International Journal of Geo-Information

MDPI

Article

Map Matching for Urban High-Sampling-Frequency GPS Trajectories

Minshi Liu [1,2], Ling Zhang [1,3,4], Junlian Ge [1,3,4,*], Yi Long [1,3,4,*] and Weitao Che [5,6]

1 Key Laboratory of Virtual Geographic Environment (Nanjing Normal University), Ministry of Education, Nanjing 210023, China; liuminshi@chzu.edu.cn (M.L.); lingzhang@njnu.edu.cn (L.Z.)
2 School of Geographic Information and Tourism, Chuzhou University, Chuzhou 239000, China
3 State Key Laboratory Cultivation Base of Geographical Environment Evolution (Jiangsu Province), Nanjing 210023, China
4 Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China
5 Key Laboratory of Guangdong for Utilization of Remote Sensing and Geographical Information System, Guangdong Open Laboratory of Geospatial Information Technology and Application, Guangzhou Institute of Geography, Guangzhou 510070, China; cheweitao@gdas.ac.cn
6 Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
* Correspondence: gejunlian@njnu.edu.cn (J.G.); longyi@njnu.edu.cn (Y.L.)

Received: 10 December 2019; Accepted: 30 December 2019; Published: 5 January 2020

Abstract: As a fundamental component of trajectory processing and analysis, trajectory map-matching can be used for urban traffic management and tourism route planning, among other applications. While there are many trajectory map-matching methods, urban high-sampling-frequency GPS trajectory data still depend on simple geometric matching methods, which can lead to mismatches when there are multiple trajectory points near one intersection. Therefore, this study proposed a novel segmented trajectory matching method in which trajectory points were separated into intersection and non-intersection trajectory points. Matching rules and processing methods dedicated to intersection trajectory points were developed, while a classic "Look-Ahead" matching method was applied to non-intersection trajectory points, thereby implementing map matching of the whole trajectory. Then, a comparative analysis between the proposed method and two other new related methods was conducted on trajectories with multiple sampling frequencies. The results indicate that the proposed method is not only competent for intersection matching with high-frequency trajectory data but also superior to two other methods in both matching efficiency and accuracy.

Keywords: map matching; GPS trajectory; high sampling frequency; road network

1. Introduction

Due to the popularity of mobile positioning devices, a significant volume of trajectory data with various types is generated. Furthermore, big data analysis and increasing location-based service applications have made mobile trajectory processing, analysis, and application a focus area of current research. Trajectory data acquisition depends on different positioning devices that vary in terms of accuracy errors, where the trajectories deviate from the original road or points of interest. Therefore, map matching is required before processing and analyzing trajectory data [1]. Trajectory map-matching is also required to add semantic information to trajectory data and attach geographic ground information to trajectories.

In the past few decades, many map matching methods have been proposed. These methods can be divided into geometric, topological, and advanced methods [1], or they can be divided into local and global methods [2]. For different application scenarios, there are real-time trajectory map

Patent Owner Ex. 2005, p. 1
Tesla v. iQar, IPR2024-00543

matching methods, offline trajectory map matching methods, GPS trajectory map matching methods, other location trajectory map matching methods, low-sampling trajectory map matching methods, high-sampling trajectory map matching, urban trajectory map matching methods, and indoor trajectory map matching methods [3,4]. However, there are few studies on high-frequency sampling trajectories in urban road networks.

Currently, GPS data are acquired at short intervals, such as 1 or 5 s, in urban areas because of advancements in GPS technology and big data processing technology development [5]. This short interval has led to massive high-sampling-frequency trajectory data for vehicles (e.g., buses and taxis) and pedestrians. Because usually moving objects stop or move at a low speed at road intersections, there are multiple trajectory points at road intersections. Moreover, because of the matching complexity of trajectory points at road intersection, there will be a higher likelihood of mismatches. An incorrect matching example is shown in Figure 1.

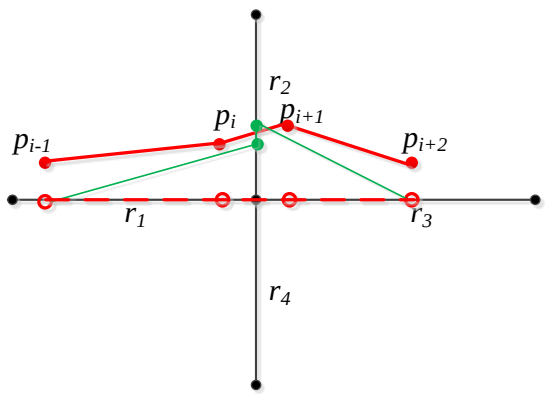


Figure 1. Incorrect matching of intersection trajectory points. Points p_i and p_{i+1} are matched to r_2 using other methods. However, p_i should be matched to r_1, and p_{i+1} should be matched to r_3.

For the high-resolution GPS data, the author proposes a global matching method that first segments and then merges [6]. This method can balance efficiency and accuracy, but cannot deal with the matching error of the trajectory points at the road intersection. Wang et al. [7] proposed a method combining the junction decision domain with the hidden Markov model. While the method improves the matching accuracy of the trajectory points at the road intersection, the matching efficiency of the method is low. Therefore, this method is not suitable for high sampling frequency data.

To address the problem of intersection trajectory mapping, this study proposes a segmented trajectory matching method. Firstly, the trajectory is interrupted at the road intersection position, and it is divided into a set of intersection trajectory segments (including intersection trajectory points) and non-intersection trajectory segments (excluding intersection trajectory points). Secondly, dedicated matching rules and processing methods are proposed for the intersection segment, and the matching of the non-intersection trajectory segment is done using a classic "Look-Ahead" matching method [8]. Finally, map matching for the entire trajectory is successful.

2. Related Works

Currently, there are two primary map-matching methods: Local matching and global matching.

2.1. Local Matching Methods

Local matching algorithms follow a greedy strategy of sequentially extending the solution from an already matched portion [9]. The key to such local matching methods is to find a locally optimal point or segment on a road network. The most commonly used local matching method is the geometry-based method [10,11], where trajectory matching is made based on constraints such as distance and direction. It features a favorable matching effect for high-sampling-frequency trajectories (one trajectory point

Patent Owner Ex. 2005, p. 2
Tesla v. iQar, IPR2024-00543

or more can be matched on one road), but it has difficulty ensuring a high matching accuracy for low-sampling-frequency trajectories. To enhance matching accuracy, some new methods have been developed, such as topology map matching [5,8], spatial-temporal feature-based map matching [2,12], and weight-based map matching [13–17]. In a study by Brakatsoulas et al. [8], an incremental matching method has been proposed using the "Look-Ahead" matching strategy. With this method, a topological relationship between the road matched by the subsequent point and that by the current point is established to correct the road matched by the current point. Wang et al. [5] propose a Kalman filter based correcting algorithm to improve the matching accuracy of the traditional topological algorithm on the complicated road sections, such as intersections and parallel roads. They also use a parallelized map-matching algorithm to improve the processing efficiency of the map matching. Lou et al., have proposed a spatio-temporal map-matching algorithm for low-sampling-rate GPS trajectories [2]. The authors model the temporal analysis with speed and travel time data to improve its accuracy. Hsueh and Chen have proposed a similar approach—STD-matching—which adds the real-time direction factor to ST-matching [12].

