Northumbria Research Link

A heuristics based global navigation satellite system data reduction algorithm integrated with map-matching

Jing-Xin Dong[1] · Christian Hicks[1,2] · Dongjun Li[3,4]

Abstract

The transmission and storage of global navigation satellite system (GNSS) data places very high demands on mobile networks and centralised data processing systems. GNSS applications including community based navigation and fleet management require GNSS data to be transmitted from a vehicle to a centralised system and then processed by a map-matching algorithm to determine the location of a vehicle within a road segment. Various data compression techniques have been developed to reduce the volume of data transmitted. There is also an independent literature relating to map-matching algorithms. However, no previous research has integrated data compression with a map-matching algorithm that accepts compressed data as an input without the need for decompression. This paper develops a novel GNSS data reduction algorithm with deterministic error bounds, which was seamless integrated with a specifically designed map-matching algorithm. The approach significantly reduces the volume of GNSS data communicated and improves the performance of the map-matching algorithm. The data compression extracts critical points in the trajectory and velocity–time curve of a vehicle. During the process of selecting critical points, the error of restoring vehicle trajectories and velocity–time curves are used as parameters to control the number of critical points selected. By setting different error bound values prior to the execution of the algorithm, the accuracy and volume of reduced data is controlled precisely. The compressed GNSS data, particularly the critical points selected from the vehicle's trajectory is directly input to the map-matching algorithm without the need for decompression. An experiment indicated that the data reduction algorithm is very effective in reducing data volume. This research will be useful in many fields including community driven navigation and fleet management.

Keywords Big data · Data reduction · GNSS data · Map-matching

✉ Jing-Xin Dong
    jingxin.dong@ncl.ac.uk

Extended author information available on the last page of the article

1 Introduction

Big data business analytics can improve the visibility, flexibility and integration of global supply chains and logistics processes, whilst effectively managing demand volatility and cost fluctuations (Genpact 2015; Wang et al. 2016). Big data arises from the widespread adoption of technologies such as GNSS, cell phones, sensors, RFID, social media, video and photos (Wamba et al. 2015). It is characterised by volume, variety and velocity (the 'Three Vs'). There is a tendency for organisations to store vast volumes of data that is too large to be captured, stored or analysed by typical databases (Manyika et al. 2011). There is a variety—data often does not have a fixed structure that is sufficiently ordered for immediate processing. Velocity refers to the dynamic creation of streams of data that may need to be processed in real time (Emani et al. 2015). The major challenge arising from big data is the huge volume of data that leads to the difficulties in data storage, transmission, and processing. Data reduction involves finding useful features to reduce the effective number of variables under consideration, whilst still achieving the goal of the task (Fayyad et al. 1996). Data reduction may be either: lossless, where no information is lost—the compression identifies and eliminates redundancy; or lossy, where some information is lost (Pujar and Kadlaskar 2010). Data reduction may be one of the most effective approaches to resolve difficulties associated with big data since it can reduce the data volume whilst preserving sufficient accuracy to achieve the intended purpose. In this study, data reduction for global navigation satellite system (GNSS) data, which is a very common type of big data was investigated.

The GNSSs currently in operation include the GPS, GLONASS, Galileo, Beidou and other regional systems (European Space Agency 2011). GPS/GNSS has become very affordable and reached the mass market through its use in mobile phones, car navigation, etc. (Stempfhuber and Buchholz 2011). As a result, the amount of GNSS data generated by these systems has become enormous which is causing great difficulties in data transfer, storage and processing. GNSS navigation systems require a GNSS receiver, a digital map and software that can match the user's position with a location on the digital map (Greenfeld 2002). Some approaches match the user's location to the nearest street node, whereas others match the location to a specific location on the travelled street (Greenfeld 2002). Quddus (2006) classified map-matching algorithms into three groups: (i) *geometric*, which consider the shape of the road centre lines (neglecting connections). These algorithms may use point-to-point, point-to-curve or curve-to-curve matching; (ii) *topological,* which takes into account the connectivity of road segments and road attributes such as width and turning restrictions; (iii) *advanced,* which use statistical, mathematical or artificial intelligence approaches. Map-matching is not an easy task as large volumes of data need to be processed in real time.

This research proposes a novel approach that significantly reduces the volume of GNSS data transmitted and processed by the map-matching algorithm. The reduced data is a direct input to a new map-matching algorithm that processes the reduced data without decompression, thus reducing the volume of data transmitted and saving processing time on the centralised server. There are many possible applications of our research, for example:

(1) *community-based traffic and navigation systems* such as WAZE (http://www.waze.com), a smart phone app that allows drivers to share real-time traffic and road information, allowing users to save time and avoid congestion. In this sort of navigation system,

GNSS technology is an important way to collect real-time link travel time between nodes from WAZE users; the collected real-time link travel time is then shared with other users. Travel time estimation is difficult in an urban environment because travel times are inherently uncertain due to fluctuations in traffic volumes, stochastic arrivals at junctions, traffic signals etc. (Zheng and Van Zuylen 2013). The collection of link travel time involves the collection of each user's vehicle trajectory, velocity, and subsequent processing by a map-matching algorithm. WAZE has 90 million active users located throughout the world (https://www.waze.com/brands/drivers/). The computation of individual WAZE user's link travel time requires appropriate data to be transferred, stored and processed by map-matching algorithms in real time. This poses significant challenges in terms of communication, storage and processing;

(2) *vehicle tracking for large fleets* online e-retailing companies such as AMAZON or jd.com in China need to deliver a large volume of products from their warehouses to customers every day over a very large geographical areas. They may have tens of thousands of vehicles in transit. To track the vehicles with GNSS, information including the longitude, latitude and velocity of each vehicle needs to be transmitted to a centralised system in real time. A huge amount of data would be generated if data is transmitted at a frequency of 1, which is commonly used for high frequency GNSS applications (Quddus and Washington 2015). The centralised system would then have to process the data using map-matching algorithms to identify the real-time locations of the vehicles. Furthermore, if a historical record was required for answering queries or for auditing purposes it would be necessary to store the original GNSS data, as well as the output of map-matching algorithm, which would require a massive amount of data storage. It would be very desirable to reduce unnecessary data transmission and storage for the sake of cost-effectiveness.

