



Article

# An Automatic Method for Detection and Update of Additive Changes in Road Network with GPS Trajectory Data

Jianbo Tang <sup>1</sup> , Min Deng <sup>2</sup>, Jincai Huang <sup>3,\*</sup> , Huimin Liu <sup>2</sup> and Xueying Chen <sup>2</sup>

<sup>1</sup> State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; tangjb@whu.edu.cn

<sup>2</sup> Department of Geo-informatics, Central South University, Changsha, Hunan 410083, China; dengmin@csu.edu.cn (M.D.); lhmgis@csu.edu.cn (H.L.); htcxycxc@csu.edu.cn (X.C.)

<sup>3</sup> Shenzhen Key Laboratory of Spatial Smart Sensing and Service, Shenzhen University, Shenzhen 518060, China

\* Correspondence: huangjincaicsu@csu.edu.cn; Tel.: +86-153-0849-4530

Received: 11 July 2019; Accepted: 4 September 2019; Published: 13 September 2019



**Abstract:** Ubiquitous trajectory data provides new opportunities for production and update of the road network. A number of methods have been proposed for road network construction and update based on trajectory data. However, existing methods were mainly focused on reconstruction of the existing road network, and the update of newly added roads was not given much attention. Besides, most of existing methods were designed for high sampling rate trajectory data, while the commonly available GPS trajectory data are usually low-quality data with noise, low sampling rates, and uneven spatial distributions. In this paper, we present an automatic method for detection and update of newly added roads based on the common low-quality trajectory data. First, additive changes (i.e., newly added roads) are detected using a point-to-segment matching algorithm. Then, the geometric structures of new roads are constructed based on a newly developed decomposition-combination map generation algorithm. Finally, the detected new roads are refined and combined with the original road network. Seven trajectory data were used to test the proposed method. Experiments show that the proposed method can successfully detect the additive changes and generate a road network which updates efficiently.

**Keywords:** map construction; road update; GPS trajectory; decomposition-combination

## 1. Introduction

The up-to-date road network is important for intelligent transportation applications such as car navigation and route planning. With the development of urban transportation, more and more roads are being built, especially in the newly developed urban regions. Traditional methods for the updating of road networks, such as field surveying and satellite image digitizing, are time-consuming and costly [1]. In the age of autonomous vehicles, many commercial companies such as Google, TomTom, HERE, and DeepMap employ employees to update road maps using high instrumented vehicles driving around. That uses significant human and financial resources to update the road maps over time. Public vehicles such as taxis or buses are natural sensors of road network information [2]. With more and more public vehicles are being equipped with GPS devices, a large amount of GPS trajectory data are now available. This new data resource contains abundant road network information and it provides new opportunities for real-time production and updating of the road network. Compared with remote sensing images and high-quality GPS data collected by professional survey vehicles, GPS trajectory data collected by public vehicles are relatively cheap and can be acquired in real-time with

comprehensive coverage. These advantages have attracted many scholars to use this new data resource to extract and update road network information, such as road centerlines, average travel time, speed limit, number of lanes, and turning rules [3–12]. However, the commonly available GPS trajectory data are usually low-quality data with noise, low sampling rates, and uneven spatial distributions. This presents big challenges to existing road network generation and updating methods [13,14]. In this paper, we focus on the automatic generation and update of a road network based on low-quality GPS trajectory data. The proposed method takes the low-quality GPS trajectories and the original road network as the raw input data and the output is a road network with new additive road updates.

The rest of the paper is organized as follows. Section 2 presents the related studies for road network generation and updating. In Section 3, the framework of this study and the details of the proposed method are described. Section 4 presents the experimental results and comparison analysis. Finally, the conclusion and future work are given in Section 5.

## 2. Related Work

Existing methods for the update of the road network using GPS trajectory data can be classified into two categories: (1) global reconstruction methods, and (2) local additive update methods.

In global reconstruction methods, the whole road network is reconstructed based on raw GPS trajectory data and then the road changes are detected and updated by comparing the reconstructed road network with the original existing road maps [15]. Global reconstruction methods can be further classified into four subclasses [16]; i.e., density image skeleton extraction, point clustering, incremental track insertion, and intersection linking. Density image skeleton extraction methods first convert the raw GPS trajectories or points into a density image and then use the skeleton extraction algorithms, such as Voronoi partitioning [4], mathematical morphology [7,17,18], and Morse theory [19], to extract the road centerlines from the density image. These methods are efficient, but some important local information of the road network may be lost during the transformation from GPS trajectory data into raster images. Point clustering methods use clustering algorithms (such as k-means) to partition the GPS sample points into different clusters, and the cluster centers are then connected to form a complete road network [3,15,20,21]. The cluster number is a critical parameter for these methods and it is usually hard to set in practice because of the noise and uneven spatial distribution of GPS trajectory data. Incremental track insertion methods assume that the road network is a graph and incrementally merge new GPS trajectories to construct a road network. Representative algorithms include the work of Cao and Krumm [5], Ahmed and Wenk [6], Tang et al. [2], and Zhang et al. [1]. These methods can reconstruct the complex structure of a road network; however, they may generate incorrect paths due to noise in GPS trajectories. Intersection linking methods first identify the intersections of the road network and then extract the road segments that link these intersections to form a complete road network [8,22]. Recently, Mariescu-Istodor and Fränti proposed a representative intersection linking algorithm called CellNet for constructing road networks from raw GPS trajectories [23]. The road network generated by these methods is consistent with the real road network in topology, but the identification of road intersections is still a challenging problem that needs to be solved further [12]. Most of the global road network reconstruction methods are designed based on the high sampling GPS trajectory data, but in practice, the common available trajectory data contains a lot of noise due to positioning errors and has a relatively low time resolution [2]. In addition, the reconstruction of a number of already existing road segments in the road network is not necessary and it reduces the efficiency of these methods for the update of the road network in a large region [24].

Local additive update methods focus on the local update of an original road network. These methods mainly detect additive changes in the original road network and reconstruct only the changed parts. Zhao et al. [25] created buffers of the original road network and recognized GPS sampling points that were not in the buffers as the additive changes of the original road network. Then, they transformed the unmatched sampling points into a binary image and extracted the skeleton of the image as the new roads. Fang et al. [26] presented a similar approach to update the road network. Tang et al. [27]

detected changes in the original road network by employing traffic direction, topology, and geometry relationship constraint rules, then they generated centerlines of new roads using a polynomial curve fitting algorithm. As far as we know, currently two map updating systems, CrowdAtlas and COBWEB, have been built by Wang et al. [28] and Shan et al. [29], respectively. CrowdAtlas matches the raw GPS trajectories onto the original road network using the hidden Markov model (HMM) based map matching algorithm, and then the unmatched trajectories are clustered together to extract the centerlines of new roads. CrowdAtlas can perform well on high-quality GPS data; however, it performs poorly on low-quality GPS data because the Hausdorff distance used for clustering the trajectories is sensitive to noise. COBWEB takes unmatched trajectories as the input data and generates new road segments through complex processing steps including ‘graph generalization,’ ‘merging,’ and ‘refinement’ [29]. Recently, Wu et al. [24] presented a local renewal method for road network generation and updating. They used GPS sampling points as the input data and adopted a modified HMM-based map matching algorithm to find the unmatched sampling points that were further clustered via PAM (Partition around medoids) clustering algorithm. Cluster centers were finally connected to generate the centerlines of new roads.

In this paper, an automatic method for generation and update of the road network based on low-quality GPS trajectory data is proposed. Our method is most related to Wu et al.’s method. Both methods first find additive changes (i.e., newly added roads) of the original road network using map matching algorithms and then reconstruct the changed parts to generate a road network with updates. However, there are three differences between these two methods: (1) Our method uses an efficient partial map matching algorithm based on flexible spatial and direction constraint rules, which has lower complexity than the HMM-based map matching algorithm used in Wu et al.’s method. (2) Wu et al.’s method’s assumptions are that all input trajectories inside a changed area follow a similar or unique path, which is not usually true in practice, whereas our method adopts a decomposition-combination strategy that has no restrictions or assumptions on the road network changes. (3) When clustering the unmatched sampling points to generate the geometry of new roads, we use a newly developed adaptive graph-based clustering algorithm, while Wu et al.’s method uses the PAM clustering algorithm that needs to set the cluster number. Details of the proposed method will be described in the following section.

### 3. Methodology

In this paper, we mainly focus on detection and update of the additive changes of the original road network and the removal, geometry, and topology changes of the road network are not discussed in this study. The proposed method consists of three steps: detection of additive changes, construction of new roads, and refinement of road topology. The workflow of the proposed method is shown in Figure 1.

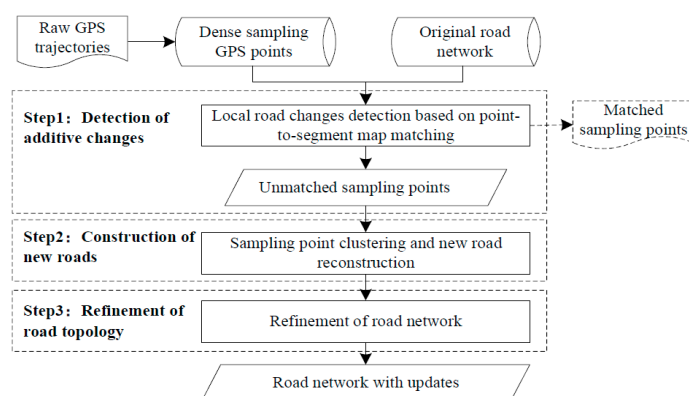


Figure 1. The workflow of the proposed method.

