that may compress the data by removing the redundant data before analyzing them.

7. Conclusion and Future Work

Trajectory compression is an efficient way to reduce the size of trajectory data and reserve the useful and valuable information in large scale dataset, which is one of the important components of data mining technology. In this paper, the research status and new development of moving object trajectory compression algorithms in recent years have been surveyed and summarized. Firstly, the representative compression algorithms proposed in recent years are analyzed and summarized from algorithmic thinking, key technology, and the advantages and disadvantages. Then, the existing algorithms are classified into several categories according to compression theories. Thirdly, some typical valid criteria of compression result are summarized. Lastly, some application scenarios are pointed out and discussed.

On the basis of summarizing and surveying on the moving object trajectory compression and its theories, methods, and techniques, we also summarize the problems and the challenges existing in moving object trajectory compression, which mainly includes the following aspects: (1) Most of the current trajectory compression algorithms pay more attention to the holistic outline geometrical characters of trajectory and ignore the movement patterns and the internal features in trajectories. (2) Most of the current trajectory compression algorithms cannot fully combine time dimension with space dimensions, and they just regard time dimension as the additional dimension of space dimension of trajectory object. (3) The general applicability of trajectory compression algorithm is low. (4) Few researchers have paid their attention to compressing trajectories by compressive sensing which reduces the data scale in the process of acquiring data.

Competing Interests

No potential conflict of interests was reported by the authors.

Authors' Contributions

Penghui Sun and Shixiong Xia contributed equally to this work.

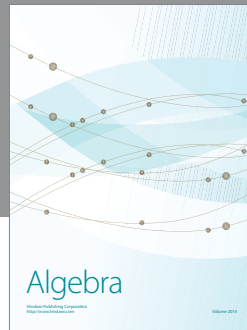
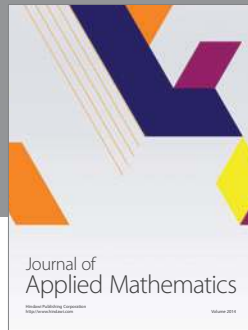
Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities, China (with Grant 2015XKMS085).