Reducing the sampling frequency, for example, from 1 to 0.5 Hz could reduce the amount of data transmitted, processed and stored. It would be possible to adapt current map-matching methods to use such compressed GNSS data if the sample interval was not too long; however, one definite disadvantage could be that the error may not have deterministic bounds (Cao et al. 2006) The error would also vary for different GNSS datasets. If the sampling frequency was reduced the error would be quite different for a vehicle travelling at constant velocity compared to one accelerating or decelerating frequently. A field experiment carried out by Quiroga and Bullock (1998) indicated that with a speed variation of 5%, the sampling period for a freeway that they observed needed to be around 4 s, but only 2 s for another highway. Another disadvantage is that crucial GNSS information such as the turning points of vehicles would be more likely to be lost with reduced sampling, which could be problematic for map-matching algorithms.

In light of the above considerations, it is important that data compression methods are perfectly linked with an appropriate map-matching algorithm. Although various generic data compression methods, such as wavelet compression (Hilton et al. 1994) have been well developed and widely used for reducing image, video and audio data, their suitability for GNSS data in a navigation system is poor due to the compressed data for being inappropriate for map-matching. This is because these data compression approaches have not considered the requirements of map-matching algorithms. Even if such methods could be used for GNSS data, a decompression process would be required to restore the GNSS data to its original frequency of 1 Hz in order to satisfy the frequency requirement of current map-matching algorithms (Quddus et al. 2007). In other

words, the GNSS data would first need to be compressed by in-vehicle equipment and then decompressed to 1 Hz GNSS data by a centralised system for processing by a map-matching algorithm. A disadvantage of this approach would be that the decompression procedure would require considerable CPU time.

Many studies relating to GNSS data compression and map-matching have been reported in the literature (Fitriya et al. 2017; Quddus et al. 2007). However, the work relating to GNSS data compression does not consider map-matching and vice versa. In the domain of GNSS data reduction, Cao et al. (2006) first proposed a lossy compression method for spatio-temporal data reduction with deterministic error bounds. Their study demonstrated significant savings in storage. The method begins by formulating the ith GNSS data point represented as (x_i, y_i, t_i). Thus, a vehicle trajectory can be viewed as a function of three variables: coordinates x, y and time t. Then, the vehicle trajectory can be simplified by the Douglas–Peucker line simplification algorithm (Douglas and Peucker 1973). With such a method some GNSS data points crucial for map-matching, such as a vehicle's turning points, are not intentionally identified or preserved. As a consequence, if the compressed data were to be used as an input to a map-matching algorithm, a decompression method would probably be needed as is the case for other generic data compression methods. Gudmundsson et al. (2009) improved the approach developed by Cao et al. (2006) by reducing the computational time required by the data reduction algorithm whilst increasing the number of queries that the compressed GNSS data could support. Both Gudmundsson et al. (2009) and Cao et al. (2006) used the Douglas–Peucker algorithm directly or in modified form with worst case $O(NlogN)$ time complexity improving to $O(N^2)$ for a straightforward implementation (Saalfeld 1999). Chen et al. (2012) devised a bottom–up multiresolution algorithm with polynomial time complexity $O(N)$. Cao and Li (2017) proposed a Directed acyclic graph based Online Trajectory Simplification (DOTS) which has better performance than the Douglas–Peucker algorithm in terms of time complexity and reduced error. The time complexity for DOTS is $O(N^2/M)$ with a cascaded version which has time complexity of $O(N)$. For a comprehensive review of trajectory simplification methods refer to Zhang et al. (2018). There has been no previous research that has considered map-matching in the GNSS data reduction process. This paper addresses this research gap.

There have been many studies related to map-matching including: Zhao (1997), Pyo et al. (2001), Shin and Sung (2001), Taylor et al. (2001), Greenfeld (2002), Gustafsson et al. (2002), Cui and Ge (2003), Ochieng et al. (2003), Yang et al. (2003), Fu et al. (2004), Syed and Cannon (2004), Blazquez and Vonderohe (2005), Chen et al. (2005), El Najjar and Bonnifait (2005), Quddus et al. (2005, 2006), Zhou and Golledge (2006), Velaga et al. (2009), Ochieng et al. (2003), Abdallah et al. (2011), Bierlaire et al. (2013), Li et al. (2013), Knapen et al. (2016) and Knapen et al. (2018). Quddus et al. (2007) pointed out that sampling frequency (termed 'continuity') should be chosen as an important performance parameter. However, most of the current algorithms have not taken it into consideration. None of the map-matching algorithms have considered the use of compressed GNSS data or simplified vehicle trajectories as an input. Thus, there is a no research that has considered the transmission, storage and map-matching in an integrated way. This paper addresses this research gap to provide an approach that: (1) reduces the volume of GNSS data that needs to be transmitted to a centralised system; (2) ensures that the data received at centralised system can be used as an input for a map-matching without the need of data decompression or restoration; (3) enhances the performance of map-matching in terms of running speed and accuracy, whilst simultaneously minimising computing time on the server. The development of an integrated approach that compresses GNSS data

which is directly input into a map-matching algorithm without decompression is a novel contribution.

The work flow used by most current navigation or tracking systems is shown in Fig. 1a. There is no data compression so the volume of data transmitted and processed by the centralised system is relatively high. Figure 1b illustrates the work flow where generic data compression methods or current GNSS data compression methods are used. In this situation, data decompression is required on the centralised system prior to map-matching. The proposed approach, which eliminates decompression is shown in Fig. 1c.

The paper is organised as follows. In the Sect. 2, the data compression method is explained in detail. Then, the development of a specifically designed map-matching algorithm that utilises compressed data in map-matching is outlined. Finally, a field experiment is reported that illustrates the efficiency of the data compression method.

