Movement-aware map construction

Haiyang Lyu[a,b,c], Dieter Pfoser[b], Yehua Sheng[c,d]

[a] Smart Health Big Data Analysis and Location Services Engineering Lab of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China, 210023; [b] Geography and Geoinformation Science, George Mason University, Fairfax, VA, USA, 22030; [c] Key Laboratory of Virtual Geographic Environment (Nanjing Normal University), Ministry of Education, Nanjing, Jiangsu, China, 210023; [d] Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing, Jiangsu, China, 210023

Abstract: Map construction algorithms attempt to derive a spatial graph representing a road network from GSP-sampled movement trajectories. Existing methods commonly use the trajectory without considering the specific sampling methodology, hence, the movement information is not preserved in the map construction results. The proposed map-construction method considers the particularities of the sampling process and how they affect the trajectory data to improve the overall result quality. Specifically, our proposed algorithm constructs nodes first by using weights to cluster sampled turn points based on their estimated proximity to the actual turn location. As nodes are the aggregate of turn points, edges are constructed using the respective trajectories that connect turn points and by attracting nearby trajectory segments that are with little turn information based on their cluster weights. An edge is constructed by aggregating the trajectory segments between nodes using so-called segment conflation. Experiments using trajectory datasets at different spatial scales, data complexities, and data sources in combination with several assessment methods show that the movement-aware map construction method produces maps of greater accuracy than the experimented existing approaches.

Keywords: VGI trajectories, map construction, movement-aware, node inference

1. Introduction

With the proliferation of the Internet as the primary medium for data publishing and information exchange, we have seen an explosion in the amount of online content available on the Web. The volumes of such User-Generated Content (UGC) are already staggering and constantly growing. With geospatial content playing an essential UGC role (Pfoser, 2011), research in this area takes advantage of this explosion in Volunteered Geographic Information (VGI) (Goodchild, 2007; Haklay & Weber, 2008) to produce datasets that complement authoritative datasets rather than replace them (Yang & Tang et al., 2018; Wu et al., 2019). User-generated geospatial content has been categorized in several ways. *Explicit content* is generated purposefully in the desired form. An example here is the OpenStreetMap road network. In contrast, *implicit content* reflects derived information from user-generated content that was produced with a different purpose in mind. However, the desired content can still be derived by using targeted data mining efforts, e.g., road boundaries (Yang et al., 2018), lanes (Tang et al., 2016), intersections (Deng et al., 2018), and turn restrictions (Efentakis et al., 2014; Wang et al., 2017). In our case, this translates to constructing a road network from GPS tracking data.

So-called map construction algorithms rely on the fact that movement typically recorded as trajectories is constrained by an underlying transportation network. Aggregating such trajectory datasets recorded by, for example, a myriad of smartphones produces up-to-date road networks (Zheng & Zhou, 2011; Ahmed et al., 2015b; Karagiorgou et al., 2017; Ni et al., 2018; Bhattacharjee et al., 2019). Such VGI-derived road network construction is especially important in areas of little commercial interest, e.g., in developing countries, or in situations where updates are costly, such as rural areas. However, VGI data quality is highly dependent on the device and application used (Zandbergen, 2009), and there is a lack of unified rules and regulations to standardize the contributed trajectories (Neis & Zipf, 2012; See et al., 2013), which poses significant challenges (Flanagin & Metzger, 2008; Haklay, 2010; Lee & Krumm, 2011; Fan et al., 2014; Mehdipoor et al., 2015; Karagiorgou et al., 2017; He & Bastani et al., 2018) for map construction research. Due to a heterogeneity of data sources, a method that works for one dataset may not perform well for others. Further, as trajectories are an approximate representation of movement affected by measurement (GPS) and sampling error (Pfoser & Jensen, 1999), they do not accurately capture the geometric features of the underlying road network and deriving a road network from a collection of trajectories becomes a data mining challenge.

A comprehensive overview of existing map construction algorithms is given in (Ahmed et al., 2015b). Distinguishing three basic categories of algorithms, (i) *point clustering methods* include *k*-means algorithms, density-based methods, and approaches based on neighborhood complexes, (ii) *incremental track insertion* methods construct a street map by incrementally inserting trajectories into an initially empty map, and (iii) *intersection linking* methods focus on the correct detection of intersections and then linking them with edges.