### 3.1. Data Preprocessing

In this study, the input original road network,  $G$ , is a directed graph and it is represented by straight road segments,  $\{e_1, e_2, \dots, e_m\}$ , as in many typical map sources, such as the OpenStreetMap (OSM) project. Each road segment is created with a start node and an end node, and the direction of the road segment is defined as the heading direction from its start node to its end node. The raw input GPS trajectories are a sequence of sampling records where each record contains the vehicle ID, latitude, longitude, timestamp, vehicle heading, and vehicle speed. For GPS trajectory data that only includes locations and timestamps; the vehicle heading of a sampling point can be estimated by the direction from the sampling point to its next sampling point, and the vehicle speed can be estimated by the ratio of the distance to the time interval between two continuous sampling points.

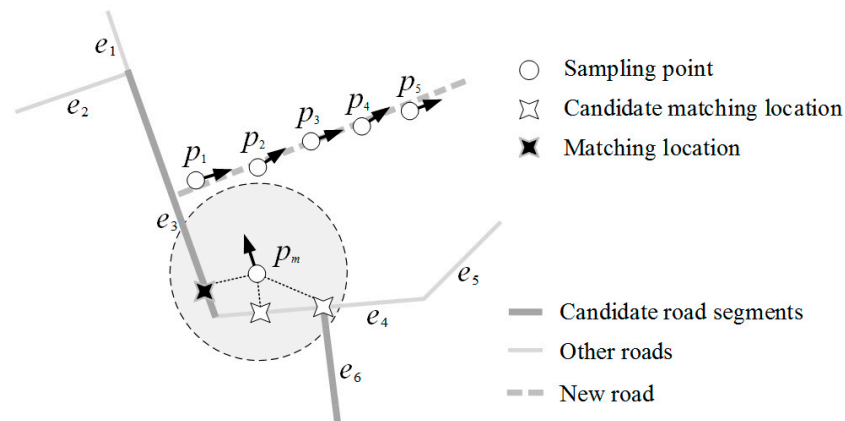
Since human usually drive a vehicle with speed of between 5 km/h to 100 km/h in the urban city, the stationary or error GPS sampling records are first removed according to the vehicle speed  $V < 5$  km/h or  $V > 100$  km/h. Furthermore, due to the loss of GPS satellite signals or the low sampling rate, there may exist some long distance gaps between two continue sampling records in a GPS trajectory. To obtain a set of relatively uniform sampling points, we transform the raw GPS trajectories (polylines) into a set of dense sampling points  $\{p_1, p_2, \dots, p_n\}$  by employing a piecewise-linear interpolation. Each sampling point contains the latitude, longitude, and heading direction. In this paper, the sampling distance for the interpolation is set to 50 meters. These dense sampling points are used as the input data for the proposed method.

### 3.2. Detection of Additive Changes

Since vehicles move on roads, vehicle trajectories are usually restricted to the road network. Therefore, vehicle trajectories that are not matched with the original road network can be used to find additive new roads in the road network. Various map matching algorithms are available for matching the trajectory data and the road network [6,30–32]. Among them, the matching method based on road buffers and the HMM-based map matching algorithm are commonly used in road network changes' detection [24–28]. The matching algorithm based on road buffers is efficient and straightforward, in which GPS trajectories or sampling points are identified as unmatched if are they located outside the buffer zone of the original road network. The matching result of this algorithm may be wrong because only the proximity relation between the trajectory and the road network is considered. The HMM-based map matching algorithm aims to find the most likely path of a GPS trajectory from one candidate position to the next, based on multiple constraints (e.g., spatial proximity, direction, and speed), and it can obtain a global optimal matching result; however, it shows low efficiency on massive GPS trajectories. In this paper, we choose a point-to-segment map matching algorithm in Brakatsoulas et al. [30] to match the input sampling points and the original road network by considering both the efficiency and the accuracy of the matching algorithms. Given a sampling point and the original road network (represented by straight road segments), the road segment matching the sampling point is identified according to the following steps:

- (a) First, the spatial distances from the sampling point to all the road segments are computed, and road segments are filtered if the distances between the road segments and the sampling point are larger than a distance threshold; e.g., 50 meters.
- (b) Second, for the remaining road segments, a road segment is recognized as a candidate road segment if the angle between the heading direction of the sampling point and the direction of the road segment is smaller than an angle threshold; e.g., 15 degrees.
- (c) In all the candidate road segments, the road segment with the smallest spatial distance to the sampling point is finally recognized as the road segment matching the sampling point. If there is not a road segment matching the sampling point, the sampling point is recognized as an unmatched sampling point.

As shown in Figure 2,  $e_1, e_2, e_3, e_4, e_5, e_6$  are road segments in the original road network, and  $\{p_1, p_2, p_3, p_4, p_5\}$  and  $p_m$  are the input sampling points from raw GPS trajectory data, in which  $\{p_1, p_2, p_3, p_4, p_5\}$  are sampling points on a new road. According to the point-to-segment matching algorithm,  $e_3$  is the matching road segment of  $p_m$ , and  $\{p_1, p_2, p_3, p_4, p_5\}$  have no matching road segments, thus they are identified as unmatched sampling points.



**Figure 2.** The point-to-segment matching (arrows indicate the heading directions of the sampling points).

The point-to-segment matching algorithm has two parameters: the distance threshold and the angle threshold. In this study, the distance threshold is used to filter the road segments that may not match the sampling points and it is set to 50 m according to the maximum value of road widths and GPS positioning errors. The angle threshold is used to determine whether two directions are approximately parallel, and the angle threshold is set to 15 degrees according to previous studies [33,34].

### 3.3. Construction of New Roads: A Decomposition-Combination Map Generation Algorithm

Based on the point-to-segment matching algorithm, the unmatched sampling points are detected. These unmatched sampling points indicate the newly added roads of the original road network. The next step is to construct the geometry of the newly added roads.

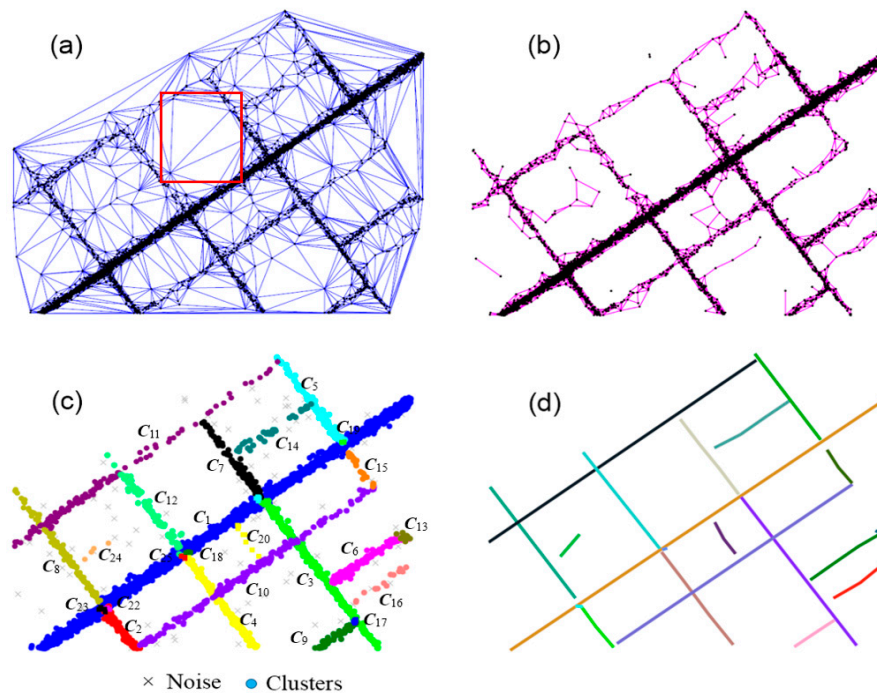
As in many typical map sources, such as the OSM, the road network is represented by straight road segments, and the arbitrarily complex curved roads can also be decomposed into a combination of some straight road segments. Based on this thought, we adopt a decomposition-combination strategy to reconstruct the geometry of new roads. First, the unmatched sampling points on a new road are partitioned into different clusters of a linear shape (corresponding to the straight road segments). Then, the centerlines of new road segments are extracted from these point clusters by employing piecewise linear curve fitting. Finally, the generated new road segments are merged together to form a relatively complete road network. Based on the decomposition-combination strategy, we could generate a road network with an arbitrary, complex shape.

#### 3.3.1. Decomposition of Complex New Roads

Various point clustering algorithms, such as k-means, PAM, and DBSCAN [35], are available for clustering the unmatched sampling points. However, most of these methods are sensitive to clustering parameters, such as the cluster number and the neighborhood size. Moreover, the k-means and PAM algorithms are only suitable for detecting spherical clusters, and the density-based clustering algorithm—DBSCAN (Density Based Spatial Clustering of Applications with Noise)—can find arbitrary shaped clusters, but DBSCAN may lose some important low-density clusters due to the uneven spatial distribution of the sampling points (e.g., the cluster  $C_{11}, C_{14}$ , and  $C_{16}$  shown in Figure 3c. In this paper,



a newly developed graph-based clustering algorithm is used to cluster the unmatched sampling points into different linear clusters, and the overview of the clustering algorithm is shown in Figure 3.



**Figure 3.** An illustration of the graph-based clustering algorithm: (a) Delaunay triangulation (DT) graph of the sampling points; (b) pruned DT graph; (c) detected linear clusters; (d) road centerlines.