In recent years, more weight-based map matching methods have been proposed. Hashemi and Karimi [14] propose a dynamic weight-based map-matching algorithm. Its factors are composed of distance between the GPS point and road segments, difference between the heading of the GPS point and direction of road segments, and difference between the direction of consecutive GPS points and direction of road segments. Its dynamic weights are calculated from positional accuracy, speed, and traveled distance from previous GPS points. Sharath et al. [15] also establish four influencing factors of GPS point matching, proximity, kinematic, turn intent prediction, connectivity, and then develop a new dynamic two-dimensional weight-based map-matching algorithm by incorporating dynamic weight coefficients and road width to enable the lane-level identification. Hu et al. [16] propose an information fusion (IF) matching method based on the moving-object-related meta-information, which includes four fields: Location, speed, direction, and timestamp. This method has a better effect on handling ambiguous cases. Zhao et al. [17] use the speed, bearing difference, perpendicular distance, and spatial correlation as the influence factor of GPS point matching. They dynamically estimate the weight of each factor based on the Dempster–Shafer theory.

Overall, the local methods only consider a few points adjacent to the point to be matched, it runs fast and performs well when the sampling frequency is very high (e.g., 2–5 s) [2]. However, as the sampling frequency decreases, its matching accuracy will decrease significantly. While some recent methods also improve the matching accuracy of low- and medium-frequency data, such methods are more suitable for high-frequency or medium-frequency data than global methods.

2.2. Global Matching Methods

Comparatively, global matching methods aim to identify a road network similar to the trajectory based on all trajectory points of the whole trajectory section, and then try to find a trajectory that is as close as the sampling track among all available trajectories in the road network [9]. The similarities among multiple line segments are measured using Frechet distance [8,18–20], using long common subsequence (LCS) [6], or using the likelihood function [21–23] in global matching methods. Yin and Wolfson [19] plot a network map using the Frechet distance among relative trajectories as the weight of a road section. Besides, Dijkstra's shortest path algorithm has been used for the shortest path calculation to obtain the final matched road. Brakatsoulas et al. [8] propose the concept of average Frechet distance to identify the overall path using the free space diagram of the relative trajectories of various road sections. Zhu et al. [6] generated candidate matched paths for the entire trajectory using a separate trajectory into segments and then got the best-matched path based on the LCS. Millard-Ball et al. [21] use a three-part quasi-likelihood function, which is composed of geometric likelihood, topologic likelihood, and temporal likelihood, to get the best match from the candidate set. Knapen et al. [22] first divide the trace of GPS into chronologically ordered and then find the maximum likelihood of partial routes based on an acyclic directed graph. Moreover, Rappos et al. [23] have

Patent Owner Ex. 2005, p. 3
Tesla v. iQar, IPR2024-00543

proposed a force-directed map matching method, which uses an attractive force model according to the distance and the angle between the GPS point and the road edge, and the length of the road edge.

Other research is based on the hidden Markov model (HMM) for map matching. Newson and Krumm [24] propose an HMM map matching for location data with noise and sparseness. Since their research, many studies have improved on this method. Koller et al. [25] propose fast map matching (FMM) based on HMM which replaces the Viterbi algorithm with a bidirectional Dijkstra and employs a lazy evaluation to reduce the number of costly route calculations. Yang et al. [26] also present a fast map matching, an algorithm integrating hidden Markov model with precomputation. Qi et al. [7] put forward a junction decision domain model, which is used to improve the map-matching algorithm based on the HMM. It effectively reduces the error rate of junction matching. In addition, in real-time matching, the HMM is also used more frequently. A new incremental map-matching algorithm based on HMM is proposed for real-time matching [27,28]. For inaccurate and sparse location data, Jagadeesh and Srikanthan [29] offer a novel map-matching solution that combines the widely-used approach based on the HMM with the concept of drivers' route choice. Algizawy et al. [30] extend the typical HMM used in map matching to accommodate for highly sparse mobile phone data by an adaptive probability.

Generally, the global methods have higher matching accuracy than the local method, especially for low-sampling-frequency trajectories (e.g., the time interval is higher than 30 s). The reason is that the global method can find the correct matching road section from a global perspective when there is a loss of road information between the matching road sections of adjacent trajectory points. However, global matching methods are more complex and have lower matching efficiency than local matching methods.

Therefore, this paper adopts the local matching strategy to improve the matching efficiency of high-frequency data and uses the intersection segment matching method to improve the matching accuracy of intersection matching points to achieve the purpose of matching efficiency and accuracy for high-sampling trajectory data in urban roads.

3. Intersection Trajectory Segment Matching

3.1. Classification of Intersection Trajectory Point

The intersection trajectory segment is composed of a series of intersection trajectory points. Given the complexity of trajectories at intersections, it is necessary to classify the spatial relations between the trajectory and the intersection. To this end, some related concepts need to be defined as follows.

Definition 1. ***(Road network).*** *This is a network structure made up of road network nodes and edges. A road network edge starts and ends according to road network nodes. Moreover, one road network node must be the starting or ending point of a road network.*

Definition 2. ***(Intersection).*** *This refers to a road network node consisting of the spatial location of the node and the topological relationship between the node and the related road network edges.*

Definition 3. ***(Road section).*** *This is a road network edge strung by intersections and coordinate points making up this edge.*

Definition 4. ***(Intersection trajectory points).*** *This is a trajectory point set adjacent to the intersection. Due to errors in trajectory data acquisition, intersection trajectory points in this study are all trajectory points falling within the circular area centering on the intersection point at a radius of the acquisition error. This is represented by Equation (1):*

$$P = \left\{ p_i(x_i, y_i) | \sqrt{(x_i - x_o)^2 + (y_i - y_o)^2} \leq \varepsilon \right\} \tag{1}$$

Patent Owner Ex. 2005, p. 4
Tesla v. iQar, IPR2024-00543

where P is the set of intersection trajectory points; (x_i, y_i) and (x_o, y_o) are the coordinates of trajectory point p_i and the intersection node, respectively; and ε is the error radius.

Definition 5. ***(Intersection trajectory segment).*** *This is a trajectory segment made up of intersection trajectory points by sequence.*

Definition 6. ***(Inbound point).*** *This is the first point of the intersection trajectory points.*

Definition 7. ***(Outbound point).*** *This is the last point of the intersection trajectory points.*

Definition 8. ***(Inbound road section).*** *This is a matched road section before the trajectory enters the intersection, and it should be the matched road with the inbound point.*

Definition 9. ***(Outbound road section).*** *This is a matched road section after the trajectory moves from the intersection, and it should be the matched road with the outbound point.*

Figure 2 illustrates examples of the above concepts.

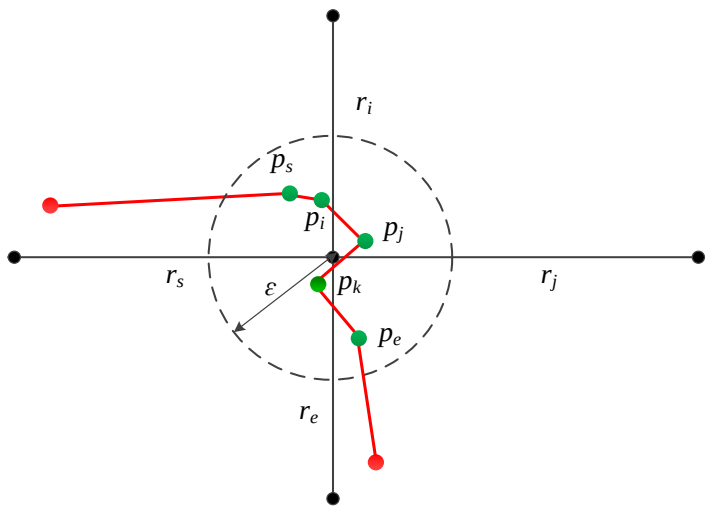


Figure 2. Examples of related concepts. There are four road sections, r_s, r_e, r_i, r_j, which are connected in an intersection o, and they form a road network. ε is the radius of the acquisition error. Trajectory points from p_s to p_e are intersection trajectory points, and they are connected as intersection trajectory segments. p_s and p_e are inbound and outbound points, respectively. r_s is an inbound road section, and r_e is an outbound road section.