This work proposes a novel intersection linking approach termed *Movement-Aware Map Construction* (MAMC). At the core the method, nodes (intersections) are derived from clustered turn points. We use an adaptive clustering approach that considers when a turn point was sampled in relation to the "true" node location based on the trajectory geometry. Edges are constructed by conflating incident as well as adjacent trajectory segments using a so-called arc editing approach.

The remainder of this paper is structured as follows. Section 2 discusses related wok. Section 3 provides a detailed description of the movement-aware map construction algorithm. Experiments and results are discussed in Section 4, followed by conclusions and future work in Section 5.

2. Related Work

Trajectory data is a semantically rich data source that has resulted in a large number of research directions ranging from database research and data mining to more recent trends such as urban science. In relation to this effort, map construction algorithms generally differ in how they consider the error that is associated with trajectories when recording movement with respect to infrastructure such as road networks. Trajectories are a representation of movement that is affected by a measurement error (e.g., GPS) and a sampling error, i.e., the movement being sampled at regular time intervals (Pfoser & Jensen, 1999). If both errors would be close to 0 then the trajectories could be used directly to reconstruct the network. Although affected by uncertainty, the trajectories can be assumed to follow a normal distribution around the true shape of the road (Guo, Iwamura, & Koga, 2007). What follows is a discussion of the various approaches on how to use such noisy data to reconstruct a road network. The various techniques can be categorized as follows (Ahmed et al., 2015a).

1. Density-based map construction methods such as (Edelkamp & Schrödl 2003; Davies et al., 2007; Worrall & Nebot, 2007; Shi et al., 2009; Ge et al., 2011; Aanjaneya et al., 2012; Biagioni & Eriksson, 2012b; Wang, Wang, & Li, 2015; Kuntzsch et al., 2016; Li et al., 2016; Chen et al., 2016; He et al., 2018; Huang et al., 2018; Ni et al., 2018; Bhattacharjee et al., 2019) take the density distribution of all the trajectory position samples and extract the density peak lines with a maximum local sampling density;

2. Movement-based map construction methods such as (Bruntrup et al., 2005; Cao & Krumm, 2009; Fathi & Krumm, 2010; Zhang et al., 2010; Torre et al., 2012; Karagiorgou & Pfoser, 2012; Karagiorgou et al., 2013;

Kasemsuppakorn & Karimi, 2013; Ahmed et al., 2015a, Wang et al., 2015, Xie et al., 2015; Karagiorgou et al., 2017; Zhang et al., 2017; He et al., 2018; Mariescu-Istodor, et al., 2018) utilize the movement aspect of the trajectories, i.e., not just a point cloud, to construct a road network and preserve those features in the resulting network.

In the case of the *density-based map construction methods*, two kinds of methods are commonly used: (i) kernel density estimation (KDE) (Davies et al., 2007; Shi et al., 2009; Biagioni & Eriksson, 2012b; Wang et al., 2015; Kuntzsch et al., 2016), and (ii) the point cluster method (Worrall & Nebot, 2007; Ge et al., 2011; Aanjaneya et al., 2012; Li et al., 2016; Chen et al., 2016; Ni et al., 2018; Bhattacharjee et al., 2019). KDE approaches typically divide space into small grid cells and density distributions of position samples are aggregated using a kernel function. For example, Davies et al. (2007) propose a scalable, distributed, real-time based map construction method by counting the trajectories that pass through each grid cell, computing a 2D histogram for the whole gridded area, and constructing the road network by deducing edge positions and computing the centerlines. Shi et al. (2009) compute the density distribution by constructing a bitmap according to the trajectory sampling points and construct the road network graph using a thinning algorithm. Wang et al. (2015) propose a topological map construction method from trajectory density images according to Morse theory, and road maps are constructed by extracting stable manifolds with local maximum density values. Biagioni & Eriksson (2012b) propose a KDE density-based map construction method utilizing a "density estimation, map generation, trace map matching, topology refinement, geometry refinement" pipeline. The method is robust to noise and disparity problems in trajectories. Kuntzsch et al. (2016) combine a raster-based method to estimate the road segments with a vector-based optimization that estimates type and location of the junctions linking the road segments. This combines the generalization effect of KDE with the topological information from the original trajectories.