2 Data compression and extraction of critical point

This section proposes a new GNSS data compression method. GNSS data is compressed by selecting critical points on velocity–time curve and spatial vehicle trajectory. Critical points can be defined as the points that have significant impact on the compression error and the performance of map-matching. In other words, two criteria will be considered when selecting critical points among all the GNSS fixings; one is that the average error is limited to a pre-determined bound and the other is that the data compression algorithm can facilitate map-matching or enhance its performance.

In general, each GNSS positioning fixing comprises two kinds of important information: time-varying spatial location and velocity. Although the velocity information of a vehicle can be inferred with time-varying location information, it cannot be viewed as redundant information because the velocity is detected based on Doppler Effect, which is more accurate than when inferred from coordinates and time. Therefore, spatial and velocity information will be considered respectively in the study. This part will focus on: (i) the determination of critical points in velocity–time curves, which are termed *velocity critical*
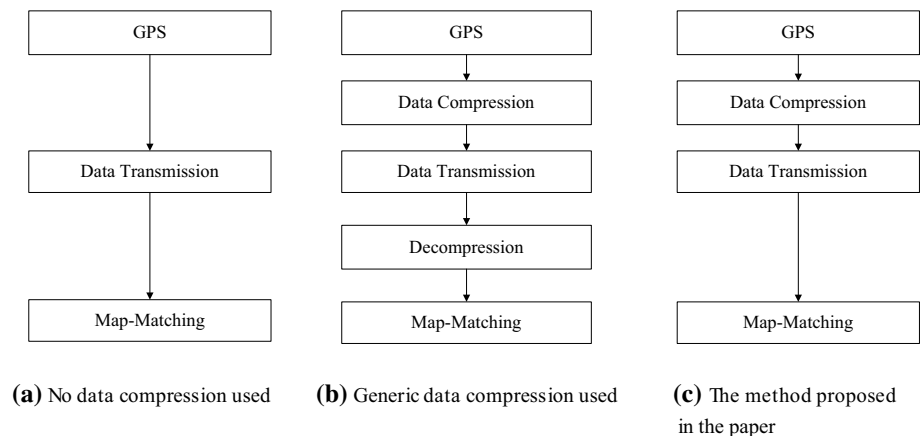

(a) No data compression used   **(b)** Generic data compression used   **(c)** The method proposed in the paper

Fig. 1 Data compression and map-matching

points in this study; and (ii) the criteria used for determining the critical points in a spatial curve termed *critical spatial points*.

## 2.1 Determination of velocity critical points

Normally, a vehicle's velocity is detected every second by a GNSS receiver, which creates an original velocity–time curve. As mentioned before, the transmission of all the data at such a frequency to a computing centre via wireless communication network is not cost-effective; therefore our task is to reduce the communication volume as much as possible within pre-defined error bounds. More specifically, it is necessary to select some of the GNSS points as critical points that can approximate the original time–velocity curve; only these selected critical points will be transmitted to computing centre.

In the study, the sampling error is evaluated based on the difference the average speeds calculated by the original GNSS data and the reduced GNSS data. In order to formulate the error, the following variables are defined. Let i denotes the ith GNSS fixings counted from the previous velocity critical point, thus $i = 1$ denotes the first GNSS fixing, which is also the critical point determined by the previous step. Assume that the Nth GNSS fixing is the next critical point to be extracted, there would be N-2 original GNSS fixings that would not be sent to computing centre. In order to distinguish critical points and inferred points based on the critical points from the original GNSS data points, an apostrophe will be used as a superscript for variables relating to critical points. Let v_1' denote the velocity of the previous critical point, v_N' denote the velocity of the next critical point to be determined, v_i denotes the velocity of ith original GNSS fixing, and v_i' denotes the inferred velocity of GNSS fixing based on the two consecutive critical points Fig. 2 illustrates the sampling of the velocity time curve, for fixings 1 to N. It should be noted that $v_1' = v_1$ and $v_N' = v_N$ in this case because GNSS fixings 1 and N are assumed to be velocity critical points, but $v_i' \neq v_i (i \neq 1, N)$ because the ith ($i \neq 1$ *or* N) GNSS fixing are not a critical points. Therefore, the sampling error can be calculated as follows:

$$\varepsilon_v = \frac{\sum_{i=1}^{N} \left| v_i' - v_i \right|}{N} \tag{1}$$

For the sake of simplicity, v_i' is inferred using a linear interpolation method. It can be determined as follows.

Fig. 2 Sampling of time–velocity curve
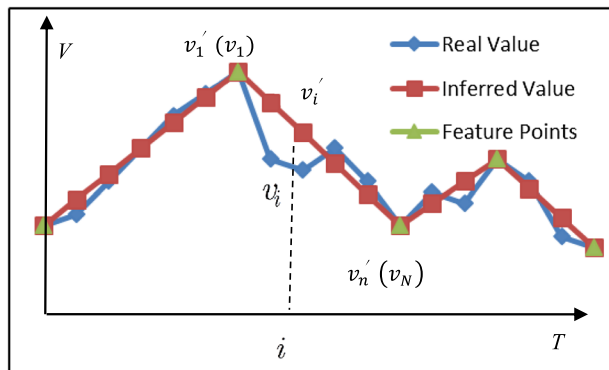
$$v'_i = v'_1 + \frac{v'_N - v'_1}{N-1} \cdot (i-1), (i = 1 \cdots N) \tag{2}$$

If an arbitrary GNSS fixing is selected as a velocity critical point, according to the above equations, the sampling error associated with the selection can be calculated with the following equation:

$$\varepsilon_v = \frac{\sum_{i=1}^{N} \left| v'_1 + \frac{v'_N - v'_1}{N-1} \cdot (i-1) - v_i \right|}{N} \tag{3}$$

Suppose that the allowed maximum sampling error is $\overset{\circ}{\varepsilon}_v$, then the criteria to decide if a GNSS fixing is a critical point can be formulated as follows:

$$\varepsilon_v = \frac{\sum_{i=1}^{N} \left| v'_1 + \frac{v'_N - v'_1}{N-1} \cdot (i-1) - v_i \right|}{N} > \overset{\circ}{\varepsilon}_v \tag{4}$$

The above formulation implies that a GNSS fixing should be selected as a velocity critical point if the sampling error caused by not sending the GNSS fixings from previous velocity critical point to current point exceeds the pre-defined error bounds.