In the algorithm, Delaunay triangulation (DT) is used to construct the spatial neighborhood graph of the sampling points, because the DT graph adapts to the uneven density distribution of spatial points and it requires no parameters. Figure 3a shows the DT graph of some unmatched sampling points. However, some inconsistent long edges (linking two distant points) may exist in the DT graph; e.g., edges in the red box in Figure 3a. To prune the inconsistent long edges, a statistical method adapted from [36] is implemented:

Let  $GT$  be the DT graph of the unmatched sampling points (denoted as  $\{p_1, p_2, p_3, \dots, p_K\}$ ),  $global\_mean(GT)$  be the mean value of the length of all edges in  $GT$ , and  $global\_std(GT)$  be the standard deviation of the length of all edges in  $GT$ . Let  $local\_mean(p_i)$  be the mean value of the length of edges linking to the vertex  $p_i$  directly; then, the cut-off value for long edges linking to  $p_i$  is calculated as:

$$cut\_off\_value(p)_i = global\_mean(GT) + \frac{global\_mean(GT)}{local\_mean(p_i)} * global\_std(GT) \quad (1)$$

The cut-off value is adaptive to the uneven density distribution of sample points by using the ratio of  $global\_mean(GT)$  to  $local\_mean(p_i)$  as a factor of the standard deviation,  $global\_std(GT)$ . For a sampling point  $p_i$  (or a vertex) of  $GT$ ,  $cut\_off\_value(p_i)$  is computed, and the edge linking to  $p_i$  is pruned if its length is longer than  $cut\_off\_value(p_i)$ . Figure 3b illustrates the pruned DT graph. The spatial neighbors of a sampling point  $p_i$  are defined as a set of sampling points linking to point  $p_i$  directly by an edge of the pruned DT graph.

Through the above steps, the neighborhood graph of the sampling points is obtained. As the vehicle usually moves on the road with a vehicle heading consistent with the direction of the road. Thus, the heading directions of the sampling points on the same road segment should be similar. Based on this thought, the sample points are clustered according to the following steps:

**Step 1:** According to the number of spatial neighbors, the sampling points are sorted and denoted as  $L = \{p(1), p(2), p(3), \dots, p(K)\}$ , in which  $N(p(1)) \geq N(p(2)) \geq N(p(K))$ , where  $n$  is the number of sampling points and  $N(\bullet)$  is the number of spatial neighbors of a sampling point.

**Step 2:** The sampling point  $p(1)$  is first chosen and initialized as a new point cluster, denoted as  $C = \{p(1)\}$ . Then, the neighboring sampling points of  $p(1)$  are gradually added into the cluster  $C$  if the neighboring sampling point (say  $p(k)$ ) satisfies the two criteria: (1)  $p(k)$  connects to a sampling point in  $C$  by a path of three or fewer edges of the pruned DT graph, and (2) the angle between the heading direction of  $p(k)$  and the direction of cluster  $C$  is less than a threshold  $\Lambda$  ( $\Lambda = 15^\circ$ ), where the direction of cluster  $C$  is defined as the median value of the heading directions of all the sampling points in  $C$ .

**Step 3:** The point cluster  $C$  grows until no neighboring points can be combined. For the remaining GPS points that have not been included in any clusters, the point with the maximum number of spatial neighbors is selected and initialized as a new point cluster. Repeat the above step to merge sampling points into a bigger cluster until all sampling points are visited. Finally, small clusters of five or fewer points are deleted and the remaining clusters are the output results.

There is one parameter that needs to be set for the graph-based clustering algorithm. The threshold  $\Lambda$  ensures the consistency of the heading directions of sampling points in a cluster. Meanwhile, this parameter controls the generation of the linear clusters. The smaller the value of  $\Lambda$  is, the straighter the shape of the point cluster is, but it may also generate more fragmental clusters. In this paper,  $\Lambda$  is set to 15 degrees.

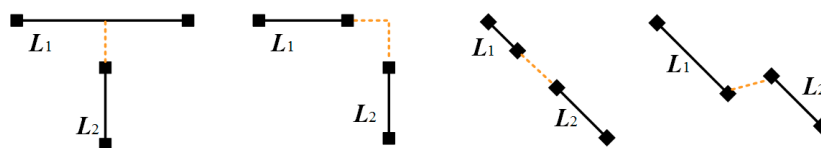
### 3.3.2. Road Centerline Extraction

The next step is to generate road centerlines from the above-detected clusters. Since the shape of the cluster detected by the proposed clustering algorithm tends to be linear, a piecewise linear fitting algorithm is applied to generate the road centerlines. In this paper, the piecewise linear curve fitting algorithm rather than a straight segment fitting algorithm is chosen, because of the reasons that the shape of the cluster found by the clustering algorithm may not be strictly straight and the piecewise linear curve fitting algorithm is more powerful than the simple straight segment fitting. In this paper, we adopt a piecewise linear curve fitting algorithm in Pittman and Murthy [37] to extract the centerlines of road segments based on the sampling point clusters.

The piecewise linear curve fitting algorithm identifies a continuous piecewise linear function  $f(x)$  to fit input data such that fitting error (least-squares error) is minimized. Pittman and Murthy [37] used genetic algorithms to find line segment ends and to identify the optimal piecewise linear curve (i.e., the centerline) of the two-dimensional input point data. The implementation of the algorithm can be found at [https://github.com/cjekel/piecewise\\_linear\\_fit\\_py](https://github.com/cjekel/piecewise_linear_fit_py). Figure 3d shows the centerlines of new roads generated using the piecewise linear fitting algorithm.

### 3.4. Topology Refinement of the Road Network

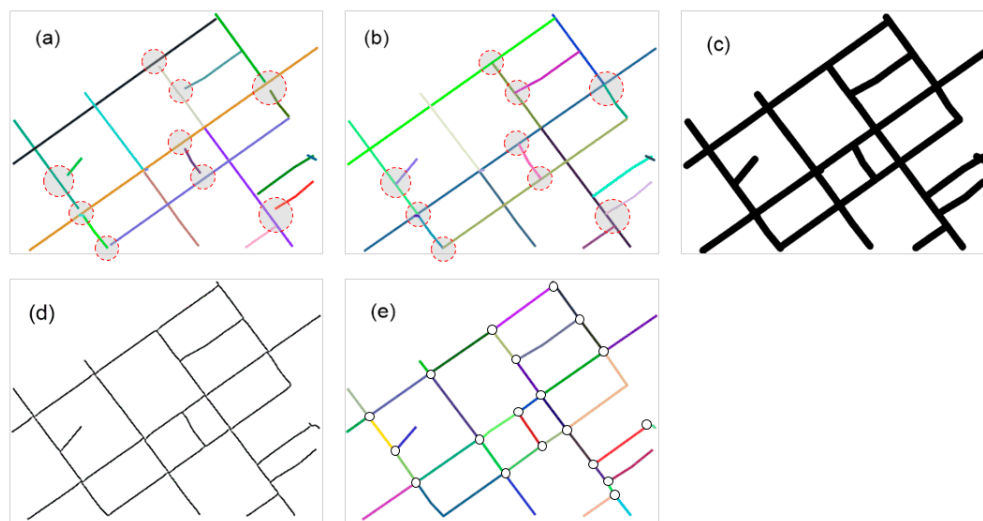
The identified new road segments (or curves) should be further combined and merged into the original road network to form a complete road network with updates. To generate a routable road network, the gaps between the new road segments (or curves) and the topology errors between additive new roads and the original road network should be refined, as shown in Figure 4.



**Figure 4.** The illustration of some topology errors between two new road segments or curves.

Some rule-based methods are commonly used to refine the topology between road segments to form a complete road network. For example, Wu et al. [25] defined two operations, ‘renewal of

intersections' and 'connection,' to extend road segments to ensure their topology and connectivity. Kuntzsch et al. [22] used a reversible jump MCMC (Markov Chain Monte Carlo) sampler to refine the geometry of road junctions. These methods are based on predefined rules and perform well for certain types of topology errors; however, there are still many other topology errors that cannot be modeled in advance. Biagioni and Eriksson [7] have developed a topology refinement method based on raw GPS trajectories, in which operations, such as spurious edge pruning and map matching, are applied repeatedly to determine whether the transitions between two road segments are allowable. This method can perform well, but for massive trajectories, it becomes time-consuming. In this paper, for simplicity and efficiency, we use a mathematical morphology based method to refine the topology of the road network, as shown in Figure 5.



**Figure 5.** Topology refinement: (a) original road segments; (b) roads after the extension operation; (c) buffers of the road segments; (d) skeleton lines extracted from the road buffers; (e) the refined road network.

The mathematical morphology based topology refinement method mainly consists of four steps, as follows:

- (a) First, each road segment (or road curve) is virtually extended outward at both the beginning and ending points along the direction of the road (specially, along the direction of the line segments where the beginning and ending points are on). We then check if the extended road segments (or curves) are intersected with each. If a road segment intersects with others, then this road segment is extended to its nearest intersecting road segment, as the roads in red circles shown in Figure 5b. The extension distance is set to 30 m in this paper, according to GPS positioning errors.
- (b) Then, the buffer zones of the above road segments are created and converted into a binary image as shown in Figure 5c. It can be seen that the gaps between the road segments that should be connected are filled. The buffer size is set to 20 m according to the road width.
- (c) Third, the centerlines of the refined road network are extracted from the binary image (in Figure 5c) based on the mathematical thinning operations in Lam et al. [38]; see Figure 5d.
- (d) Finally, the above-generated road skeleton centerlines are divided into different road segments at the junction points (see Figure 5e). The junction points are detected by checking whether this point connects three or more road segments. The Douglas–Peucker algorithm [39] is further applied to remove the unnecessary intermediate points in the road segments, and the distance parameter for the Douglas–Peucker algorithm is set to 3 m in our experiments. The allowable transitions between two road segments are determined according to whether there exists at least one trajectory matching both these two road segments. Figure 5e shows the refined road network.