Next, intersection trajectory point matching is needed. There are three places that intersection trajectory points should match with: The inbound road section, the outbound road section, and the intersection. Therefore, relations between intersection trajectory points and the intersection and intersection-related road sections could be classified as long as the inbound and outbound sections of the trajectory at the intersection are determined. They are classified into the following four types:

- Type 1 (Inside point). The intersection trajectory point is located within the angle between the inbound road section and the outbound road section, such as point p_k in Figure 2.
- Type 2 (Inbound-road-related point). The intersection trajectory point is located within the angle between the inbound road section and any other road section, except the outbound section, such as points p_s and p_i in Figure 2.
- Type 3 (Outbound-road-related point). The intersection trajectory point is located within the angle between the outbound road section and any other road section, except the inbound section, such as point p_e in Figure 2.

Patent Owner Ex. 2005, p. 5
Tesla v. iQar, IPR2024-00543

- Type 4 (Outside point). The intersection trajectory point is located within the angle between the other two road sections, except the inbound and outbound sections, such as point p_j in Figure 2.

These four types of trajectory points cover the relations between the intersections and road sections at all trajectory points. Based on the different relationships, the road sections or intersections can be matched.

3.2. Matching Rules

According to the intersection trajectory point classification described above, there are four types of relations between trajectory points and the intersection and intersection-related road sections. Considering that the matched trajectory should be consistent with the inbound and outbound sections, the positions' intersection trajectory points are matched with the inbound section, the outbound section, and the intersection only. Thus, the following matching rules are made targeting the four abovementioned types:

- Rule I: An inside point is matched using the shortest distance method.
- Rule II: An inbound-road-related point is matched to the inbound road section.
- Rule III: An outbound-road-related point is matched to the outbound road section.
- Rule IV: An outside point is matched to the intersection.

In addition, the inbound point is matched to the inbound road section, and the outbound point is matched to the outbound road section. However, when there is only one point in the intersection trajectory points, the point is the inbound point and also the outbound point, so the point should be matched by the above four matching rules.

As shown in Figure 3, since point p_1 is located within the angle between the inbound road section r_1 and any other road section r_2, p_1 is an inbound-road-related point, and is directly matched to r_1. Since point p_2 is located within the angle between the other two road sections r_2 and r_3, p_2 is an outside point and is matched to the intersection o. Since point p_3 is located within the angle between the inbound road section r_1 and the outbound road section r_4, point p_3 is an inside point and is matched to r_4 by calculating the shortest distance from p_3 to r_1, and that from p_3 to r_4. Since point p_4 is located within the angle between the outbound road section r_4 and any other road section r_3, p_4 is an outbound-road-related point and is directly matched to r_4.

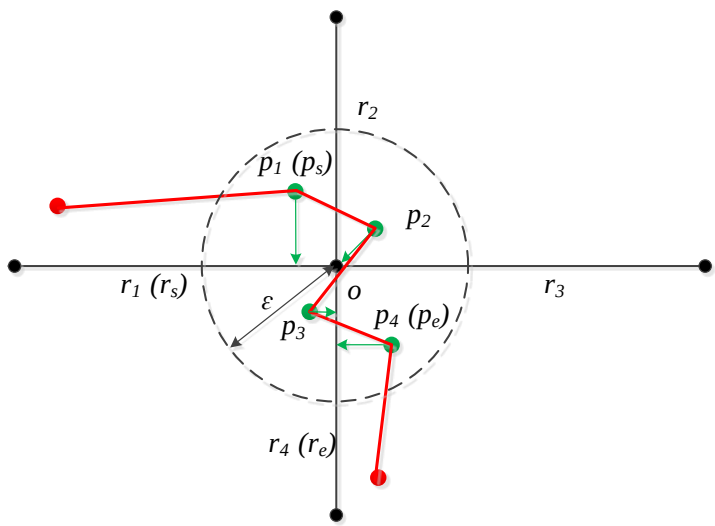


Figure 3. Classification and matching of intersection trajectory points. There are four road sections, r_1, r_2, r_3, r_4, which are connected in an intersection o. r_1 is the inbound road sections (r_s) and r_4 is outbound road sections (r_e). $\{p_1, p_2, p_3, p_4\}$ are the intersection trajectory points. p_1 and p_4 are the inbound point (p_s) and outbound point (p_e), respectively.

Patent Owner Ex. 2005, p. 6
Tesla v. iQar, IPR2024-00543

3.3. Abnormity Adjustment

Errors, however, could occur during trajectory point acquisition. In particular, when the trajectory point stopped or moved at a low speed at the intersection, the incurred error would cause the matched result to show returns on the trajectory along the road network. For example, the inbound section falls behind the intersection, and the outbound section or the intersection falls behind the outbound section after trajectory matching. There is only one correctly matched trajectory sequence: Inbound road section, intersection, outbound road section.

Figure 4 shows an abnormity example of intersection trajectory point matching.

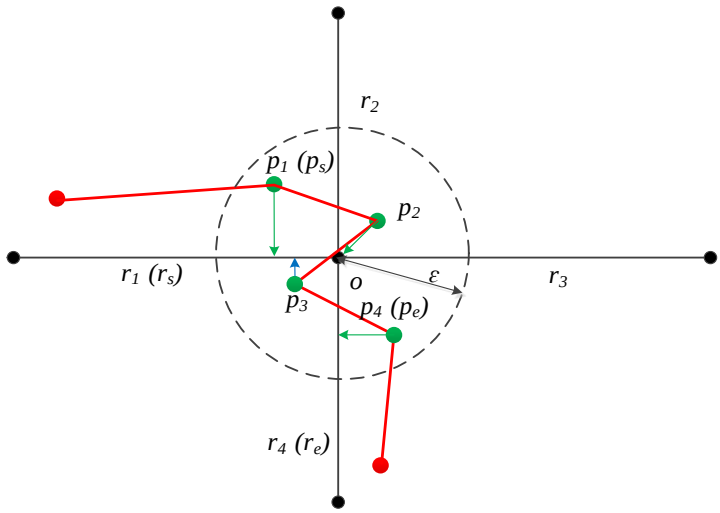


Figure 4. Intersection trajectory point matching abnormity. Points p_1, p_2, p_3, and p_4 are matched to $r_1(r_s)$, o, $r_1(r_s)$, and $r_4(\mathrm{r_e})$, respectively. Thus, the matched trajectory sequence is (r_s, o, r_s, $\mathrm{r_e}$), which suggests that the trajectory is retraced in road section r_s. Since this is an incorrect retrace, it is necessary to transform such disorder into an orderly sequence.

It can be seen from rules II–IV that the trajectory points that match to the inbound and outbound sections can be adjusted to the intersection; such adjustment does not apply to those matched to the intersection. According to rule I, the trajectory points that match to the inbound section can be adjusted to the intersection and the outbound section, and those that match to the outbound section can be adjusted to the intersection and the inbound section.

A more reasonable result can be reached through adjustment, but this is complicated since it is necessary to determine not only the rule where the matched trajectory point is generated but also how to adjust. Therefore, a simplification is made in the proposed method by specifying that the adjustment is made only from the inbound section to the intersection or from the outbound section to the intersection. This is how rule V for abnormity adjustment is made.