2.2 Determination of spatial critical points

The representation of spatial trajectory mainly involves four types of data: longitude, latitude, heading and time, which can be denoted with (x, y, α, t) respectively. These four variables contain almost all the crucial information required by a map-matching algorithm, therefore this data should be kept. Spatial critical points are selected for the purpose of map-matching, therefore, the data compression algorithm should be able to identify these GNSS fixings that are crucial for map-matching among all the original GNSS fixings. Furthermore, it also needs to ensure the sampling error does not to exceed the predefined error bounds, likewise for the determination of velocity critical points. As a result, two categories of GNSS positioning fixings will be selected as spatial critical points, one for the purpose of map-matching, the other for limiting errors.

For the purpose of map-matching, three categories of GNSS fixings need to be identified as critical points by the algorithm proposed in this study. The first category contains GNSS fixings where the heading of vehicle changes dramatically. These GNSS fixings are normally found at places where a vehicle turns right or left. To identify these GNSS fixings, the concept of approximate curvature is introduced as follows. Assume that $(x_1, y_1, \alpha_1, t_1)$ and $(x_2, y_2, \alpha_2, t_2)$ denote the longitude, latitude and heading of two consecutive original GNSS fixings respectively. Then, approximate curvature k at point $(x_2, y_2, \alpha_2, t_2)$ is defined as follows.

$$k = \frac{|\alpha_2 - \alpha_1|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \tag{5}$$

For the sake of simplicity, in the above formulation, the linear distance between two GNSS points is used to calculate approximate curvature instead of arc length. If the approximate curvature of an arbitrary GNSS fixing k is greater than predefined bound $\overset{\circ}{k}_s$, then the GNSS fixing will be selected as a spatial critical point.

The second category contains the first GNSS fixing found after a time period with signal loss. In an urban area, GNSS receivers often lose the signal, which makes it difficult for the map-matching procedure to identify the correct link. Therefore, keeping the first found GNSS location as a spatial critical point is very important to improve the performance of map-matching algorithm.

The third category includes those GNSS fixings falling in some special regions where there is a high possibility of the map-matching algorithm failing to identify the correct road segment, e.g. fly-over regions, Y-junctions and those areas with a high-density road network.

So far, for most applications, critical GNSS fixings determined with above procedure can satisfy the accuracy requirement by adjusting the threshold level k_s°. However, it has a deficiency that the average error of data compression is not bounded (see Fig. 3). Suppose that a vehicle travels in a circle with a radius equal to or greater than $1/k_s^\circ$, all the GNSS fixings except the start and end points on the trajectory will be ignored because approximate curvature is within the allowed scope. In practice, however, the approximate curvature is normally set to a small value, and the radius of such a circle would be very large, therefore, the probability of a vehicle travelling on such a circle for a long time is very low.

However, some practical applications require strictly keeping the error within the predefined bounds. For this purpose, another additional condition is proposed. Suppose that (x_1', y_1') denotes the most recent critical point. Similar to the above notation method, (x_1', y_1') has the same value as (x_1, y_1). (x_n, y_n) denotes the nth GNSS fixing counted from (x_1', y_1'), and (x_i, y_i) denotes the ith arbitrary GNSS fixing $(1 < i < n)$. If (x_n, y_n) satisfy the following condition, it should be selected as a spatial feature point.

$$\bar{d} = \frac{1}{n-2} \sum_{i=2}^{n-1} \frac{\left| (x_n - x_1)(y_1 - y_i) - (x_1 - x_i)(y_n - y_1) \right|}{\sqrt{(x_n - x_1)^2 + (y_n - y_1)^2}} > d^\circ \qquad (6)$$

The above formulation implies that if the average distance from $(x_i, y_i)(1 < i < n)$ to the line specified by (x_1', y_1') and (x_n, y_n) is greater than the predefined threshold value d°, then (x_n, y_n) should be selected as a spatial critical point; otherwise not.

Fig. 3 The worst situation in selecting spatial critical points
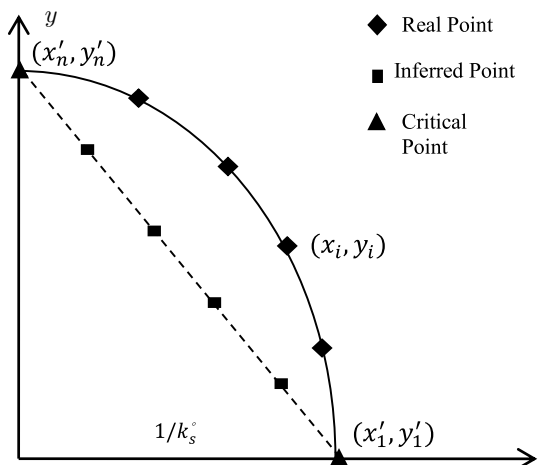
In line with the above discussion, the criteria for selecting a spatial critical point is that either $k > k > k_s^\circ$ or $\bar{d} > d^\circ$.

3 Map-matching with compressed data

The above discussion provides an approach to compress GNSS fixings. The advantage of the proposed data compression method over others is that there is no need for decompression when applied to a specifically designed map-matching algorithm proposed below.

3.1 Definition of error/confidence area

Due to errors arising from both GNSS and map databases, previous map-matching algorithms generally define an error ellipse which contains a set of candidate links on which the vehicle might be travelling. At present, however, we cannot employ the same method to define the error ellipse for each critical point one-by-one. This is because, in our case, the distance between two consecutive critical points may be greater than the radius of the error eclipse, and we consequently need to generate all the possible travel routes that connect the two consecutive sets of candidate links. Because there may be many possible travel routes, this imposes difficulties when attempting to identify the only correct route that connects the two consecutive critical spatial points.