#### 4. Experimental Results

In this paper, we designed two experiments to evaluate the effectiveness of the proposed method. The first experiment (experiment I) was designed to test the power the proposed method for construction of new complex roads, and seven state-of-the-art global reconstruction methods were compared. The second experiment (experiment II) was designed to evaluate the effectiveness of the proposed method for the detection and local update of the additive changes of the original road network, and two related local update methods were also compared.

##### 4.1. Experiment I: Performance on the Construction of New Roads

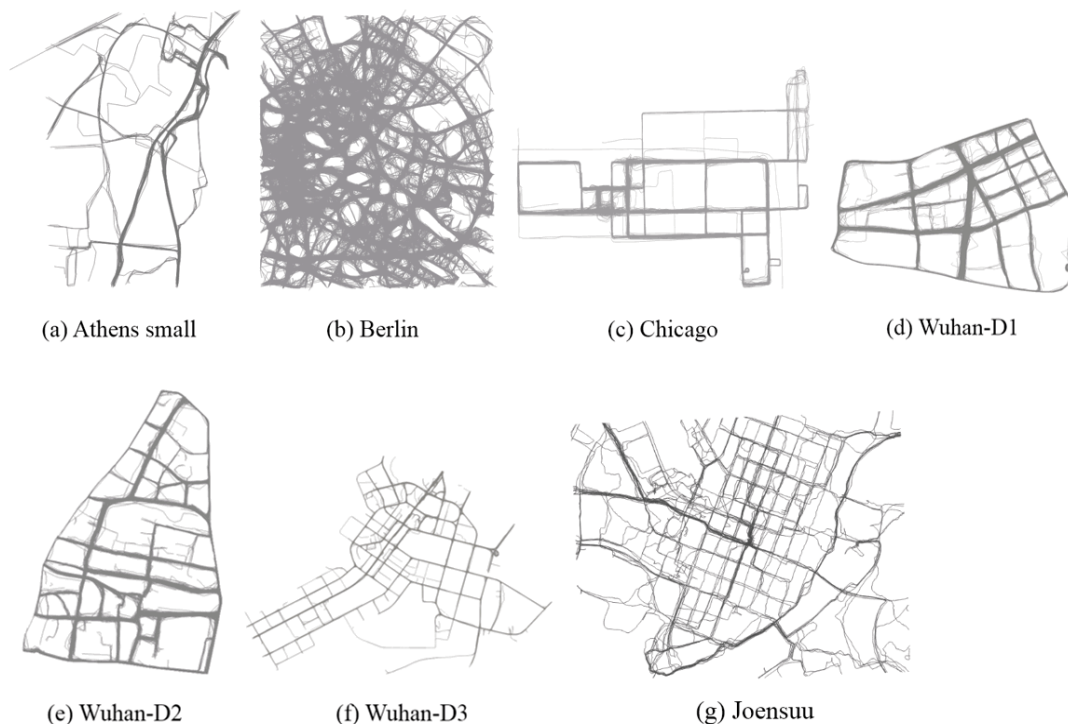
In the first experiment, seven real-world GPS trajectory data sets from different cities were used (see Figure 6). These GPS trajectory data were used as unmatched GPS sampling points for the construct of new roads. The experimental data sets include four public GPS trajectory datasets from <http://mapconstruction.org> and <http://cs.uef.fi/mopsi/routes/network>, and three low-quality taxi GPS trajectory datasets from Wuhan, China. The statistics for the seven trajectory data sets are shown in Table 1. To evaluate the efficiency of the proposed method, seven state-of-the-art global reconstruction algorithms (in Table 2) were also compared (implementations of the first six algorithms are available from <http://mapconstruction.org>). To quantitatively evaluate the performance of the proposed method, recall, precision, and F-score measures adapted from Biagioni and Erikson [7] are used:

$$\text{recall} = \text{length}(\text{TP}) / \text{length}(\text{ground\_truth\_map}) \quad (2)$$

$$\text{precision} = \text{length}(\text{TP}) / \text{length}(\text{reconstructed\_map}) \quad (3)$$

$$\text{F-score} = \frac{2(\text{recall} \bullet \text{precision})}{\text{recall} + \text{precision}} \quad (4)$$

where TP is a set of constructed roads that match the underlying ground-truth roads with related to a matching distance threshold, and  $\text{length}(\text{TP})$  is the total length of all the roads in TP.



**Figure 6.** Experimental trajectory data sets for construction of new roads: (a) Athens small; (b) Berlin; (c) Chicago; (d) Wuhan-D1; (e) Wuhan-D2; (f) Wuhan-D3; (g) Joensuu

**Table 1.** Statistics for datasets used in experiment I.

ID	Dataset	Country	Number of Trajectories	Average Sampling Rate (s)	Trajectory Length (km)
1	Athens small	Greece	129	34.07	443
2	Berlin	Germany	26,831	41.98	41,116
3	Chicago	USA	889	3.61	2869
4	Wuhan-D1	China	5843	49.24	4107
5	Wuhan-D2	China	9584	49.24	5471
6	Wuhan-D3	China	2706	49.24	3253
7	Joensuu	Finland	108	2.00	250

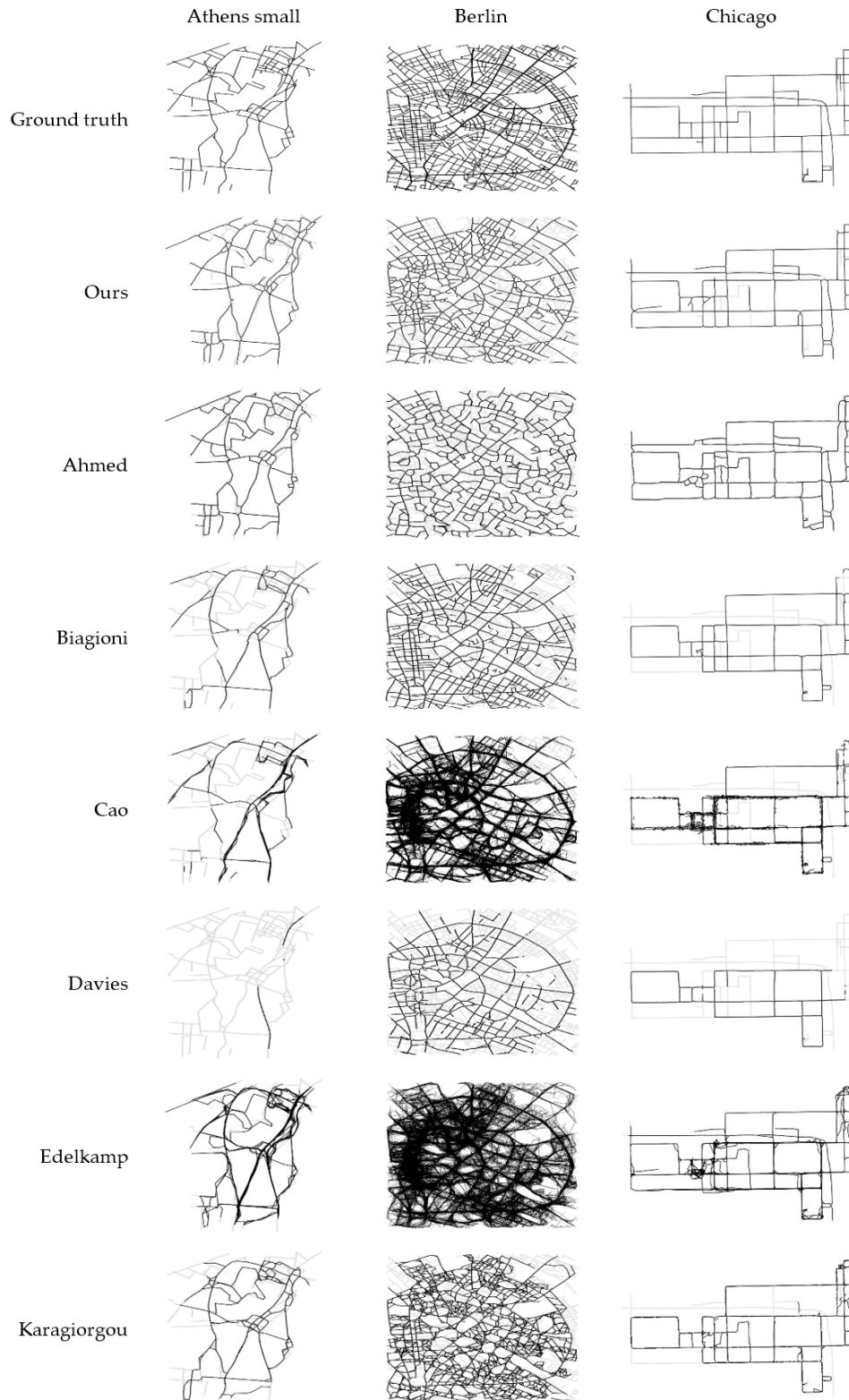
**Table 2.** Seven state-of-the-art map generation algorithms.