- Rule V: The road segments matched by intersection trajectory points strictly follow the "inbound road section–intersection–outbound road section" sequence.

Rule V can be done by the following method. Suppose that the inbound road section is r_s, the outbound road section is r_e, the intersection is o, and the matched road segments are $\{r_i | 1 \le i \le n, \mathrm{r_i} \in \{r_s, r_e, o\}\}$, each element in the set $\{\mathrm{r_i}\}$ should be handled from position 1 to $n - 1$ by the following four situations:

1. IF $r_i = r_e$ AND $r_{i+1} = \mathrm{o}$, THEN $r_i = o$.
2. IF $r_i = r_e$ AND $r_{i+1} = r_s$, THEN $r_i = \mathrm{o}$, $r_{i+1} = o$.
3. IF $r_i = o$ AND $r_{i+1} = r_s$, THEN $r_{i+1} = o$.
4. No adjustment is made in any other situation.

Patent Owner Ex. 2005, p. 7
Tesla v. iQar, IPR2024-00543

For example, there is an intersection trajectory segment containing nine points, expressed as (p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9). Suppose that, according to match rules I–IV, the sequence of the matched road section is (r_s, r_s, o, o, r_e, o, r_s, r_e, r_e). Then, in the sequence, there are two abnormities (r_e, o) and (o, r_s) because o must be before r_e and r_s must be before o. Therefore, according to rule V, (r_e, o) is adjusted to (o, o), and (o, r_s) is adjusted to (o, o). Therefore, the entire adjusted road section sequence is (r_s, r_s, o, o, o, o, o, r_e, r_e).

3.4. Intersection Trajectory Segment Matching Algorithm

The intersection trajectory segment matching is shown in Algorithm 1. It outlines the framework of the intersection trajectory segment matching (InterectTrajMatch) algorithm. Firstly, the algorithm computes the candidate distance sets *dlist* between each intersection trajectory point on P and intersection-related road sections R. Secondly, the algorithm sorts intersection-related road sections R by distance value on *dlist* and then gets the two road sections with the shortest distance. Thirdly, the algorithm finds matched road sections using rules I–IV and adds it to candidate matched road section sets *rmlist*. Finally, after getting the candidate matching road sections of all trajectory points, the algorithm adjusts matched road sections by rule V and returns *RM* as a result.

Algorithm 1 The intersection trajectory segment matching algorithm (InterectTrajMatch)

Input: Intersection trajectory points $P = \{p_i | i = 1, \ldots, n\}$, inbound road section r_s, outbound road section r_e, intersection point o, intersection-related Road Sections $R = \{r_i | i = 1, \ldots, m\}$
Output: The matched road section sequence $RM = \{rm_i | i = 1, \ldots, n\}$

```
Initialize dlist as an empty list; //dlist is a list of candidate distances
Initialize rmlist as an empty list; //rmlist is a list of candidate matched road sections
for i = 1 to n do //Traversing P
    for j = 1 to m do //Traversing R
        d = Distance(p_i, r_j);
        dlist.add(d);
    end for
    R.sort(dlist); //sort R by distance value on dlist
    GetShortestTwoSections(R, ref r_1, ref r_2); //get two road sections with the shortest distance
    rm = MatchbyRule1234(r_1, r_2, r_e, r_s, o); //find matched road section by rule I, II, III, IV
    rmlist.add(rm);
end for
RM = AdjustbyRule5(rmlist); //adjust match road section by rule V.
return RM;
```

3.5. Matching of Inbound Road Section and Outbound Road Section

It is crucial to match the correct inbound and outbound road section. This affects the correct execution of Algorithm 1. As defined in Definitions 8 and 9, the inbound road section is matched before the trajectory enters the intersection, which is usually matched with the inbound point, and the outbound road section is matched after the trajectory moves from the intersection, which is usually matched with the outbound point. However, since the inbound point is very close to its next point, the road matching of the inbound point is often wrong according to the "Look Ahead" method, as shown in Figure 1. Therefore, the inbound road section is defined as the road that matches with the previous point of the inbound point. Obviously, this requires that the matching road of the entry point is the same as the matching road of its previous point. This is achievable at high-frequency sampling, but in the case of medium-frequency or low-frequency sampling and some exceptional cases the distance between the inbound point and its previous point is great, so the matching road of the inbound point and its previous point are not the same road section, as shown in Figure 5.

As shown in Figure 5a, points p_s to p_e are in the ε-neighborhood of the intersection o, the matching road section at the previous point of the inbound point $p_{s\text{-}1}$ is $r_{s'}$, and the adjacent intersection of

Patent Owner Ex. 2005, p. 8
Tesla v. iQar, IPR2024-00543

$r_{s'}$ is o'. Therefore, it is necessary to judge whether the point p_s and its subsequent points are in the ε-neighborhood of o' instead of o. When p_s is not in the ε-neighborhood of o', the inbound point p_s in Figure 5a is not an intersection point. As shown in Figure 5b, the inbound point p_s in Figure 5a is converted to $p_{s\text{-}1}$, and the intersection trajectory point is the new p_s to p_e. Moreover, due to the adjacent relationship between $r_{s'}$ and r_s, the matching road section of $p_{s\text{-}1}$ is still r_s instead of r_i, according to the "Look Ahead" method.

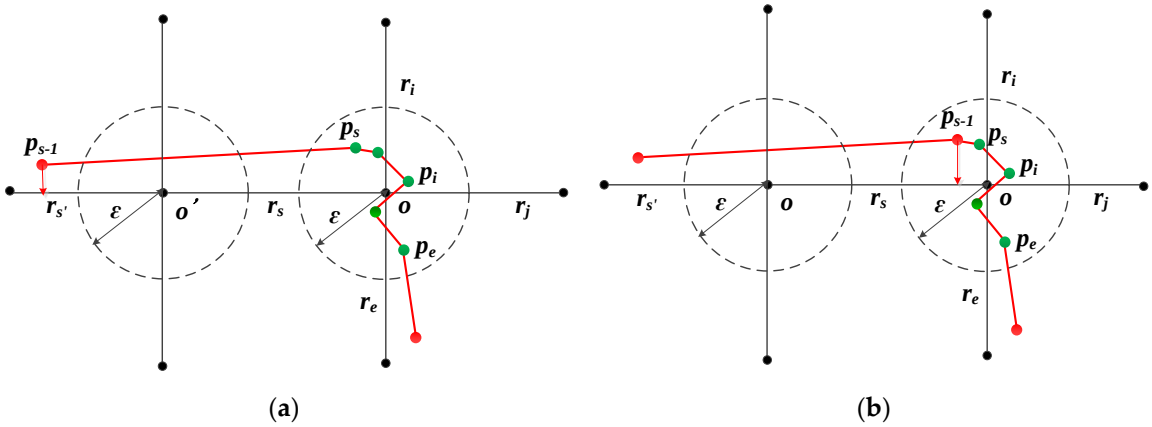


Figure 5. Matching of inbound road section with the previous point of the inbound point. Point p_s is the inbound point, and Point $p_{s\text{-}1}$ is the previous point of the inbound point. (**a**) The relationship between the inbound point and its previous point in the form, and (**b**) the relationship between the inbound point and its previous point in actual processing.

3.6. Error Radius ε

Error radius ε includes trajectory point positioning error and road data error. This error radius is represented by Equation (2):

$$\varepsilon = \varepsilon_l + \varepsilon_r \tag{2}$$

where ε_l is positioning error, which is determined by the positioning technique; ε_r is road data error, which is mainly caused by the difference between the actual road width and the road line data, and its calculation is as shown in Equation (3) [7]:

$$\varepsilon_r = 0.5 \times \frac{w}{sin\frac{\alpha}{2}} \tag{3}$$

where w is the width of road, α is the angle between two intersecting roads. In order to simplify the calculation, the angle is generally considered to be 90 degrees.