In light of above discussion, we propose to define a rectangular error area with two consecutive critical spatial points. This enables us to form candidate routes in a relatively small area, and facilitate the identification of the correct route for the two consecutive critical spatial points by reducing the number of candidate routes. The reason why we can define the error area in such a way is related to the way we select critical points and the characteristic of errors arising from GNSS and the digitalised map.

As discussed above, the main criterion for selecting critical points on a spatial trajectory is an approximate curvature, which can guarantee that the heading direction of a vehicle between two consecutive spatial critical points does not change dramatically. Therefore, the vehicle trajectory between any two spatial critical points can be assumed to be a straight line. If all the GNSS points between two continuous critical points use the same error variance–covariance matrix to define an error ellipse, then the boundary of the error area will be an external parallelogram with two borders parallel with the line formed by the two critical points (See Fig. 4). In practice, for the sake of simplicity, the map-matching algorithm often searches candidate links within a certain distance; it means that the error ellipse is assumed to be a circle with a large enough radius. Under such circumstances, the error area illustrated in Fig. 5 can be defined as a rectangle. In the figure, P_1 and P_2 are critical points with coordinates (x_1, y_1) and (x_2, y_2) respectively, which thus forms a vector $\overrightarrow{P_1 P_2}$. Let l denote the length of $\overrightarrow{P_1 P_2}$, ϵ the radius of error circus and α the angle between $\overrightarrow{P_1 P_2}$ and x-axis. The vertexes of the rectangular error area are 1, 2, 3, 4 with coordinates (x_1', y_1'), (x_2', y_2'), (x_3', y_3'), (x_4', y_4'), respectively. According to the theory of computational geometry, the coordinates of the four vertexes are given as follows:

$$\begin{bmatrix} x_1' & x_2' & x_3' & x_4' \\ y_1' & y_2' & y_3' & y_4' \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\epsilon & l+\epsilon & l+\epsilon & -\epsilon \\ \epsilon & \epsilon & -\epsilon & -\epsilon \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (7)$$

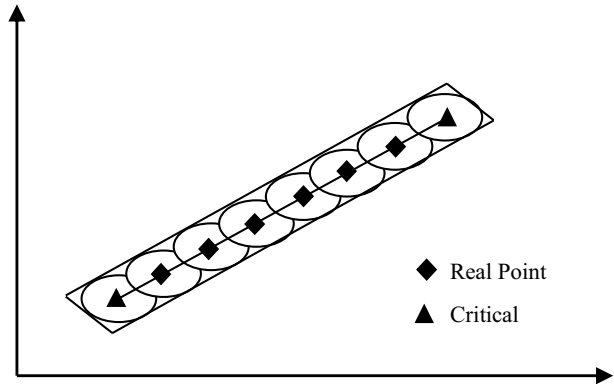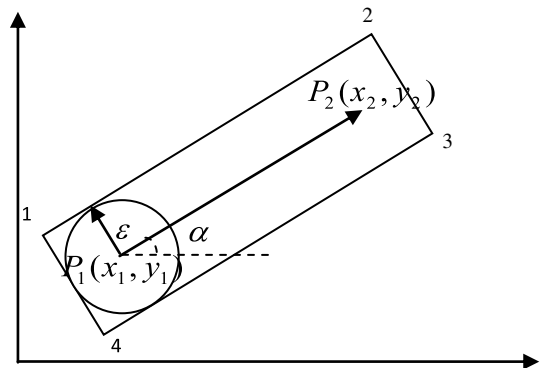**Fig. 4** Error area formed by ellipse between two critical spatial points

Fig. 5 Error area formed with two critical points

3.2 Identification of correct route

Once the error area has been established, all the candidate links therein could be identified by judging whether or not the coordinates of the nodes of each link fall within the error area. Thus, various candidate routes are formed by those candidate links. The analysis in this section demonstrates how to choose the most likely route based on the following steps:

(1)　Filter the candidate links according to the heading of candidate links.

Each link can be described as a vector. For example, consider the link starting from (x_1, y_1) ending at (x_2, y_2), it can be expressed by the vector $\vec{p} = (a_1, a_2)$ where $a_1 = x_2 - x_1$, $a_2 = y_2 - y_1$. Let two vectors $\vec{p_1} = (a_1, a_2)$, $\vec{p_2} = (b_1, b_2)$ denote the vectors for a candidate link and current segment of vehicle trajectory under consideration, respectively. The discrepancy in the directions of the two can be described by the following formula:

$$S_A = \sqrt{\left(\frac{a_1}{\sqrt{a_1^2 + a_2^2}} - \frac{b_1}{\sqrt{b_1^2 + b_2^2}}\right)^2 + \left(\frac{a_2}{\sqrt{a_1^2 + a_2^2}} - \frac{b_2}{\sqrt{b_1^2 + b_2^2}}\right)^2} \qquad (8)$$

The most likely link should be chosen from the links whose heading is consistent with the vehicle's trajectory, which means the following condition must be satisfied:

$$S_A < S° \tag{9}$$

where $S°$ is the predefined threshold value of discrepancy in the vector's direction. The determination of S° is subject to both the sampling error bound relating to spatial critical points, $k°$, and the error relating to digitalised map. The discrepancy between the two normalised vectors with an angle, k° is: $\sqrt{\left(\cos^2 k° - 1\right)^2 + \sin^2 k°}$. Therefore, $S°$ should be no less than $\sqrt{\left(\cos^2 k° - 1\right)^2 + \sin^2 k°}$.

This step filters the candidate links according to the heading of candidate links and thus the scope of the candidate links considered is narrowed.

(2) Identification of the start and end links among candidate links

To identify the start and end links of the vehicle trajectory between two continuous critical points, two constraint conditions must be satisfied. First, the start link must be connected with the matched route; secondly, the distance from the critical points to the possible start and end link should be within the pre-defined scope. Note that the start link and end link could be the same.