Number	Source	Method
1	Edelkamp & Schrödl (2003)	Point clustering
2	Davies et al. (2006)	Density image skeleton extraction
3	Cao & Krumm (2009)	Incremental track insertion
4	Biagioni & Eriksson (2012)	Density image skeleton extraction
5	Ahmed & Wenk (2012)	Incremental track insertion
6	Karagiorgou & Pfoser (2012)	Intersection linking
7	Mariescu-Istodor & Fränti (2018)	Intersection linking

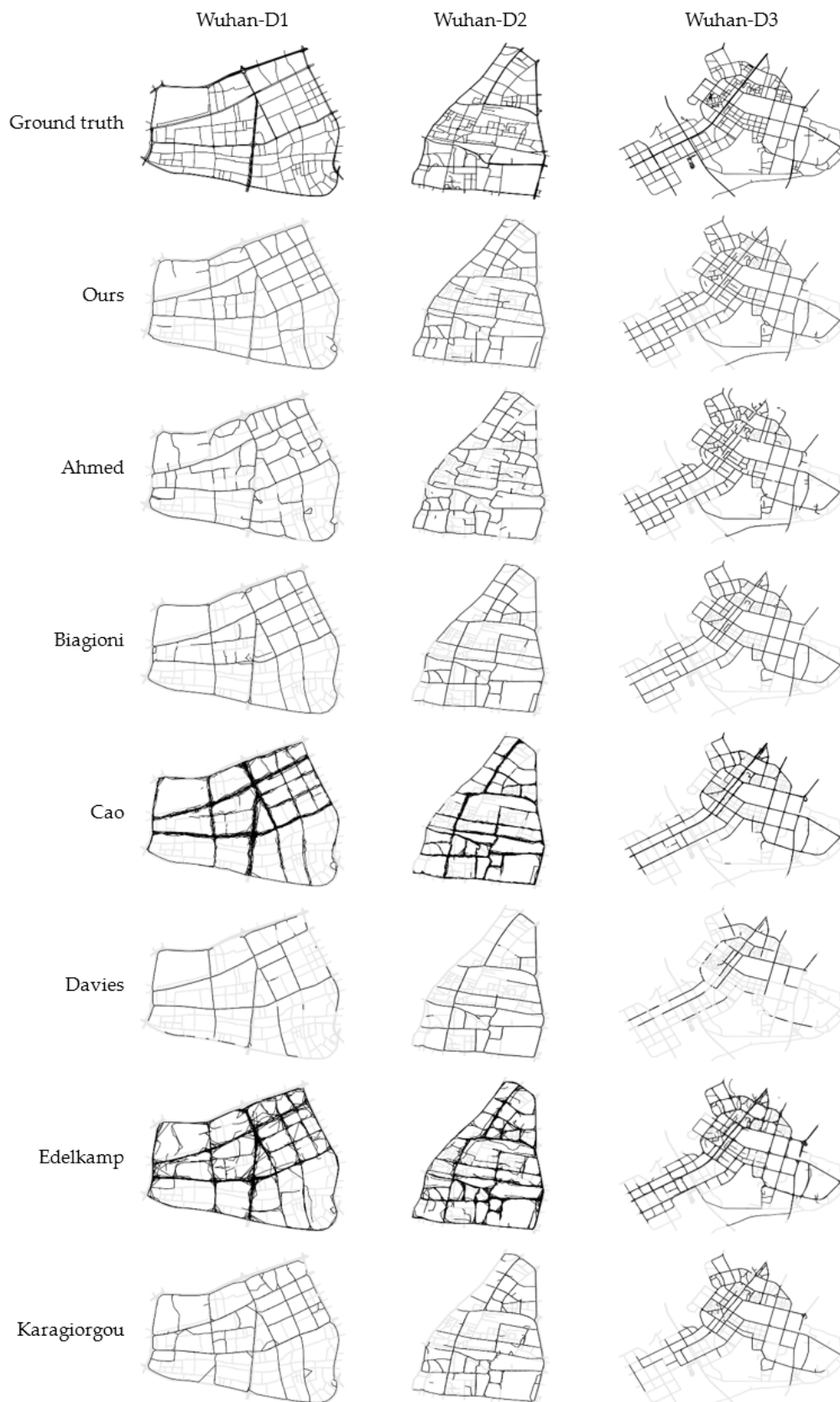
For the four public trajectory data sets of Athens small, Berlin, Chicago, and Joensuu, the ground-truth road networks were obtained from <http://mapconstruction.org>, and <http://cs.uef.fi/mopsi/routes/network>. For the datasets Wuhan-D1, Wuhan-D2, and Wuhan-D3, the ground-truth road networks were downloaded from the OSM website ([www.openstreetmap.org](http://www.openstreetmap.org)). To avoid the bias in recall computing, a subset rather than the whole underlying road network was used as a ground-truth map in the performance evaluation because the raw trajectory data only covered parts of the underlying road network. We overlapped the raw trajectory data and the underlying road network together and manually selected the underlying roads that were traversed by one or more trajectory as the ground-truth map. The ‘reconstructed\_map’ in Equation (3) was a set of constructed roads by the map generation algorithm based on the input GPS trajectory data.

We first tested our method using the low sampling trajectory data sets of Athens small, Berlin, Wuhan-D1, Wuhan-D2, and Wuhan-D3. The generated road networks by the proposed method and the first six map construction algorithms in Table 2 are shown in Figures 7 and 8. Since the Chicago data set was commonly used as benchmark data to evaluate the performance of map construction algorithms in the literature, the results for the Chicago data set are also displayed for comparison. It can be seen that most of the underlying roads were correctly constructed by the proposed method, and roads traversed infrequently were also found by our method. In the experiments, the clustering parameters for Ahmed and Wenk’s method were all set to 120 m for Wuhan-D1, Wuhan-D2, and Wuhan-D3, and the parameters of these six algorithms (i.e., Ahmed and Wenk’s method, Edelkamp and Schrödl’s method, Davies et al.’s method, Cao and Krumm’s method, Biagioni and Eriksson’s method, and Karagiorgou and Pfoser’s method) were set according to Ahmed et al. [16]. By comparing the generated road networks (in black) by visual inspection with the underlying road maps (in grey), it can be seen that the proposed method generated better results than the other algorithms, followed by Biagioni and Eriksson’s method, Karagiorgou and Pfoser’s method, and Ahmed and Wenk’s method. Cao and Krumm’s method and Edelkamp and Schrödl’s method both performed poorly because they failed to cluster trajectories together in regions where many trajectories exist, and they produced multiple edges on a single road. Davies et al.’s method generated the smallest number of edges and identified only those underlying roads that a large number of trajectories traveled. Biagioni and Eriksson’s method, and Karagiorgou and Pfoser’s method produced the better results among the six algorithms; however, these two methods may also lose some important roads where only a few

trajectories passed through; for example, the roads at the bottom-left corner for the Chicago data. Mariescu-Istodor and Fränti’s method was not compared because of the computing complexity of their method for a large number of trajectories (such as the Berlin and Wuhan-D1 data sets).



**Figure 7.** Generated road networks (in black) and the ground-truth road maps (in grey) for data sets of Athens small, Berlin, and Chicago.



**Figure 8.** Generated road networks (in black) and the ground-truth road maps (in grey) for the data sets of Wuhan-D1, Wuhan-D2, and Wuhan-D3.

Figure 9 shows the F-scores of the results by Ahmed and Wenk’s method, Biagioni and Eriksson’s method, Davies et al.’s method, Karagiorgou and Pfofer’s method, and our method, respectively.

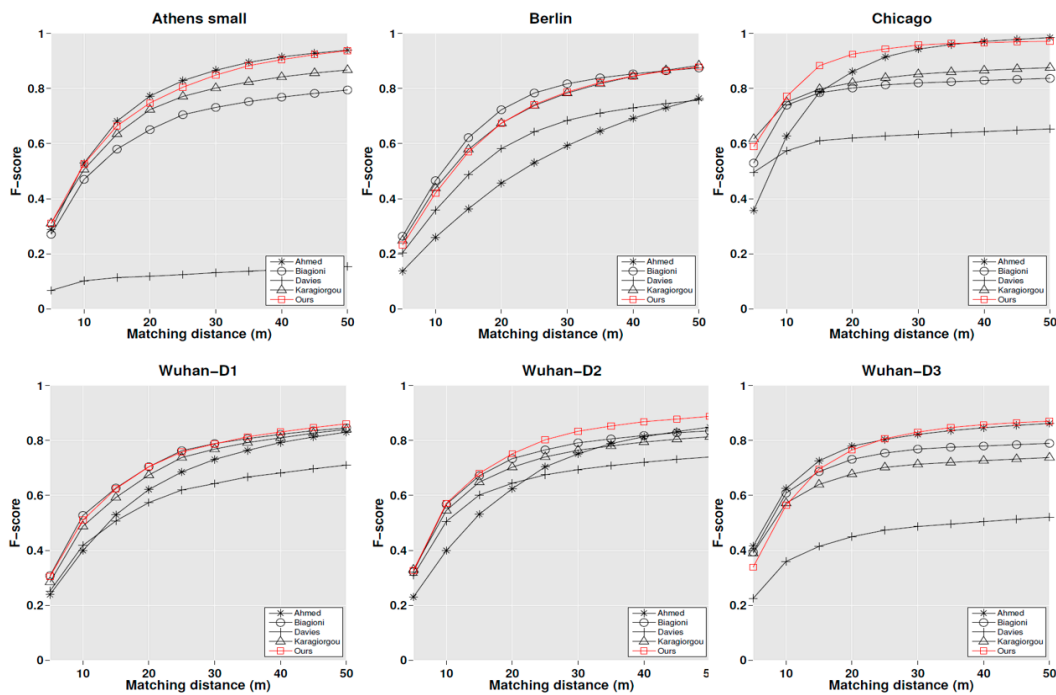


Figure 9. F-scores of the test algorithms for several matching distances on the test data sets.