Common to all KDE methods, space is subdivided by means of a grid, which affects the resolution of the final map. The point cluster method does not rely on grids. Edelkamp & Schrödl (2003), Worrall & Nebot (2007) and Bhattacharjee et al. (2019) propose automated map construction methods through trajectory compression by clustering the trajectory points. Road networks are generated by linking the cluster centers. Chen et al. (2016) utilize a junction detection and recognition process with some pre-defined node structures. Li et al. (2016) analyze the linear nature of the trajectory points and propose a spatial linear cluster method to construct the road edges in linear directions, which can be well applicable for straight road edges. Ge et al. (2011) and Aanjaneya et al. (2012) use a Reeb graph to generate a network from the graph model with topological structures. A challenge with these methods is that nearby clusters might be miss-connected due to not fully utilizing the movement information available through trajectories. He et al. (2018) approach the map construction process as an image signal reconstruction problem and extract the road segment geometries using a Hough transformation. Huang et al. (2018) propose a method to generate the road segments by means of structure learning, similar to He & Bai et al. (2018). Ni et al. (2018) compress the trajectories by extracting representative points using Delaunay Triangulation and interpret the road segments as short triangle edges to construct a network. However, the method is still a density-based approach, as the length of the edges in the Delaunay Triangulation is affected by the density of the position samples. Overall, without considering the complex connection relationship between position samples, the density-based methods are usually fast in terms of run time and are robust with respect to noise by using a proper density threshold. However, movement information is usually eliminated in these approaches. Another drawback is that they typically need a lot of data to calculate meaningful networks.

Movement-based map construction methods usually construct road networks incrementally (Bruntrup et al., 2005; Zhang, et al., 2010; Zhang et al., 2017; He & Bastani et al., 2018) by combining sets of trajectories. Prior to map construction, map matching methods identify similarity portions between of trajectories and known/constructed road networks (cf. Brakatsoulas et al., 2005; Wenk et al., 2006). In the past, trajectories were used to complement existing road networks (Torre et al., 2012; Kasemsuppakorn & Karimi, 2013). Bruntrup et al. (2005) proposed an incremental map construction method to identify new network edges based on trajectories. The uncertainty in relation to trajectories does not always make it easy to identify the disparities between the road networks and gaps identified by trajectories. Towards a robust map construction approach, Cao & Krumm (2009) use a trace clarification process prior to an incremental graph construction step by simulating trace attractions to nearby trajectories. Ahmed & Wenk (2012) propose a partial matching-based method that uses a free space diagram to match trajectories to an incrementally grown road network. The Fréchet distance is used to locate similar (close) and dissimilar (far) portions of trajectories to the road network. Here, the network initially only consists of a trajectory. The impact of uncertainty is reduced to some extent by the free space diagram. Zhang et al. (2017) and He & Bastani et al. (2018) propose a similar method that incrementally merges single trajectories with a growing graph. A problem with all these approaches is the identification of intersections, given that they are not always properly captured as turns in a sampling-based trajectory recording approach. In addition, these methods are susceptible to noise in the trajectory data. Here, Fathi & Krumm (2010) proposed a road intersection detection method based on machine learning and a shape descriptor. A road network is constructed by connecting the detected intersections/nodes using trajectories. The generated road network geometry depends on the quality and quantity of available trajectory data. For tackling this challenge, a number of approaches exist. Xie et al. (2015) and Mariescu-Istodor et al. (2018) proposed a map construction method that uses trajectory segmentation in relation to the constructed intersections and merges the segments to construct a network. Karagiorgou & Pfoser (2012) and Karagiorgou et al. (2013 & 2017) propose the so-called trace bundle algorithm that just does that, i.e., merges/bundles trajectory segments between constructed intersection nodes to generate edges of the road network. To construct intersection nodes, position samples related to changes in movement are identified in trajectories based on turn types.