(3) Formation of candidate routes

When the start and the end link of the candidate route is determined, the following step selects links among the set of candidate links to form one or several candidate routes. In the implementation of the algorithm, the formation of each candidate route should start from the start link, and the search direction should be consistent with the heading direction of the vehicle. However, a minor deviation should be allowed because system errors and compression error must be considered. The search stops whenever no more links can be found in the set of candidate links, or the next link falls into the set of end links. Note that end link and start link can be the same link.

(4) Filtering of candidate routes

Three criteria can be used to filter the candidate routes obtained in the previous step, to determine the most likely route for the vehicle's trajectory. The first one is that length of a candidate route within the error area should be similar to that of the vehicle trajectory between the two critical points. The second criterion is that candidate route should be very close to the vehicle's trajectory. Generally, the closer the candidate route is to the vehicle's trajectory, the more likely it is that the candidate route is correct. The third criterion is that the selected route should be linked with the matched route.

The second criteria mentioned above involves computing the distance from candidate routes to the segment of vehicle's trajectory between the current pair of spatial critical points. The distance from a critical point (x_i, y_i) to the candidate route with start node (x_i^A, y_i^A) and end node (x_i^B, y_i^B) is calculated based on the following formulation:

$$d = \sqrt{\left(x_i - x_i^A\right)^2 + \left(y_i - y_i^A\right)^2 - \frac{\left[\left(x_i^B - x_i^A\right)\left(x_i - x_i^A\right) + \left(y_i^B - y_i^A\right)\left(y_i - y_i^A\right)\right]^2}{\left(x_i^B - x_i^A\right)^2 + \left(y_i^B - y_i^A\right)^2}} \tag{10}$$

For each segment of a vehicle's trajectory between two spatial critical points, there might be some other critical points used to transmit the velocity curve. All these critical points should be used to measure the distance between candidate routes and the vehicle's trajectory, and the average of these distances should be used as the distance between the candidate link and the segment of vehicle trajectory.

3.3 Determination of the vehicle location on the selected link

So far, we have determined the most likely candidate routes with each consisting of a number of links for the trajectory between two continuous critical points based on the previous steps. The following step determines the link on which the vehicle is running and the physical location of the vehicle on that link, which is also the final aim of a general map-matching algorithm.

In order to reduce the error during the procedure, a vehicle's trajectory will be translated, rotated and scaled segment by segment. Suppose that vector $\vec{AB}$ has a start point $A(x_A, y_A)$ at the beginning of the start link and the end point $B(x_B, y_B)$ of the end link on the route identified in Sect. 3.2. Assume that $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ is the link to point B, then P_1 and P_2 form another vector $\vec{P_1P_2}$. Let $\Delta\alpha$ denote the angle between $\vec{AB}$ and $\vec{P_1P_2}$, and $\lambda = |\vec{AB}|/|\vec{P_1P_2}|$, then all critical points (x, y) between $\vec{P_1P_2}$, inclusive, should be transformed with the following formula:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Delta\alpha & -\sin\Delta\alpha & x_A(1 - \cos\Delta\alpha) + y_A\sin\Delta\alpha \\ \sin\Delta\alpha & \cos\Delta\alpha & y_A(1 - \cos\Delta\alpha) + x_A\sin\Delta\alpha \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \lambda & 0 & x_A(1-\lambda) \\ 0 & \lambda & y_A(1-\lambda) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & x_A - x_1 \\ 0 & 1 & y_B - y_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{11}$$

The number of links that comprise the route might be greater than one because the vehicle might be travelling on different links between two spatial critical points. Under this situation, the vehicle trajectory should be split at some proper dividing point where a vehicle is travelling to another link. These dividing points are determined by constructing a perpendicular line to a vehicle's trajectory through the nodes of the link, and the perpendicular foot of the vehicle trajectory is the dividing point.

In terms of the final results, there are two choices, either of which can be taken based on the purpose of the application. For some applications, such as the estimation of link travel time, the final result needed is only to identify the proper link for each critical point. So far, the link each critical point belongs to has been determined, thus the link travel time can be estimated by averaging the velocity of each critical point corresponding to the link. For some other applications, it might be necessary to identify the link and the correct coordinates for a total set of GNSS data with a

frequency of 1 Hz like most map-matching algorithms do. In this situation, we should decompress the GNSS data first; each decompressed GNSS data point has a link based on the dividing points. Then, a perpendicular line to the corresponding link can be constructed from each new decompressed GNSS data point. The coordinates of the perpendicular foot is the corrected coordinate for the corresponding GNSS data point. A set of new GNSS data points with corrected coordinates is thus obtained. Please note the above decompression is required after map-matching for some special applications. This is an optional process. As a contrast, the current map-matching reported in the literature require data decompression before the start of map-matching and it is a necessary process.

4 Experiments

The above algorithm was tested using a GNSS data set collected in Beijing, which contains 44,000 s of GNSS data.

A segment of velocity curve with critical points is shown in Fig. 6. The curve in the figure is drawn based on 1 Hz GNSS data, the dotted points in the figure are critical points determined based on sampling strategy of velocity proposed as above. The allowed error was set as 0.5 m/s, and the sampling rate was 6.6%.

In Fig. 7, a segment of vehicle trajectory with spatial critical points is shown. There were 5082 GNSS data points originally; however, only 182 spatial critical points among them are needed to keep the curve when minor error is allowed. The sampling rate for this curve was about 3.5%.