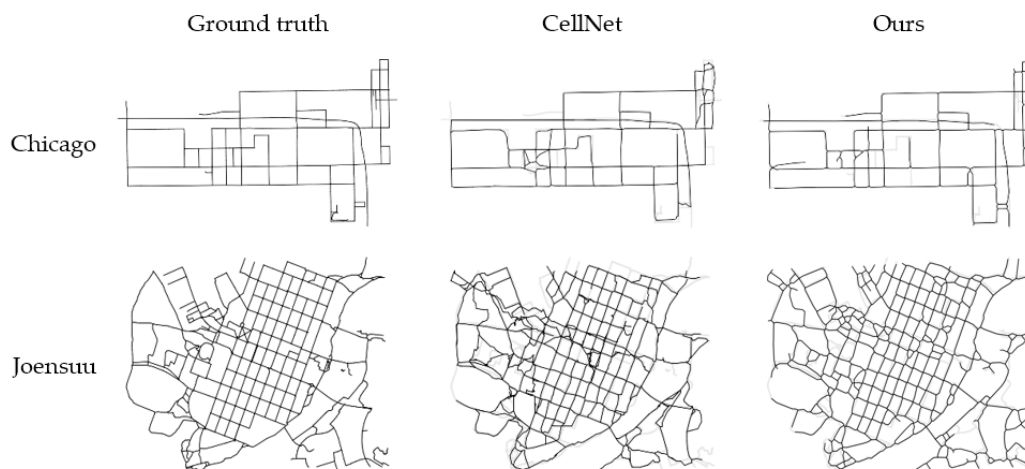
It can be seen that Ahmed and Wenk’s method performed best on Athens small data, followed by our method, and Karagiorgou and Pfofer’s method. Among these methods, Davies et al.’s method found only parts of the underlying roads and has relatively high precision, but its recall is lower than other algorithms. For the Chicago data, our method generated a better result than the other algorithms. For the Berlin data, Biagioni and Eriksson’s method performed slightly better than our method and Karagiorgou and Pfofer’s method, and Ahmed and Wenk’s method performed worst and produced numerous spurious roads due to the impact of noise. For low-quality taxi trajectory data in Wuhan (i.e., Wuhan-D1, Wuhan-D2, and Wuhan-D3), Ahmed and Wenk’s method performed better than ours on Wuhan-D3 data for small matching distances; and on other two datasets, our method produced better results than the other algorithms.

Both qualitative and quantitative analysis of the test data show that the proposed method generally performed better than other algorithms, followed by Biagioni and Eriksson’s method, Karagiorgou and Pfofer’s method, and Ahmed and Wenk’s method. Davies et al.’s method performed poorly because of the uneven density distribution of GPS trajectory data, and this method is only suitable for finding roads that are traversed frequently. Ahmed and Wenk’s method and Karagiorgou and Pfofer’s method may generate some spurious edges due to noise. Biagioni and Eriksson’s method is robust to noise, and it has high precision, but some important road segments (such as newly-built roads) with sparse trajectories may be omitted. In order to find roads with sparse trajectories, our method sacrifices some precision to ensure a higher recall rate and compromises the accuracy and the completeness of the reconstructed road network. It should be noted that the F-scores in Figure 8 are different from the results of Ahmed et al. [16], because the ground truth maps are different in these two studies (Ahmed et al. used the whole original road network to compute the F-scores).

Further, two trajectory data sets with high sampling rates (i.e., the Chicago and Joensuu data sets) were used to evaluate the performance of the proposed method. The recently developed map construction algorithm by Mariescu-Istodor and Fränti [23] (called CellNet hereafter) was also

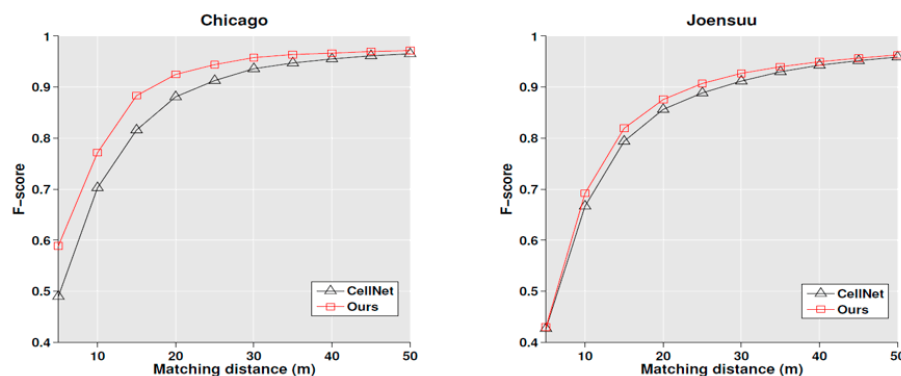


compared. The generated road networks for the Chicago and Joensuu data sets by the proposed method and CellNet are shown in Figure 10.



**Figure 10.** Results of generated road networks (in black) by CellNet and the proposed method and the ground-truth road maps (in grey) for data sets from Chicago and Joensuu.

By visual inspection, it can be seen that both CellNet and the proposed method can successfully construct the road network from the input raw trajectories, even for roads with a few trajectories traveled. The graph of F-scores for the proposed method and CellNet for several matching distances on the two test data sets are shown in Figure 11. It can be seen that for both the Chicago data and the Joensuu data, the F-scores of the proposed method are slightly higher than those of the CellNet algorithm. The CellNet algorithm may also generate multiple paths for a single road since the intersection could be identified as multiple parts due to the impact of noise. In the proposed method, the noise sampling points will not be clustered and the road centerlines are generated through an optimal piecewise linear curve fitting from the point clusters, which can also reduce the impact of noise points. Thus, compared with existing methods, the proposed method shows more robustness against noise. On the other hand, the proposed method generates roads from clusters of similar sampling points (usually on the same road) instead of finding the high-density trajectories on the roads; therefore, the proposed method can also find the roads with sparse trajectories.



**Figure 11.** F-scores of the proposed method and CellNet for several matching distances on the test data sets.

#### 4.1.1. Time Complexity

The proposed method is implemented in Octave, and the implementations of the first six algorithms in Table 2 are available at <http://mapconstruction.org>. Due to these algorithms being implemented

based on different coding languages (i.e., Java, Python, and MATLAB/Octave), the algorithms' running times are not comparable in theory. However, to at least give an impression, Table 3 shows the respective running times of these algorithms on the test data sets for the construction of road networks. All these algorithms were run on Intel Core i3 CPUs running at 3.0 GHz with 8 GB of RAM using a Windows 7 operating system.

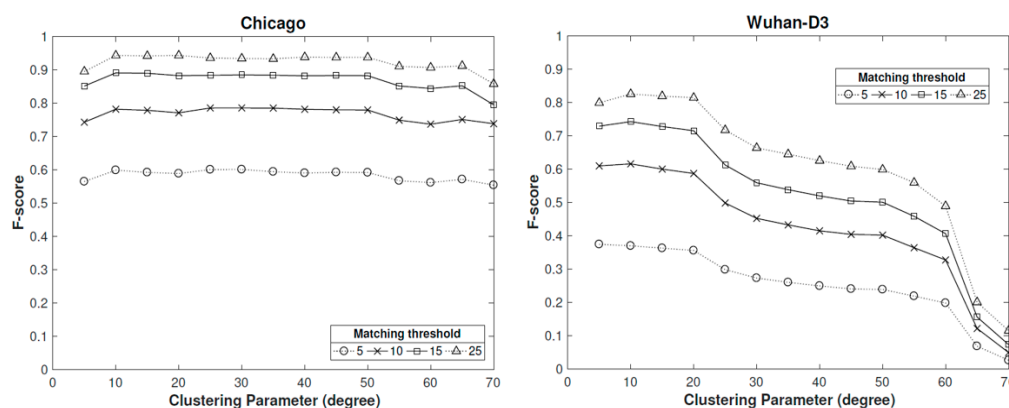
**Table 3.** Running times of various algorithms on the test datasets (min).

Dataset	Ahmed (Java)	Biagioni (Python)	Cao (Python)	Davies (Python)	Edelkamap (Python)	Karagiorgou (MATLAB)	Ours (Octave)
Athens small	0.087	16.19	4.73	0.41	1.47	6.95	0.56
Berlin	82.54	301.70	727.96	108.77	>20 h	>48 h	13.14
Chicago	23.67	31.25	373.01	1.68	18.49	>20 h	3.64
Wuhan-D1	6.15	54.40	127.48	4.86	42.44	426.44	3.57
Wuhan-D2	13.48	44.21	162.43	10.67	84.64	451.28	4.06
Wuhan-D3	10.65	20.78	21.93	9.38	53.46	85.67	1.51

For the CellNet algorithm, the above-compared results of the Chicago data and the Joensuu data were obtained from <http://cs.uef.fi/mopsi/routes/network>. As reported in [23], the running times of the CellNet algorithm for the Chicago data and the Joensuu data were about 2 hours and 1 hour, respectively. The proposed method required 48.18 seconds for the Joensuu data on the above system platform (i.e., Intel Core i3 CPUs with 8 GB of RAM using a Windows 7 operating system).