Error radius ε influences the efficiency and accuracy of the intersection matching method. Due to errors in positioning data and the road network, the trajectory point within the intersection is excluded if ε is too small, which might result in a mismatch. Otherwise, the trajectory point beyond the intersection will be included, which will lead to lower matching efficiency and a new mismatch (i.e., when the road this trajectory point matched is one of the adjoining roads of the intersection).

4. Segmented Trajectory Matching Method

A segmented trajectory matching strategy is used for the map matching of the whole trajectory. First, the trajectory is divided into the intersection trajectory segment and the non-intersection trajectory segment based on ε. The proposed intersection trajectory segment matching method is applied to the intersection trajectory segment, and the "Look-Ahead" method is applied to the non-intersection trajectory segment. The algorithm of the proposed method is shown in Algorithm 2.

Algorithm 2 outlines the framework of the segmented trajectory matching algorithm. Firstly, it finds matching road sections of the first point of trajectory by the shortest distance method from

Patent Owner Ex. 2005, p. 9
Tesla v. iQar, IPR2024-00543

all roads and adds it to the matched road section sequence *RM*. Secondly, it calculates the distance between each trajectory point and current intersection point *oc*. If the distance is less than the radius ε, it will be added to candidate intersection trajectory points *plist* until the distance of the next point is greater than ε. If the set *plist* is not empty, the algorithm matches each point on *plist* to a road section using Algorithm 1 and adds them to the *RM*; otherwise, it matches the current point by the "Look Ahead" method and adds it to the *RM*. Finally, the algorithm returns *RM* as a result.

In the algorithm, the inbound point (r_s) and the outbound point (r_e) are matched using the "Look Ahead" method, so the correctness of their matching depends on the "Look Ahead" method. Since the method is more suitable for high-frequency data, when the data sampling frequency is low, the matching accuracy will be significantly affected. Therefore, the algorithm is suitable for processing high and medium frequency trajectory data, which means that there is at least one trajectory point on each road. However, due to data errors, the frequency of the trajectory data is not consistent. There are some trajectory points with a large time interval in the high-frequency data. In order to avoid this problem, a time interval threshold is set. If the time interval between the current trajectory point and the previous point does not exceed the threshold, the current point is matched by the "Look Ahead" method. If it exceeds, the current point is treated as the first point of the trajectory and matched by the *MatchFirstPoint* function.

Algorithm 2 Segmented trajectory matching algorithm

Input: Trajectory points $P = \{p_i | i = 1, \ldots, n\}$, intersection points $O = \{o_i | i = 1, \ldots, m\}$, road sections $R = \{r_i | i = 1, \ldots, k\}$, error radius ε
Output: The matched road section sequence *RM*

```
Initialize plist as an empty list; //plist is a list of candidate intersection trajectory point
Initialize rrlist as an empty list; //rrlist is a list of candidate intersection-related road section
Initialize rmlist as an empty list; //rmlist is a list of candidate matched road section.
rf = MatchFirstPoint(p_1, R); //match first point by shortest distance method from all roads.
RM.add(rf);
r_s = rf; //r_s means inbound road section
for i = 2 to n do //Traversing P from p_2
    oc = FindCurrentIntersectionPoint(rf, O); //oc is the intersection close to point p_i
    if (Distance(p_i, oc) ≤ ε)
        plist.add(p_i);
    else
        rrlist = FindRelatedRoadSections (oc, R); //rrlist is oc-related road section sets
        if (plist.count > 0)
            r_e = MatchbyLookAhead(p_{i-1}, rrlist); //match outbound point by Look Ahead method
                                                     [8], r_e is the outbound road section
            rmlist = InterectTrajMatch(plist, r_s, r_e, oc, rrlist); //Agorithm 1
            RM.add(rmlist);
            plist.clear();
        else
            r_e = MatchbyLookAhead(p_i, rrlist);
            RM.add(r_e);
        end if
        r_s = r_e; //current matched road section is next inbound section
    end if
end for
return RM
```

Patent Owner Ex. 2005, p. 10
Tesla v. iQar, IPR2024-00543

5. Experiment and Results

5.1. Experimental Data and Scheme

Experimental data: This includes three trajectory data of taxis with different sampling frequencies during a week within Beijing [31,32] and the road network of Beijing, as shown in Figure 6.

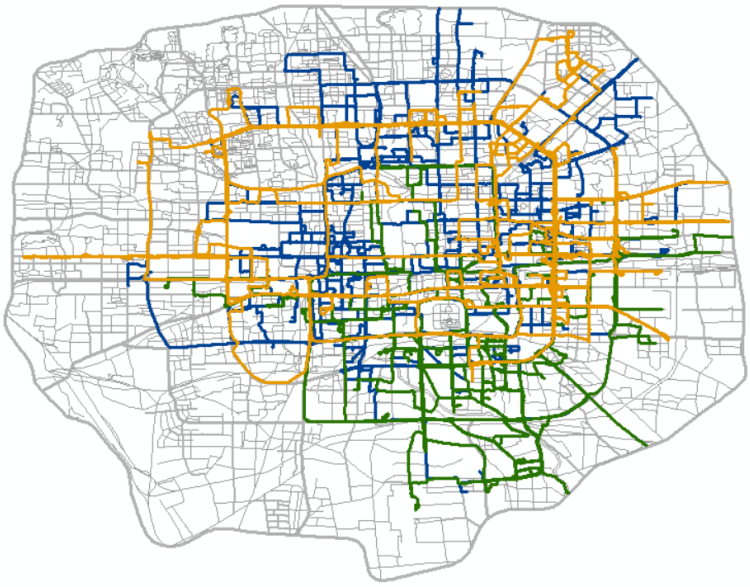


Figure 6. Road network and trajectory data. Gray lines are the road network. Color lines are trajectory data, where the blue lines are data 1, the orange lines are data 2, and the green lines are data 3.

The three different frequency trajectory data are selected from the entire taxi trajectory dataset, which contains the GPS trajectories of 10,357 taxis during the period of 2–8 February 2008 within Beijing. As shown in Table 1, the trajectories are collected at three different sampling intervals: 1 s, 5 s, and 15 s. Precisely, the sampling interval of data 1 is 1 s, and there are 151,542 trajectory points in data 1. The sampling interval of data 2 is 5 s, and the number of trajectory points of data 2 is 30,156. The sampling interval of data 3 is 15 s, and the number of trajectory points of data 3 is 7141.

Table 1. Trajectory data information.

Number	Sampling Interval (s)	Number of Trajectory Points
1	1	151,542
2	5	30,156
3	15	7141

Experiment implementation: To access and visualize the trajectories and map data, ArcGIS 10 plug-in development is carried out using C# on the. NET platform.

Analysis: The analysis process consists of two parts: First, this method is analyzed using different error radii from the efficiency and accuracy; second, a comparative analysis of this method with the LCS method [6] and the decision domain HMM method [7] is conducted from the efficiency and accuracy.