It can be observed that only about 10.1% of GNSS data was needed to be transmitted, when minor error was allowed using the proposed sampling strategy. In a real application, the sampling algorithm can be implemented through in-vehicle GNSS equipment. The potential benefits for the algorithm to reduce data communication is very large, which is beneficial when the data is transmitted via a wireless

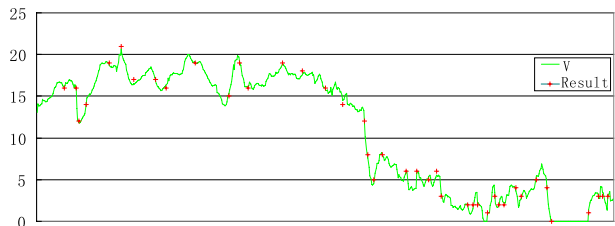**Fig. 6** A segment of velocity curve with velocity critical points

Fig. 7 A segment of vehicle trajectory with spatial critical points

communications system. Furthermore, the compressed GNSS data was easily used in the map-matching without the need of decompression.

5 Conclusion and further study

Various data compression techniques have been developed to reduce the volume of data transmitted. There is also an independent literature relating to map-matching algorithms. However, no previous research has integrated data compression with a map-matching algorithm that accepts compressed data as an input without the need for decompression.

In this paper, we have presented a data reduction algorithm for GNSS data transmission. Since it is common to link GNSS data with a digitalised map in practical applications such as navigation or vehicle monitoring, we also designed a curve-to-curve map-matching algorithm for the compressed GNSS data. It is worth mentioning, the compressed data does not need to be de-compressed when they are fed into map-matching algorithm. Therefore, significant computational time could be saved. Our numerical experiment indicates our data compression algorithm is very efficient and it can reduce the GNSS data transmission volume by 79.9%.

Although the proposed data reduction algorithm has demonstrated good performance in reducing the GNSS data volume to be transmitted, our proposed algorithm is constrained by the precision of GNSS equipment and digitalised maps. For instance, our proposed algorithm may be unable to identify a vehicle's precise position when lane changing. To address the issue, further work is being undertaken to improve the algorithm.

References

Abdallah, F., Nassreddine, G., & Denoeux, T. (2011). A multiple-hypothesis map-matching method suitable for weighted and box-shaped state estimation for localization. *IEEE Transactions on Intelligent Transportation Systems, 12*(4), 1495–1510. https://doi.org/10.1109/tits.2011.2160856.

Bierlaire, M., Chen, J., & Newman, J. (2013). A probabilistic map matching method for smartphone GPS data. *Transportation Research Part C: Emerging Technologies, 26,* 78–98. https://doi.org/10.1016/j.trc.2012.08.001.

Blazquez, C. A., & Vonderohe, A. P. (2005). Simple map-matching algorithm applied to intelligent winter maintenance vehicle data. *Transportation Research Record, 1935,* 68–76.

Cao, W., & Li, Y. (2017). DOTS: An online and near-optimal trajectory simplification algorithm. *Journal of Systems and Software, 126,* 34–44. https://doi.org/10.1016/j.jss.2017.01.003.

Cao, H., Wolfson, O., & Trajcevski, G. (2006). Spatio-temporal data reduction with deterministic error bounds. *VLDB Journal, 15*(3), 211–228. https://doi.org/10.1007/s00778-005-0163-7.

Chen, W., Li, Z., Yu, M., & Chen, Y. (2005). Effects of sensor errors on the performance of map matching. *The Journal of Navigation, 58*(2), 273–282.

Chen, M., Xu, M., & Franti, P. (2012). A fast $ o (n) $ multiresolution polygonal approximation algorithm for GPS trajectory simplification. *IEEE Transactions on Image Processing, 21*(5), 2770–2785.

Cui, Y., & Ge, S. S. (2003). Autonomous vehicle positioning with GPS in urban canyon environments. *IEEE Transactions on Robotics and Automation, 19*(1), 15–25.

Douglas, D. H., & Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization, 10*(2), 112–122.

El Najjar, M. E., & Bonnifait, P. (2005). A road-matching method for precise vehicle localization using Belief Theory and Kalman filtering. *Autonomous Robots, 19*(2), 173–191. https://doi.org/10.1007/s10514-005-0609-1.

Emani, K. C., Cullot, N., & Nicolle, C. (2015). Understandable big data: A survey. *Computer Science Review, 17,* 70–81. https://doi.org/10.1016/j.cosrev.2015.05.002.

European Space Agency. (2011). *GNSS Signal*. https://gssc.esa.int/navipedia/index.php/GNSS_signal. Accessed 1 Jan 2019.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). Knowledge discovery and data mining: Towards a unifying framework. *KDD, 96,* 82–88.

Fitriya, L. A., Purboyo, T. W., & Prasasti, A. L. (2017). A review of data compression techniques. *International Journal of Applied Engineering Research, 12*(19), 8956–8963.

Fu, M., Li, J., & Wang, M. (2004). A hybrid map matching algorithm based on fuzzy comprehensive judgment. In *IEEE conference on intelligent transportation systems, Washington, DC, 3rd–6th October 2004* (pp. 613–617).

Genpact (2015). Driving supply chain excellence through Data-to-Action Analytics. https://www.genpact.com/downloadable-content/insight/driving-supply-chain-excellence-through-data-to-action-analytics.pdf. Accessed 1 Jan 2019.

Greenfeld, J. S. (2002). Matching GPS observations to locations on a digital map. In *81th Annual meeting of the transportation research board* (Vols. 1, 3, pp. 164–173).

Gudmundsson, J., Katajainen, J., Merrick, D., Ong, C., & Wolle, T. (2009). Compressing spatio-temporal trajectories. *Computational Geometry, 42*(9), 825–841. https://doi.org/10.1016/j.comgeo.2009.02.002.

Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., et al. (2002). Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing, 50*(2), 425–437. https://doi.org/10.1109/78.978396.

Hilton, M. L., Jawerth, B. D., & Sengupta, A. (1994). Compressing still and moving images with wavelets. *Multimedia Systems, 2*(5), 218–227.

Knapen, L., Bellemans, T., Janssens, D., & Wets, G. (2018). Likelihood-based offline map matching of GPS recordings using global trace information. [Article]. *Transportation Research Part C: Emerging Technologies, 93,* 13–35. https://doi.org/10.1016/j.trc.2018.05.014.