#### 4.1.2. Parameter Sensitivity

For our method, the accuracy of the construction of new roads is very related to the graph-based clustering results. The developed graph-based clustering algorithm needs one parameter ( $\Lambda$ ) to set.  $\Lambda$  is an angle threshold to determine whether two directions are similar. In this paper, the angle threshold  $\Lambda$  is set to  $15^\circ$  by default, in accordance to previous studies. To analyze the impact of the values of  $\Lambda$  (i.e., the clustering parameter) on the performance of the construction of new roads, we conducted a parameter sensitivity analysis using the Chicago data (high-quality GPS data) and the Wuhan-D3 data (low-quality GPS data). Some results of this analysis are shown in Figure 12.

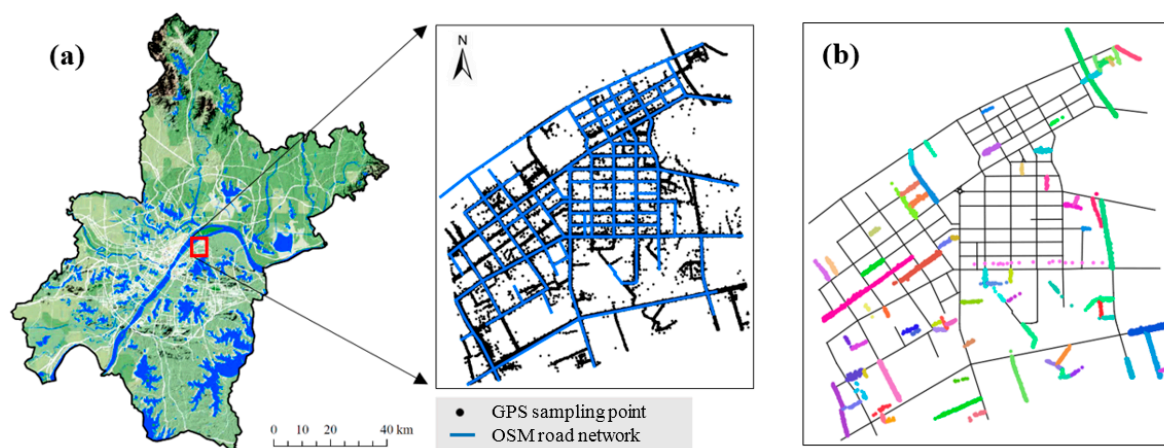


**Figure 12.** Parameter sensitivity test on the Chicago and Wuhan-D3 datasets.

As the figure illustrates, for the Chicago data, when the clustering parameter ( $\Lambda$ ) is within the range of  $5^\circ$  to  $70^\circ$ , the F-scores are relatively stable and decrease slightly when  $\Lambda$  reaches  $70^\circ$ . For the Wuhan-D3 data, the F-score of the constructed roads decreases with increasing  $\Lambda$ . From both data sets, it can be found that when  $\Lambda$  is within the range of  $5^\circ$  to  $25^\circ$ , the F-score of the constructed roads is relatively high and stable. This is consistent with the suggested value of the angle threshold as to whether two directions are similar according to the previous studies [33,34]. Therefore,  $\Lambda$  is set to  $15^\circ$  in the proposed method.

#### 4.2. Experiment II: Application of the Proposed Method for the Update of the Road Network

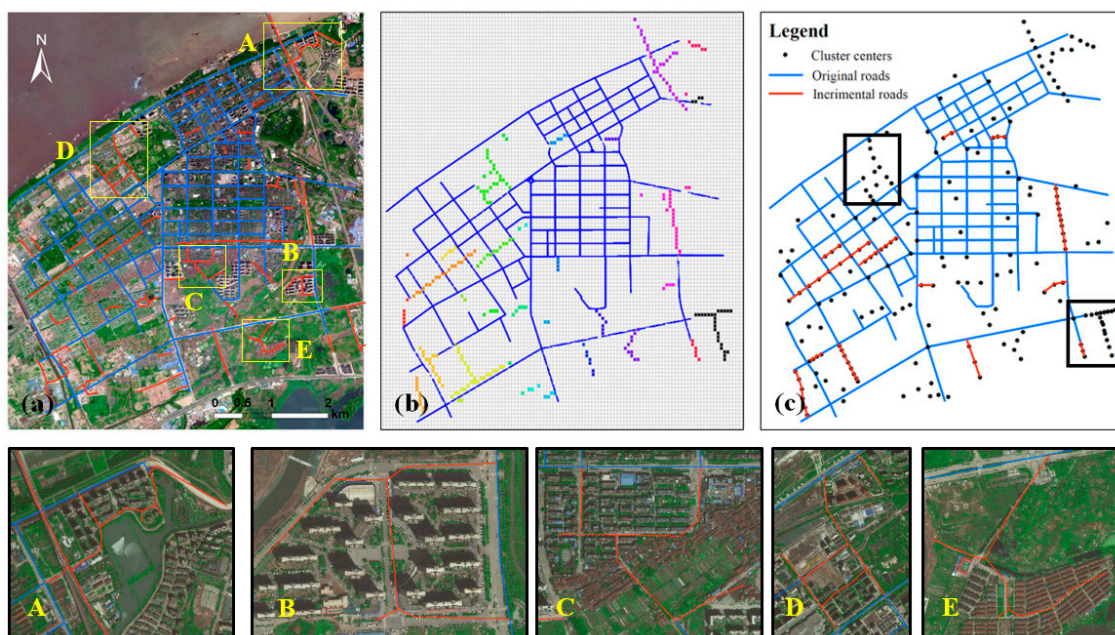
In Experiment II, the proposed method was applied for the update of the OSM road network in the Qingshan district of Wuhan, China. The OSM road network of this area in 2014 was used as the original road network to be updated. The GPS trajectory data from 6150 taxis on May 1, 2015, were used as the input trajectory data, which included 10,662 trajectories, and the sampling rate varied from 10 to 180 seconds. The original OSM road network and the GPS sampling points in the study area are shown in Figure 13a. The clusters detected from the unmatched sampling points are shown in Figure 13b.



**Figure 13.** Study area and the clustering result: (a) the open street map (OSM) road network and the GPS sampling points in the study area; (b) clustering result by our method and the clusters are shown with different colors.

Figure 14a shows the updated road network with newly added roads in the study area. By overlapping the updated road network on the satellite image, it can be seen that the newly added roads constructed by our method match the underlying ground-truth roads in the image well (see the locally enlarged view of regions A, B, C, D, and E in Figure 14a). The newly added roads generated by Zhao et al.'s method and Wu et al.'s method are shown in Figure 14b and Figure 14c, respectively. Zhao et al.'s method detected most of the additive roads in the study area, but some new roads with sparse trajectories were not found. For Wu et al.'s method, H and T-shaped newly added roads were not well constructed because of their assumptions (the unmatched trajectories inside a problem area follow a unique path); for example, roads in the black boxes in Figure 14c where there are several newly added roads in the same area that need to be updated.

We evaluated the updated road network in Figure 14a by using the updated OSM road network in 2018 as the ground-truth road map. The F-score was 0.725 when the matching distance threshold was set to 20 m. When compared with the satellite image, we found some underlying newly added roads, such as roads in region B and E, are still missing in the OSM road data in 2018, which may reduce the F-score of our result.



**Figure 14.** Updated road network: (a) results of our method; (b) results of Zhao et al.'s method (buffer size = 30 m; raster size =  $128 \times 128$ ; density threshold =  $0.2 \times 10^{-6}$ ); (c) results of Wu et al.'s method (cluster number = 200).

## 5. Conclusions

This paper proposes an automatic method for the detection and updating of the newly added roads of the existing road network based on low-quality GPS trajectory data. Two experiments were designed to evaluate the proposed method. Six real-world GPS trajectory data sets were used to test the performance on the construction of new roads of the proposed method, and the taxi trajectory data in Wuhan, China was used to test the ability of the proposed method for the update of the road network. Experimental results demonstrated the validity, efficiency, and superiority of the proposed method. Compared with six state-of-the-art map reconstruction algorithms and two local update algorithms, the proposed method performed better on low-quality GPS data (see Figure 8). The proposed method can recognize newly added roads that are traversed sparsely, which most state-of-the-art map generation and update algorithms may fail to find (see Figure 7).

The proposed method still has some limitations. First, the developed graph-based clustering algorithm tends to divide roads at the intersection, thus the detailed geometry of the intersection will not be well reconstructed. In our future work, fine structures of the intersection will be generated and added to the updated road network. Based on the updated road network, the location of road intersections can be easily found by employing network analysis, and the fine geometry of the intersection can then be further extracted. Second, removal, geometry, and topology changes of the road network are not considered in this study. Those changes of road networks will be studied in our future work. In general, the experiments demonstrated the ability and efficiency of the proposed method for detection and updating of the newly added roads based on the commonly available low-quality trajectory data. In the age of autonomous vehicles, the proposed method could be used to quickly find the new, additive change-areas in the original road network. This may guide the professional surveying vehicles to update the high-resolution road map of these change areas in time.

**Author Contributions:** Min Deng and Jianbo Tang were responsible for the research design and experiments. Jincai Huang led the writing of the paper, with Huimin Liu and Xueying Chen contributing material.



**Acknowledgments:** This research was funded by the funds of the National Science Foundation of China (grant numbers 41901406, 4173000380, 41730105, 41771492 and 41601424), the National Key Research and Development Program of China (grant number 2017YFB0503500), the Cooperative Research and Development Project of BOTONG Smart City Research Institute (grant number BTZH2018002) and the Philosophy and Social Science Foundation of Hunan Province, China (grant number 18YBQ050). We also express our thanks to the anonymous reviewer for constructive comments.