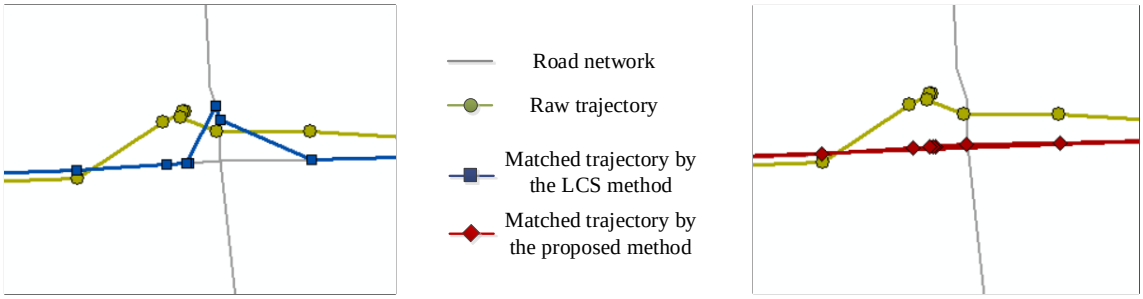
The error radius needs to be determined before analysis. The error radius includes positioning error and road data error. The trajectory data in the experiment uses civil GPS positioning data, and the error is within 20 m [7]. According to China's urban road design standards [33], the width of urban roads ranges from 10 m to 60 m. Since the road network data in the experiment include various levels of road data, its maximum width is 60 m. Therefore, the maximum value of the road data error is $60/2 \times \sqrt{2} \approx 42$ m [7], and the maximum error radius is 62 m. Then, in order to comprehensively

Patent Owner Ex. 2005, p. 11
Tesla v. iQar, IPR2024-00543

analyze the effects of different error radii on the efficiency and accuracy of the method, eleven error radii (10 m, 20 m, 30 m, 40 m, 50 m, 60 m, 70 m, 80 m, 90 m, 100 m, and 110 m) are determined.

5.2. Comparison of Intersection Matching Methods

Figure 7 shows part of the matching result at the intersection, where the gray lines, yellow dotted lines, blue dotted lines, and red dotted lines are the road network, original trajectories, the matching result using the LCS method, and the matching result using this method, respectively. It can be seen from the figure that there is a mismatch at the intersection in the matching result with the LCS method (Figure 7a), whereas the matching result with the proposed method is correct (Figure 7b).

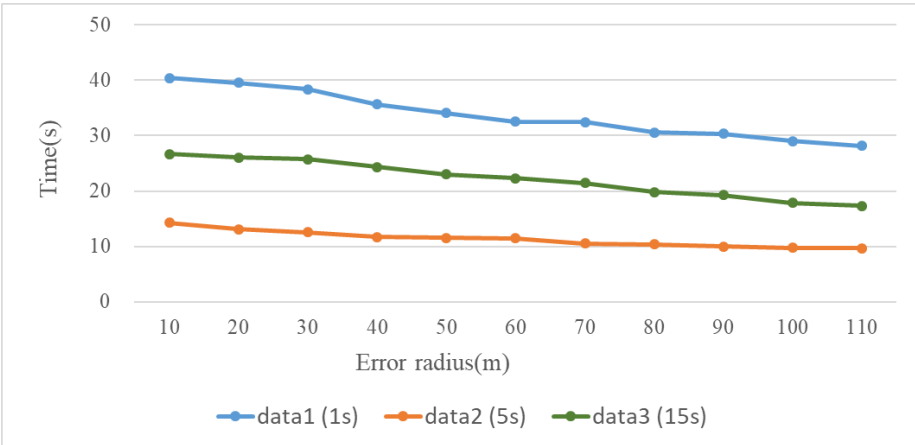


(**a**) (**b**)

Figure 7. Comparison of matching results: (**a**) The LCS method, and (**b**) the proposed method.

5.3. Efficiency Analysis

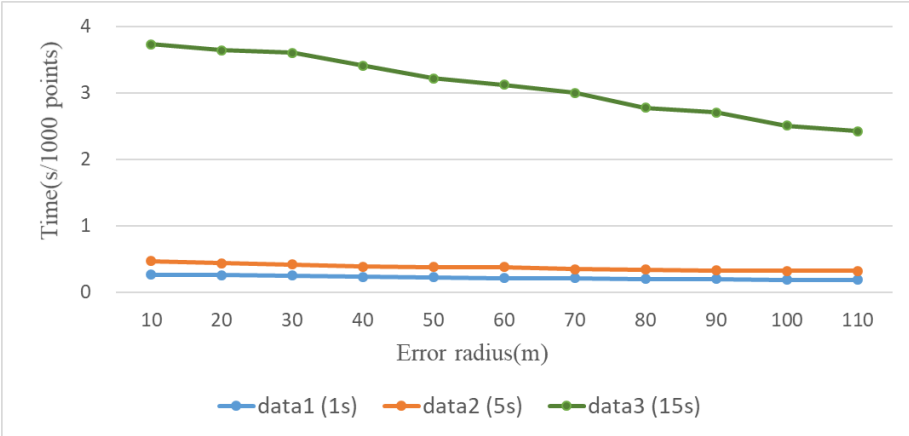
The efficiencies of this method at the eleven error radii are compared. The efficiency analysis results of data 1–3 are shown in Figure 8. Figure 8a shows the total running time of each trajectory, and Figure 8b shows the average running time per 1000 points of the trajectory.



(**a**)

Figure 8. *Cont.*

Patent Owner Ex. 2005, p. 12
Tesla v. iQar, IPR2024-00543



(**b**)

Figure 8. Comparison of the proposed method's efficiency at the eleven error radii: (**a**) The total running time of each trajectory data, and (**b**) the average running time per 1000 points of trajectory data.

According to the experimental results in Figure 8, the following is observed: (1) With the increase of the radius, the algorithm efficiency of this method shows a decreasing trend. However, the rate of descent is low, especially after the error radius exceeds 70 m. (2) From the average time result of Figure 8b, it can be shown that the higher the sampling frequency of the trajectory data, the higher the efficiency of the method. Moreover, the average duration of data 3 is much larger than data 1 and 2, which means that when the sampling interval is greater than 5 s, the efficiency of the method decreases very rapidly with the increase of the sampling interval.

In order to compare the method with the LCS method and the decision domain HMM method, the error radius is set to 60 m, and the similarity score threshold of LCS is 0.95. The results of the analysis are shown in Table 2.

Table 2. Comparison of the efficiency of this method with the LCS method and the decision domain HMM method.

Methods	Running Total Time (s)		
	Data 1	Data 2	Data 3
this method, 60 m	32.5	11.5	22.3
LCS, 95%	348.5	99.5	67.8
decision domain HMM, 60 m	454.6	120.6	96.4

According to the results of the efficiency comparison in Table 2, the following is observed: (1) Compared to the LCS method and decision domain HMM method, the efficiency of the proposed method is higher. (2) The higher the sampling frequency of the trajectory data, the higher the efficiency of this method. For example, when the sampling interval is 15 s, the running time of the LCS method and the decision domain HMM method is about 3 times and 4.5 times this method respectively, and when the sampling interval is 5 s, the running time of the LCS method and the decision domain HMM method is increased to 9 times and 11 times the method, respectively; when the sampling interval is 1 s, the running time of the LCS method and the decision domain HMM method is increased to 11 times and 13 times this method, respectively.

Therefore, the result of efficiency analysis not only indicates that the method is more efficient but also shows that the method is more suitable for high frequencies.

Patent Owner Ex. 2005, p. 13
Tesla v. iQar, IPR2024-00543

5.4. Accuracy Analysis

The accuracy analysis adopts two evaluation standards: One is all trajectory points matching accuracy, and the other is intersection trajectory points matching accuracy.

All trajectory points matching accuracy is represented by Equation (4):

$$c_{all} = \frac{n_{all_m}}{n_{all}} \tag{4}$$

where c_{all} is all trajectory points matching accuracy, n_{all_m} is the number of trajectory points correctly matched, and n_{all} is the total number of trajectory points.