Knapen, L., Hartman, I. B. A., Schulz, D., Bellemans, T., Janssens, D., & Wets, G. (2016). Determining structural route components from GPS traces. *Transportation Research Part B: Methodological, 90,* 156–171. https://doi.org/10.1016/j.trb.2016.04.019.

Li, L., Quddus, M., & Zhao, L. (2013). High accuracy tightly-coupled integrity monitoring algorithm for map-matching. *Transportation Research Part C: Emerging Technologies, 36,* 13–26. https://doi.org/10.1016/j.trc.2013.07.009.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., et al. (2011). *Big data: The next frontier for innovation, competition, and productivity*. New York: McKinsey Global Institute.

Ochieng, W. Y., Quddus, M. A., & Noland, R. B. (2003). Map-matching in complex urban road networks. *Brazilian Journal of Cartography, 55*(2), 1–14.

Pujar, J. H., & Kadlaskar, L. M. (2010). A new lossless method of image compression and decompression using huffman coding techniques. *Journal of Theoretical and Applied Information Technology, 15*(1), 18–23.

Pyo, J. S., Shin, D. H., & Sung, T. K. (2001). Development of a map matching method using the multiple hypothesis technique. In *IEEE intelligent transportation systems conference, Oakland, USA, 25th–29th August 2001* (pp. 23–27).

Quddus, M. A. (2006). *High integrity map matching algorithms for advanced transport telematics applications*. PhD, Imperial College London, London.

Quddus, M. A., Noland, R. B., & Ochieng, W. Y. (2005). Validation of map matching algorithms using high precision positioning with GPS. *The Journal of Navigation, 58*(2), 257–271.

Quddus, M. A., Noland, R. B., & Ochieng, W. Y. (2006). A high accuracy fuzzy logic based map matching algorithm for road transport. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, 10*(3), 103–115. https://doi.org/10.1080/15472450600793560.

Quddus, M. A., Ochieng, W. Y., & Noland, R. B. (2007). Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies, 15*(5), 312–328. https://doi.org/10.1016/j.trc.2007.05.002.

Quddus, M., & Washington, S. (2015). Shortest path and vehicle trajectory aided map-matching for low frequency GPS data. *Transportation Research Part C: Emerging Technologies, 55,* 328–339. https://doi.org/10.1016/j.trc.2015.02.017.

Quiroga, C. A., & Bullock, D. (1998). Travel time studies with global positioning and geographic information systems: An integrated methodology. *Transportation Research Part C: Emerging Technologies, 6C*(1–2), 101–127. https://doi.org/10.1016/s0968-090x(98)00010-2.

Saalfeld, A. (1999). Topologically consistent line simplification with the Douglas-Peucker algorithm. *Cartography and Geographic Information Science, 26*(1), 7–18. https://doi.org/10.1559/1523040997 82424901.

Shin, D. H., & Sung, T. K. (2001). Analysis of positioning errors in radionavigation systems. In *IEEE intelligent transportation systems conference, Oakland, USA, 25th–29th August 2001* (pp. 156–159).

Stempfhuber, W., & Buchholz, M. (2011). A precise, low-cost RTK GNSS system for UAV applications. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science, 38,* 1-C22.

Syed, S., & Cannon, M. E. Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons. In *ION National technical meeting, San Diego, CA, 2004* (Vol. 1, pp. 26–28).

Taylor, G., Blewitt, G., Steup, D., Corbett, S., & Car, A. (2001). Road reduction filtering for GPS-GIS navigation. *Transactions in GIS, 5*(3), 193–207.

Velaga, N. R., Quddus, M. A., & Bristow, A. L. (2009). Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems. *Transportation Research Part C: Emerging Technologies, 17*(6), 672–683. https://doi.org/10.1016/j.trc.2009.05.008.

Wamba, F. S., Akter, S., Edwards, A., Chopin, G., & Gnanzou, D. (2015). How 'big data' can make big impact: Findings from a systematic review and a longitudinal case study. *International Journal of Production Economics, 165,* 234–246. https://doi.org/10.1016/j.ijpe.2014.12.031.

Wang, G., Gunasekaran, A., Ngai, E. W. T., & Papadopoulos, T. (2016). Big data analytics in logistics and supply chain management: Certain investigations for research and applications. *International Journal of Production Economics, 176,* 98–110. https://doi.org/10.1016/j.ijpe.2016.03.014.

Yang, D., Cai, B., & Yuan, Y. (2003). An improved map-matching algorithm used in vehicle navigation system. In 2003 (Vol. 2, pp. 1246–1250). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/itsc.2003.1252683.

Zhang, D., Ding, M., Yang, D., Liu, Y., Fan, J., & Shen, H. T. (2018). Trajectory simplification: An experimental study and quality analysis. *Proceedings of the VLDB Endowment, 11,* 934–946.

Zhao, Y. (1997). *Trajectory simplification: an experimental study and quality analysis.* Norwood, MA: Artech House Inc.

Zheng, F., & Van Zuylen, H. (2013). Urban link travel time estimation based on sparse probe vehicle data. *Transportation Research Part C: Emerging Technologies, 31,* 145–157. https://doi.org/10.1016/j.trc.2012.04.007.

Zhou, J., & Golledge, R. (2006). A three-step general map matching method in the GIS environment: Travel/transportation study perspective. *International Journal of Geographical Information System, 8*(3), 243–260.

Affiliations

Jing-Xin Dong[1] (iD) **· Christian Hicks[1,2] · Dongjun Li[3,4]**

Christian Hicks
chris.hicks@ncl.ac.uk

Dongjun Li
d.li20@ncl.ac.uk

[1]   Business School, Newcastle University, 5 Barrack Road, Newcastle upon Tyne NE1 4SE, UK

[2]   Department of Management Studies, Indian Institute of Technology Delhi, New Delhi 110016, India

[3]   School of Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, UK

[4]   Institute of Logistics, Infrastructure, Supply and Transport (LIST), Innovation Centre, The University of Northampton, Green Street, Northampton NN1 1SY, UK