**Conflicts of Interest:** The authors declare no conflict of interest. The founding sponsors had no role in the design of the study, nor in the collection, analyses, or interpretation of data, nor in the writing of the manuscript or the decision to publish the results.

## References

1. Zhang, Y.C.; Liu, J.P.; Qian, X.L.; Qiu, A.G.; Zhang, F.H. An automatic road network construction method using massive gps trajectory data. *ISPRS Int. J. Geo Inf.* **2017**, *6*, 400. [\[CrossRef\]](#)
2. Tang, L.L.; Ren, C.; Liu, Z.; Li, Q.Q. A road map refinement method using Delaunay triangulation for big trace data. *ISPRS Int. J. Geo Inf.* **2017**, *6*, 45. [\[CrossRef\]](#)
3. Edelkamp, S.; Schrodl, S. Route planning and map inference with global positioning traces. In *Computer Science in Perspective*; Springer: New York, NY, USA, 2003; Volume 2598, pp. 128–151.
4. Davies, J.J.; Beresford, A.R.; Hopper, A. Scalable, distributed, real-time map generation. *IEEE Pervas. Comput.* **2006**, *5*, 47–54. [\[CrossRef\]](#)
5. Cao, L.; Krumm, J. From GPS traces to a routable road map. ACM SIGSPATIAL international conference on advances in geographic information systems. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 4–6 November 2009; pp. 3–12.
6. Ahmed, M.; Wenk, C. Constructing street networks from GPS trajectories. In Proceedings of the European Symposium on Algorithms, Ljubljana, Slovenia, 10–12 September 2012; pp. 60–71.
7. Biagioni, J.; Eriksson, J. Map inference in the face of noise and disparity. In Proceedings of the ACM 20th International Conference on Advances in Geographic Information Systems, Redondo Beach, CA, USA, 6–9 November 2012; pp. 79–88.
8. Karagiorgou, S.; Pfoser, D. On vehicle tracking data-based road network generation. In Proceedings of the ACM 20th International Conference on Advances in Geographic Information Systems, Redondo Beach, CA, USA, 6–9 November 2012; pp. 89–98.
9. Wang, J.; Rui, X.P.; Song, X.F.; Tan, X.S.; Wang, C.L.; Raghavan, V. A novel approach for generating routable road maps from vehicle GPS traces. *Int. J. Geogr. Inf. Sci.* **2015**, *29*, 69–91. [\[CrossRef\]](#)
10. Mori, U.; Mendiburu, A.; Alvarez, M.; Lozano, J.A. A review of travel time estimation and forecasting for advanced traveller information systems. *Transp. A Trans. Sci.* **2015**, *11*, 119–157. [\[CrossRef\]](#)
11. Yang, X.; Tang, L.L.; Stewart, K.; Dong, Z.; Zhang, X.; Li, Q.Q. Automatic change detection in lane-level road networks using gps trajectories. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 601–621. [\[CrossRef\]](#)
12. Deng, M.; Huang, J.C.; Zhang, Y.F.; Liu, H.M.; Tang, L.L.; Tang, J.B.; Yang, X.X. Generating urban road intersection models from low-frequency gps trajectory data. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 2337–2361. [\[CrossRef\]](#)
13. Wang, W.; Jin, J.; Ran, B.; Guo, X.C. Large-scale freeway network traffic monitoring: A map-matching algorithm based on low-logging frequency GPS probe data. *J. Intell. Transp. Syst.* **2011**, *15*, 63–74. [\[CrossRef\]](#)
14. Huang, J.C.; Deng, M.; Tang, J.B.; Hu, S.L.; Liu, H.M.; Wariyo, S.; He, J.Q. Automatic generation of road maps from low quality GPS trajectory data via structure learning. *IEEE Access* **2018**, *6*, 71965–71975. [\[CrossRef\]](#)
15. Schroedl, S.; Wagstaff, K.; Rogers, S.; Langley, P.; Wilson, C. Mining GPS traces for map refinement. *Data Min. Knowl. Disc.* **2004**, *9*, 59–87. [\[CrossRef\]](#)
16. Ahmed, M.; Karagiorgou, S.; Pfoser, D.; Wenk, C. A comparison and evaluation of map construction algorithms using vehicle tracking data. *Geoinformatica* **2015**, *19*, 601–632. [\[CrossRef\]](#)



17. Chen, C.; Cheng, Y.H. Roads digital map generation with multi-track GPS data. In Proceedings of the International Workshop on Geoscience and Remote Sensing, Shanghai, China, 21–22 December 2008; pp. 508–511.
18. Shi, W.H.; Shen, S.H.; Liu, Y.C. Automatic generation of road network map from massive GPS vehicle trajectories. In Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems (ITSC 2009), St. Louis, MO, USA, 4–7 October 2009; pp. 48–53.
19. Wang, S.Y.; Wang, Y.S.; Li, Y.J. Efficient map reconstruction and augmentation via topological methods. In Proceedings of the 23rd ACM International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 3–6 November 2015.
20. Worrall, S.; Nebot, E. Automated process for generating digitised maps through GPS data compression. In *Australasian Conference on Robotics and Automation*; ACRA: Brisbane, Australia, 2007; Volume 6.
21. Agamennoni, G.; Nieto, J.I.; Nebot, E.M. Robust inference of principal road paths for intelligent transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 298–308. [[CrossRef](#)]
22. Kuntzsch, C.; Sester, M.; Brenner, C. Generative models for road network reconstruction. *Int. J. Geogr. Inf. Sci.* **2016**, *30*, 1012–1039. [[CrossRef](#)]
23. Mariescu-Istodor, R.; Fränti, P. CellNet: Inferring road networks from GPS trajectories. *ACM Trans. Spat. Algorithms Syst.* **2018**, *4*, 1–22. [[CrossRef](#)]
24. Wu, T.; Xiang, L.G.; Gong, J.Y. Updating road networks by local renewal from gps trajectories. *ISPRS Int. J. Geo Inf.* **2016**, *5*, 163. [[CrossRef](#)]
25. Zhao, Y.; Liu, J.; Chen, R.Q.; Li, J.; Xie, C.; Niu, W.J.; Geng, D.Y.; Qin, Q.M. A new method of road network updating based on floating car data. In Proceedings of the 2011 IEEE International Geoscience and Remote Sensing Symposium, Vancouver, BC, Canada, 24–29 July 2011; pp. 1878–1881.
26. Fang, W.; Hu, R.; Xu, X.; Xia, Y.; Hung, M.H. A novel road network change detection algorithm based on floating car tracking data. In *Telecommunication Systems*; Springer: New York, NY, USA, 2016; pp. 1–7.
27. Tang, L.L.; Huang, F.; Zhang, X.; Xu, H. Road network change detection based on floating car data. *J. Netw.* **2012**, *7*, 1063–1070. [[CrossRef](#)]
28. Wang, Y.; Liu, X.; Wei, H.; Forman, G.; Chen, C.; Zhu, Y. CrowdAtlas: Self-updating maps for cloud and personal use. In Proceedings of the ACM International Conference on Mobile Systems, Taipei, Taiwan, 25–28 June 2013; pp. 27–40.
29. Shan, Z.Q.; Wu, H.; Sun, W.W.; Zheng, B.H. Cobweb: A robust map update system using GPS trajectories. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp 2015), Osaka, Japan, 7–11 September 2015; pp. 927–937.
30. Brakatsoulas, S.; Pfoser, D.; Salas, R.; Wenk, C. On map-matching vehicle tracking data. In Proceedings of the 31st International Conference on Very Large Databases, Trondheim, Norway, 30 August–2 September 2005; pp. 853–864.
31. Newson, P.; Krumm, J. Hidden Markov map matching through noise and sparseness. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems—GIS, Seattle, WA, USA, 4–6 November 2009.
32. Chen, B.Y.; Yuan, H.; Li, Q.Q.; Lam, W.H.K.; Shaw, S.L.; Yan, K. Map-matching algorithm for large-scale low-frequency floating car data. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 22–38. [[CrossRef](#)]
33. Li, Z.; Yan, H.; Ai, T.; Chen, J. Automated building generalization based on urban morphology and gestalt theory. *Int. J. Geogr. Inf. Sci.* **2004**, *18*, 513–534. [[CrossRef](#)]
34. Deng, M.; Tang, J.B.; Liu, Q.L.; Wu, F. Recognizing building groups for generalization: A comparative study. *Cartogr. Geogr. Inf. Sci.* **2018**, *45*, 187–204. [[CrossRef](#)]
35. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*; Institute for Computer Science, University of Munich: Munich, Germany, 1996; pp. 226–231.
36. Deng, M.; Liu, Q.L.; Cheng, T.; Shi, Y. An adaptive spatial clustering algorithm based on Delaunay triangulation. *Comput. Environ. Urban.* **2011**, *35*, 320–332. [[CrossRef](#)]
37. Pittman, J.; Murthy, C.A. Fitting optimal piecewise linear functions using genetic algorithms. *IEEE Trans. Pattern Anal.* **2000**, *22*, 701–718. [[CrossRef](#)]

38. Lam, L.; Lee, S.W.; Suen, C.Y. Thinning methodologies—A comprehensive survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE Compute Society: Washington, DC, USA, 1992; Volume 14, pp. 869–885. [[CrossRef](#)]
39. Douglas, D.; Peucker, T. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Can. Cartograph.* **1973**, *10*, 112–122. [[CrossRef](#)]



© 2019 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/>).