References

- [1] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '07)*, pp. 593–604, ACM, Beijing, China, June 2007.
- [2] X.-C. Wu and Z.-Y. Cao, "Basic conception, function and implementation of temporal GIS," *Earth Science-Journal of China University of Geosciences*, vol. 27, no. 3, pp. 241–245, 2002.
- [3] N. Meratnia and A. Rolf, "Spatiotemporal compression techniques for moving point objects," in *Advances in Database Technology—EDBT 2004*, vol. 2992 of *Lecture Notes in Computer Science*, pp. 765–782, Springer, Berlin, Germany, 2004.
- [4] M. Prior-Jones, *Satellite Communications Systems Buyers' Guide*, British Antarctic Survey, Cambridge, UK, 2008.
- [5] H. Wang, H. Su, K. Zheng, S. Sadiq, and X. Zhou, "An effectiveness study on trajectory similarity measures," in *Proceedings of the Twenty-Fourth Australasian Database Conference (ADC '13)*, vol. 137, pp. 13–22, 2013.
- [6] D. Zhang and X. Zhang, "A spatiotemporal compression algorithm for GPS trajectory data," *Journal of Transport Information and Safety*, vol. 31, no. 3, pp. 32–48, 2013.
- [7] W. R. Tobler, "Numerical map generalization," Discussion Paper, Michigan Inter-University Community of Mathematical Geographers, Ann Arbor, Mich, USA, 1966.
- [8] R. Bellman, "On the approximation of curves by line segments using dynamic programming," *Communications of the ACM*, vol. 4, no. 6, p. 284, 1961.
- [9] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica*, vol. 10, no. 2, pp. 112–122, 1973.
- [10] M. Potamias, K. Patroumpas, and T. Sellis, "Sampling trajectory streams with spatiotemporal criteria," in *Proceedings of the 18th International Conference on Scientific and Statistical Database Management (SSDBM '06)*, pp. 275–284, Vienna, Austria, July 2006.
- [11] J. Gudmundsson, J. Katajainen, D. Merrick, C. Ong, and T. Wolle, "Compressing spatio-temporal trajectories," *Computational Geometry*, vol. 42, no. 9, pp. 825–841, 2009.
- [12] F. Schmid, F. K. Richter, and P. Laube, "Semantic trajectory compression," in *Advances in Spatial and Temporal Databases*, pp. 411–416, Springer, Berlin, Germany, 2009.
- [13] K. F. Richter, F. Schmid, and P. Laube, "Semantic trajectory compression: representing urban movement in a nutshell," *Journal of Spatial Information Science*, no. 4, pp. 3–30, 2015.
- [14] R. Song, W. Sun, B. Zheng, and Y. Zheng, "PRESS: a novel framework of trajectory compression in road networks," *Proceedings of the VLDB Endowment*, vol. 7, no. 9, pp. 661–672, 2014.
- [15] G. Kellaris, N. Pelekis, and Y. Theodoridis, "Map-matched trajectory compression," *Journal of Systems and Software*, vol. 86, no. 6, pp. 1566–1579, 2013.
- [16] R. Rana, M. Yang, T. Wark, C. T. Chou, and W. Hu, "SimpleTrack: adaptive trajectory compression with deterministic projection matrix for mobile sensor networks," *IEEE Sensors Journal*, vol. 15, no. 1, pp. 365–373, 2015.
- [17] K. Liu, Y. Li, J. Dai et al., "Compressing large scale urban trajectory data," in *Proceedings of the 4th International Workshop on Cloud Data and Platforms (CloudDP '14)*, p. 3, ACM, Amsterdam, The Netherlands, 2014.
- [18] I. S. Popa, K. Zeitouni, V. Oria, and A. Kharrat, "Spatio-temporal compression of trajectories in road networks," *Geoinformatica*, vol. 19, no. 1, pp. 117–145, 2014.
- [19] G. Liu, M. Iwai, and K. Sezaki, "An online method for trajectory simplification under uncertainty of GPS," *Information and Media Technologies*, vol. 8, no. 3, pp. 665–674, 2013.
- [20] W. Pan, C. Yao, X. Li, and L. Shen, "An online compression algorithm for positioning data acquisition," *Informatica*, vol. 38, no. 4, pp. 339–346, 2014.
- [21] G. Yuan, S. Xia, and Y. Zhang, "Interesting activities discovery for moving objects based on collaborative filtering," *Mathematical Problems in Engineering*, vol. 2013, Article ID 380871, 9 pages, 2013.
- [22] G. Trajcevski, H. Cao, P. Scheuermann, O. Wolfsonz, and D. Vaccaro, "On-line data reduction and the quality of history in moving objects databases," in *Proceedings of the 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE '06)*, pp. 19–26, ACM, 2006.
- [23] J. Birnbaum, H.-C. Meng, J.-H. Hwang, and C. Lawson, "Similarity-based compression of GPS trajectory data," in *Proceedings of the 4th International Conference on Computing for Geospatial Research and Application (COM.Geo '13)*, pp. 92–95, San Jose, Calif, USA, July 2013.
- [24] C. Long, R. C. W. Wong, and H. V. Jagadish, "Direction-preserving trajectory simplification," *Proceedings of the VLDB Endowment*, vol. 6, no. 10, pp. 949–960, 2013.
- [25] A. Nibali and Z. He, "Trajic: an effective compression system for trajectory data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 11, pp. 3138–3151, 2015.
- [26] J. Muckell, J. H. Hwang, V. Patil, C. T. Lawson, F. Ping, and S. S. Ravi, "SQUISH: an online approach for GPS trajectory compression," in *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications*, article 13, Washington, DC, USA, May 2011.
- [27] M. Chen, M. Xu, and P. Fränti, "A fast multiresolution polygonal approximation algorithm for gps trajectory simplification," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2770–2785, 2012.
- [28] J. Muckell, P. W. Olsen Jr., J.-H. Hwang, C. T. Lawson, and S. S. Ravi, "Compression of trajectory data: a comprehensive evaluation and new approach," *Geoinformatica*, vol. 18, no. 3, pp. 435–460, 2014.
- [29] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," in *Proceedings of the 1st IEEE International Conference on Data Mining (ICDM '01)*, pp. 289–296, San Jose, Calif, USA, December 2001.
- [30] J. G. Lee, J. Han, and X. Li, "Trajectory outlier detection: a partition-and-detect framework," in *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE '08)*, pp. 140–149, IEEE, Washington, DC, USA, 2008.
- [31] Y. Zheng, "Trajectory data mining: an overview," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 3, article 29, 2015.
- [32] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing multidimensional time-series," *The VLDB Journal*, vol. 15, no. 1, pp. 1–20, 2006.
- [33] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '05)*, pp. 491–502, ACM, June 2005.
- [34] G. Yuan, S. Xia, L. Zhang, Y. Zhou, and C. Ji, "An efficient trajectory-clustering algorithm based on an index tree," *Transactions of the Institute of Measurement and Control*, vol. 34, no. 7, pp. 850–861, 2012.

- [35] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [36] R. Rana, W. Hu, T. Wark, and C. T. Chou, "An adaptive algorithm for compressive approximation of trajectory (AACAT) for delay tolerant networks," in *Wireless Sensor Networks*, vol. 6567 of *Lecture Notes in Computer Science*, pp. 33–48, Springer, Berlin, Germany, 2011.
- [37] J. G. Harper, "Traffic violation detection and deterrence: implications for automatic policing," *Applied Ergonomics*, vol. 22, no. 3, pp. 189–197, 1991.
- [38] M. Karpiriski, A. Senart, and V. Cahill, "Sensor networks for smart roads," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, p. 310, IEEE, Pisa, Italy, 2006.
- [39] N. R. Senner, M. A. Verhoeven, J. M. Abad-Gómez et al., "When Siberia came to the Netherlands: the response of continental black-tailed godwits to a rare spring weather event," *Journal of Animal Ecology*, vol. 84, no. 5, pp. 1164–1176, 2015.
- [40] M. Wikelski, E. Arriero, A. Gagliardo et al., "True navigation in migrating gulls requires intact olfactory nerves," *Scientific Reports*, vol. 5, Article ID 17061, 2015.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