The various approaches differ by how much movement-derived information (e.g., speed, direction, road category, and vehicle type) is considered and how. This work presents a novel map-construction algorithm that considers the characteristics of recorded movements such as the sampling rate and the position samples in related to turns as captured in the trajectory data. Our approach belongs to the family of movement-based map construction methods, i.e., methods that aggregate polylines representing trajectories when reconstructing the road network. We will compare our results to various known map construction methods and on different open datasets.

3. Methodology

The proposed Movement-Aware Map Construction (MAMC) algorithm infers a road network graph from a set of user-generated trajectories and preserves movement features such as turn restrictions and edge direction using the following three steps: (i) node generation using an adaptive clustering approach that considers the position samples of turn points in relation to the "true" node locations, (ii) trajectory segmentation based on an adaptive attraction of trajectories to the nodes themselves, and (iii) segment conflation to construct the road network edges.

These three MAMC steps are described in detail in the following sections.

3.1 Motivation

Trajectories are the time-sequence (denoted as in Eq. 1) ordered movement records comprising position samples $p_i\left(x, y, t\right)$, with each sample consisting of a *x*, *y* coordinate pair and a time stamp. Such trajectories record the vehicular movement of users typically constrained by a movement network such as a road.

$$T:\{p_1,p_2,\cdots,p_n\} \tag{1}$$

Users move along a road and turn at intersections. Trajectories can thus be represented by a sequence of segments and turn points (cf. Eq. 2). Edges and nodes of the road network can be extracted from trajectories as shown in Eq. 3.

$$T:\{s_1,p_1,s_2,p_2,\cdots,s_m,p_m\} \tag{2}$$

$$R:\begin{cases} N \Leftarrow \{p_1,p_2,\cdots,p_m\} \\ E \Leftarrow \{s_1,s_2,\cdots,s_m\} \end{cases} \tag{3}$$

However, as the road network is usually oversampled, i.e., the same road is sampled by different users carrying different types of smartphones and apps that use different sampling methods, map construction becomes a challenging task. This article proposes a *movement-aware map construction algorithm* as shown in Fig. 1.

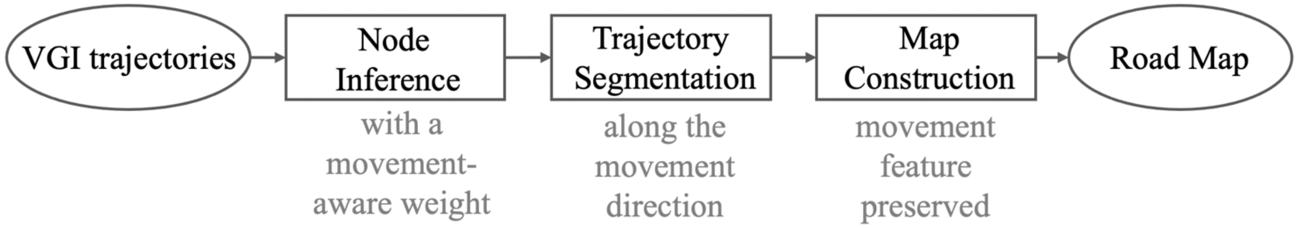

VGI trajectories
Node Inference
with a movement-aware weight
Trajectory Segmentation
along the movement direction
Map Construction
movement feature preserved
Road Map

Fig. 1: The movement-aware map construction framework

A road network is constructed from trajectories in three steps. First, we detect the nodes of the potential road network graph. This is done by using a weight-based clustering of turn points. Turn points that are deemed to be closer to the actual node are given a higher weight. As we will see in the next section, the weight is derived from the knowledge we have about the sampling process and the resulting trajectory geometry. Second, the constructed nodes are then connected using trajectories. Here, nodes attract trajectories based on their cluster weights, i.e., nodes that are the aggregate of many turn points attract trajectory segments that can be further away than nodes comprising fewer turn points. Third, the trajectories between nodes are aggregated using a so-called segment conflation approach.