Intersection trajectory points matching accuracy is represented by Equation (5):

$$c_i = \frac{n_{i_m}}{n_i} \tag{5}$$

where c_i is intersection trajectory points matching accuracy, n_{i_m} is the number of intersection trajectory point correctly matched, and n_i is the total number of intersection trajectory point.

Similarly, the accuracy of the method at different error radii is analyzed, and then the accuracy of this method is compared with the LCS method and decision domain HMM method.

Figure 9 presents the accuracy comparison of this method at eleven error radii in data 1–3.

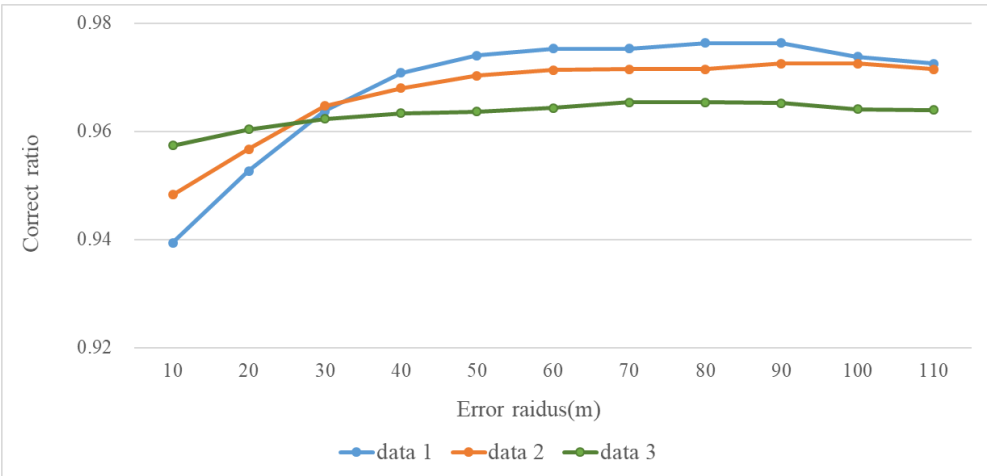


Figure 9. Accuracy comparison of this method at eleven error radii.

Figure 9 shows that: (1) With the increase of the error radius, the accuracy of this method shows an increasing trend. However, the increased speed is not stable. When the error radius is less than 40 m, the increased speed is faster; when the error radius is between 40 m and 90 m, the increased speed is slower; when the error radius is higher than 90 m, the accuracy no longer increases at all, and it even decreases slightly. Thus, a suitable error radius should be between 60 m and 90 m. (2) The proposed method is significantly affected by the trajectory data with different sampling frequencies. When the sampling frequency is high, the matching accuracy varies among different error radii; otherwise, the matching accuracy changes less. This feature is especially noticeable when the error radius is less than 40 m. Therefore, the method is more suitable for high-frequency sampling trajectory data, and the sampling interval does not exceed 15 s. (3) When the error radius continues to increase (e.g., more than 90 m in the experiment), the matching accuracy may decrease. The reason is that the intersection trajectory points have mistakenly included the trajectory points of roads that do not adjoin with the intersection when the excessive error threshold exceeds the minimum length of the road adjoining with the intersection, which led to a new mismatch.

Patent Owner Ex. 2005, p. 14
Tesla v. iQar, IPR2024-00543

The results of comparing the accuracy of this method with the LCS method and the decision domain HMM method are shown in Table 3.

Table 3. Accuracy comparison result.

Methods	All Trajectory Points Matching Accuracy			Intersection Trajectory Points Matching Accuracy		
	Data 1	Data 2	Data 3	Data 1	Data 2	Data 3
this method, 60 m	0.978	0.971	0.964	0.995	0.987	0.980
LCS, 95%	0.952	0.958	0.963	0.857	0.869	0.894
decision domain HMM, 60 m	0.972	0.969	0.965	0.992	0.985	0.977

According to the results of the accuracy comparison in Table 3, we can find that: (1) This method is higher in matching accuracy than two other methods. Specifically, the method is slightly higher than the two other methods in the matching accuracy of all trajectory points. It is slightly higher than the decision domain HMM method, but it is much higher than the LCS method in the matching accuracy of the intersection points. (2) The sampling frequency of the trajectory data has different effects on different methods. As the sampling frequency decreases, the accuracy of the proposed method and the decision domain HMM method also decreases, while the LCS method increases slightly. The results show that this method and the decision domain HMM method are more suitable for high-frequency data, while the LCS method is more suitable for low-frequency data. Besides, compared with the decision domain HMM method, this method has a more considerable difference in matching accuracy among the three experimental data. For example, the accuracy difference between data 1 and 3 of the decision domain HMM method is 0.007, but this method reaches 0.014. Therefore, the proposed method is more sensitive to the sampling frequency of the trajectory data.

6. Conclusions

This study has proposed a segmented matching method by which trajectory matching is divided into intersection matching and non-intersection matching. The proposed method not only addresses mistakes in intersection trajectory matching but also provides a higher matching efficiency and better matching accuracy than the LCS method and decision domain HMM method. However, from the results of the experimental analysis, the proposed method also has its applicable data and application scenarios.

First of all, the method is more suitable for trajectory data of high-frequency sampling. It can be found from the experiment that the higher the sampling frequency of the data, the higher the accuracy of the method. When the frequency is gradually reduced, the accuracy of the method gets gradually closer to the LCS method and decision domain HMM method. The reason is that when the trajectory data sampling frequency is low, there may be fewer or no points at the intersection, so the intersection trajectory point matching method in this research would be useless. Second, because the core of the method is intersection trajectory points matching, the application scenario of the method should be in a road network with multiple road intersections. Therefore, the method is more suitable for processing the moving trajectory in the area with dense roads. Third, in this paper, the error radius ε is analyzed in detail through the combination of theoretical derivation and experimental analysis. With the error radius maximum (62 m) as the center value, 11 error radius values are selected for experimental analysis. Experimental results show that the appropriate error radius values range from 50 m to 90 m. However, it is still deficient in the error radius ε, which remains a dynamic range of values since it is closely associated with trajectory data accuracy, road network data accuracy, and road network data density. Therefore, ε should be set as large as possible, but less than the minimum length of the trajectory matched. In addition, it is difficult for this method to deal with trajectory data where there is a sizeable positional deviation. Therefore, before using this method for map matching, the trajectory data needs to be preprocessed to eliminate the abnormal points.

Patent Owner Ex. 2005, p. 15
Tesla v. iQar, IPR2024-00543

For future work, this method is based on a local matching method to deal with high-frequency trajectory data in urban road networks, and it is difficult to achieve high accuracy when the trajectory data has multiple road network scenes or contains multiple sampling frequencies. Therefore, a map method combining a global matching method and a local matching method can be researched to be applicable to various trajectory data.