3.2 Node detection

To infer the nodes of a road network, we use an adaptive approach that (i) detects turn points in trajectory data and (ii) aggregates turn points to nodes by considering the geometry of the sampled trajectories around the turn points.

3.2.1 Turn point detection

As a trajectory represents user movement, a change in direction is reflected as a heading change captured by the trajectory. We can quantify this change as the difference in heading towards and from a suspected turn point.

$$\alpha_i^{\leftarrow}=\begin{cases} \measuredangle(p_1,p_2) & i=1 \\ \measuredangle(p_{i-1},p_i) & i=2,3,\cdots,n \end{cases} \tag{4}$$

$$\alpha_i^{\rightarrow} = \begin{cases} \measuredangle\left(p_i, p_{i+1}\right) & i = 1, 2, \cdots, n-1 \\ \measuredangle\left(p_{n-1}, p_n\right) & i = n \end{cases} \tag{5}$$

Eq. 4 captures the heading towards a potential turn point p_i with respect to the previous point p_{i-1} of the trajectory, and Eq. 5 captures the heading with respect to the next point p_{i+1} of the trajectory. Based on the definitions of Eq. 4 and 5, the first and the last point of a trajectory will have the same heading from and to the point, i.e., for the first point $\alpha_1^{\leftarrow} = \alpha_1^{\rightarrow}$, which means that they cannot be a turn point. The function $\measuredangle$ is used to compute the angle of a vector between two trajectory points with respect to a reference direction (e.g., North). Thus, the turn angle of position samples can be computed based on the change in heading as Eq. 6.

$$\alpha_i = \alpha_i^{\rightarrow} - \alpha_i^{\leftarrow} \tag{6}$$

Based on turn angles, we can use a threshold, e.g., 45°, for identifying turn points.

3.2.2 Node inference

Given the inherent uncertainty of VGI, it is hard to infer a road network node from a single turn point. However, also thanks to VGI, we typically have many "observations" for each node, i.e., users moving through the same intersection. Since we use GPS-based sampling, the associated error for all position samples follows a normal distribution. Assuming normal driving behavior, the true location of all position samples is on the centerlines of roads and lanes (Guo, Iwamura, & Koga, 2007). Given reduced speeds and constant sampling rates, a considerable number of position samples will be recorded around intersections. We refer to those samples as turn points. We can infer node locations by aggregating turn points. We propose a new node inference method based on *weighted turn point clustering*. Turn points that are deemed to be closer to the eventual node are given a higher weight. The weight is derived from the knowledge we have about the sampling process and the resulting trajectory geometry.

(1) *Mean shift clustering*

Since the spatial network coverage of a trajectory is usually unknown, the number of expected nodes as a result of the map construction process is also not known. In a similar fashion, Karagiorgou & Pfoser (2012) merely use spatial proximity to cluster turn points and do not consider the movement characteristics in relation to the intersection. We use the mean shift clustering method (Fukunaga & Hostetler (1975); Cheng (1995)) to derive intersection nodes from turn points. It is a non-parametric point clustering method based on density gradient ascent, and the input parameters are the detected turn points and the distance for the cluster search. The cluster center C_i is shifted to higher density locations with each iteration according to the shift vector $M(C_i)$ (cf. Eq. 7).

$$M\left(C_i\right) = \frac{\sum_{j=1}^{n} p_j g\left(\frac{\left\|p_j - C_i\right\|}{d}\right)}{\sum_{j=1}^{n} g\left(\frac{\left\|p_j - C_i\right\|}{d}\right)} - C_i \tag{7}$$

$$g\left(x\right)=e^{-\frac{\|x\|^2}{2\sigma^2}} \tag{8}$$

In Eq. 7, d is the *search distance* for neighboring turn points p_j around cluster center C_i in the i-th iteration, and $g\left(\bullet\right)$ is the kernel function for the density distribution around C_i. There are many kinds of kernel functions, and in this work, we use the most commonly used Gaussian kernel function (denoted as in Eq. 8). In Eq. 8, with $\sigma = d$, the latter is the only input parameter and the method will compute cluster centers for turn points until $M(C_i)$ converges. The resulting centers are nodes in the road network graph that typically represent intersections. The choice of a parameter d is a non-trivial decision, since (i) if d is too small, clustering will produce too many cluster centers around the very same intersection and (ii) if d is too large, few cluster centers will represent a greater number of intersections.