Author Contributions: Conceptualization, Yi Long and Minshi Liu; Methodology, Minshi Liu and Ling Zhang; Validation, Minshi Liu and Junlian Ge; Formal Analysis, Minshi Liu, Ling Zhang and Junlian Ge; Writing—Original Draft Preparation, Minshi Liu; Writing—Review & Editing, Minshi Liu, Junlian Ge, Weitao Che, and Ling Zhang; Supervision, Yi Long. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 41601499, 41571382, 41501496; the National Key Research and Development Program of China, grant number 2017YFB0503500; Guangdong Innovative and Entrepreneurial Research Team Program, grant number 2016ZT06D336 and GDAS' Special Project of Science and Technology Development, grant number 2017GDASCX-0802.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Quddus, M.A.; Ochieng, W.Y.; Noland, R.B. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transp. Res. Part C Emerg. Technol.* **2007**, *15*, 312–328. [CrossRef]
2. Lou, Y.; Zhang, C.; Zheng, Y.; Xie, X.; Wang, W.; Huang, Y. Map-matching for low-sampling-rate GPS trajectories. In Proceedings of the 17th ACM Sigspatial International Symposium on Advances in Geographic Information Systems, Seattle, WA, USA, 4–6 November 2009.
3. Bataineh, S.; Bahillo, A.; Díez, L.; Onieva, E.; Bataineh, I. Conditional random field-based offline map matching for indoor environments. *Sensors* **2016**, *16*, 1302. [CrossRef] [PubMed]
4. Hashemi, M.; Karimi, H.A. A critical review of real-time map-matching algorithms: Current issues and future directions. *Comput. Environ. Urban Syst.* **2014**, *48*, 153–165. [CrossRef]
5. Wang, H.; Li, J.; Hou, Z.; Fang, R.; Mei, W.; Huang, J. Research on parallelized real-time map matching algorithm for massive GPS data. *Clust. Comput.* **2017**, *20*, 1123–1134. [CrossRef]
6. Zhu, L.; Holden, J.R.; Gonder, J.D. Trajectory Segmentation Map-Matching Approach for Large-Scale, High-Resolution GPS Data. *Transp. Res. Rec.* **2017**, *2645*, 67–75. [CrossRef]
7. Qi, H.; Di, X.; Li, J. Map-matching algorithm based on the junction decision domain and the hidden Markov model. *PLoS ONE* **2019**, *14*, e0216476. [CrossRef] [PubMed]
8. Brakatsoulas, S.; Pfoser, D.; Salas, R.; Wenk, C. On map-matching vehicle tracking data. In Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, 30 August–2 September 2005; pp. 853–864.
9. Yuan, J.; Zheng, Y.; Zhang, C.; Xie, X.; Sun, G.Z. An Interactive-Voting Based Map Matching Algorithm. In Proceedings of the Eleventh International Conference on Mobile Data Management, Kanas City, MO, USA, 23–26 May 2010.
10. Greenfeld, J.S. Matching GPS Observations to Locations on a Digital Map. In Proceedings of the 81st Annual Meeting of the Transportation Research Board, Washington, DC, USA, 13–17 January 2002; pp. 164–173.
11. White, C.E.; Bernstein, D.; Kornhauser, A.L. Some map matching algorithms for personal navigation assistants. *Transp. Res. Part C Emerg. Technol.* **2000**, *8*, 91–108. [CrossRef]
12. Hsueh, Y.L.; Chen, H.C. Map matching for low-sampling-rate GPS trajectories by exploring real-time moving directions. *Inf. Sci.* **2018**, *433*, 55–69. [CrossRef]
13. Quddus, M.A.; Ochieng, W.Y.; Zhao, L.; Noland, R.B. A general map matching algorithm for transport telematics applications. *GPS Solut.* **2003**, *7*, 157–167. [CrossRef]
14. Hashemi, M.; Karimi, H.A. A weight-based map-matching algorithm for vehicle navigation in complex urban networks. *J. Intell. Transp. Syst.* **2016**, *20*, 573–590. [CrossRef]
15. Sharath, M.N.; Velaga, N.R.; Quddus, M.A. A dynamic two-dimensional (D2D) weight-based map-matching algorithm. *Transp. Res. Part C Emerg. Technol.* **2019**, *98*, 409–432. [CrossRef]
16. Hu, G.; Shao, J.; Liu, F.; Wang, Y.; Shen, H.T. If-matching: Towards accurate map-matching with information fusion. *IEEE Trans. Knowl. Data Eng.* **2016**, *29*, 114–127. [CrossRef]

Patent Owner Ex. 2005, p. 16
Tesla v. iQar, IPR2024-00543

17. Zhao, X.; Cheng, X.; Zhou, J.; Xu, Z.; Dey, N.; Ashour, A.S.; Satapathy, S.C. Advanced topological map matching algorithm based on D–S theory. *Arabian J. Sci. Eng.* **2018**, *43*, 3863–3874. [CrossRef]
18. Alt, H.; Efrat, A.; Rote, G.; Wenk, C. Matching planar maps. *J. Algorithms* **2003**, *49*, 262–283. [CrossRef]
19. Yin, H.; Wolfson, O. A Weight-based Map Matching Method in Moving Objects Databases. In Proceedings of the 16th International Conference on Scientific and Statistical Database Management, Santorini Island, Greece, 21–23 June 2004; pp. 437–438.
20. Wei, H.; Wang, Y.; Forman, G.; Zhu, Y. *Map Matching by Fréchet Distance and Global Weight Optimization*; Technical Paper; Department of Computer Science and Engineering, 2013; p. 19. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.716.7193&rep=rep1&type=pdf (accessed on 10 December 2019).
21. Millard-Ball, A.; Hampshire, R.C.; Weinberger, R.R. Map-matching poor-quality GPS data in urban environments: The pgMapMatch package. *Transp. Plan. Technol.* **2019**, *42*, 539–553. [CrossRef]
22. Knapen, L.; Bellemans, T.; Janssens, D.; Wets, G. Likelihood-based offline map matching of GPS recordings using global trace information. *Transp. Res. Part C Emerg. Technol.* **2018**, *93*, 13–35. [CrossRef]
23. Rappos, E.; Robert, S.; Cudré-Mauroux, P. A force-directed approach for offline GPS trajectory map matching. In Proceedings of the 26th ACM Sigspatial International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 6–9 November 2018; pp. 319–328.
24. Newson, P.; Krumm, J. Hidden Markov map matching through noise and sparseness. In Proceedings of the ACM Sigspatial International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 4–6 November 2009.
25. Koller, H.; Widhalm, P.; Dragaschnig, M.; Graser, A. Fast hidden Markov model map-matching for sparse and noisy trajectories. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, Spain, 15–18 September 2015; pp. 2557–2561.
26. Yang, C.; Gidofalvi, G. Fast map matching, an algorithm integrating hidden Markov model with precomputation. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 547–570. [CrossRef]
27. Szwed, P.; Pekala, K. An Incremental Map-Matching Algorithm Based on Hidden Markov Model. In Proceedings of the International Conference on Artificial Intelligence and Soft Computing, ICAISC 2014, Zakopane, Poland, 1–5 June 2014; pp. 579–590.
28. Mohamed, R.; Aly, H.; Youssef, M. Accurate real-time map matching for challenging environments. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 847–857. [CrossRef]
29. Jagadeesh, G.R.; Srikanthan, T. Online map-matching of noisy and sparse location data with hidden Markov and route choice models. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2423–2434. [CrossRef]
30. Algizawy, E.; Ogawa, T.; El-Mahdy, A. Real-time large-scale map matching using mobile phone data. *ACM Trans. Knowl. Discov. Data* **2017**, *11*, 52. [CrossRef]
31. Yuan, J.; Zheng, Y.; Xie, X.; Sun, G. Driving with knowledge from the physical world. In Proceedings of the 17th ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 316–324.
32. Yuan, J.; Zheng, Y.; Zhang, C.; Xie, W.; Xie, X.; Sun, G.; Huang, Y. T-drive: Driving directions based on taxi trajectories. In Proceedings of the 18th Sigspatial International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 99–108.
33. Ministry of Housing and Urban-Rural Development of the People's Republic of China. *Code for Design of Urban Road Engineering (CJJ37-2012)*; China Building Industry Press: Beijing, China, 2012.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).