In general, d is chosen based on expected road network properties. The heuristic we employed is based on the expected density of nodes and lengths of edges of the road network to be constructed. A good choice for d was found to be half of the length of a typical road network edge, i.e., in our case this equates to a d between 50*m* and 100*m*.

(2) *Clustering and weights*

To compute the best possible clusters, we need to consider the particularities of trajectory sampling, especially when it comes to turn points. Consider the example of Fig. 2, which shows several trajectories turning at node *N*. We see that very few turn points are actually located close to *N*. Thus, we have to consider the specific data collection approach when clustering turn points.

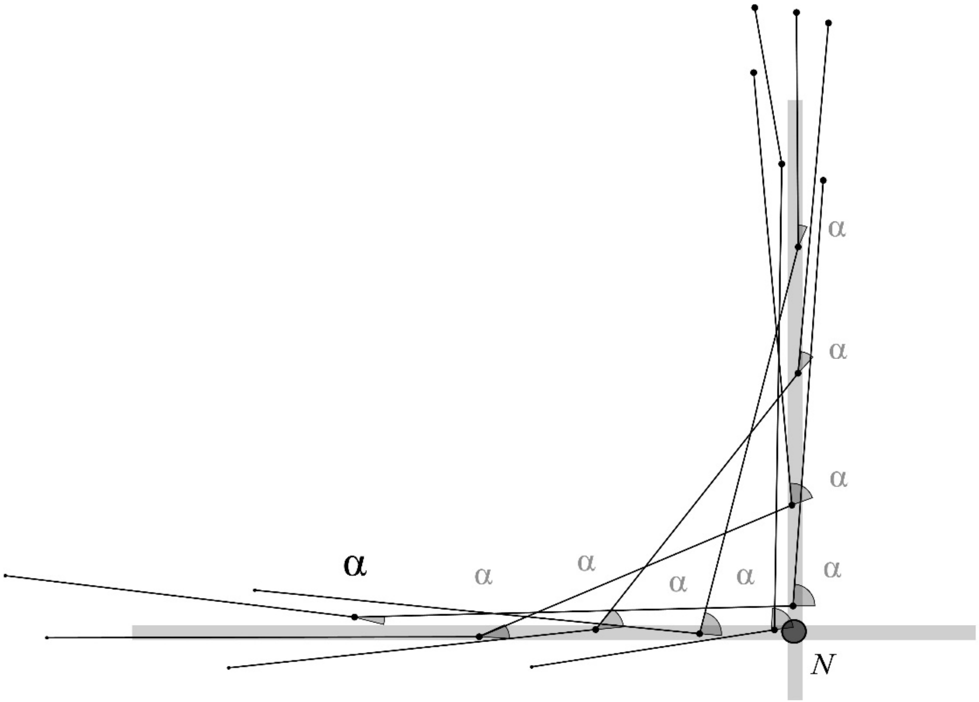


Fig. 2: Possible turn points and angles for one intersection node *N*

We observed in our ground-truth road network datasets that a typical turn angle at intersections is 90°. As such turn points with angles close to 90° are considered to be of higher quality and more credible than others. Assuming a fixed sampling rate for trajectories such as 30*s*, there is a relationship between the distance of the turn point to the intersection node and the *measured turn angle*. Further away turn points have more oblique angles. A turn point with a measured angle of 90° is expected to be closest to the node. To cluster the turn points, we thus weigh them based on the obliqueness of their angles.

(3) *Node inference*

We now introduce weighted turn points to our mean-shift clustering method. Step 1 categorizes the turn points based on their turn angles and assign different weights to each category, i.e., the closer the angle is to 90°, the higher will be the weight. For example, for $\alpha \in [45°, 60°)$, $\alpha \in [60°, 80°)$, and $\alpha \in [80°, 100°]$ we can assign the weights 1, 2, and 3, respectively. In Step 2, we choose a search distance *d* with respect to the targeted type of road network and randomly pick a turn point as the seed point and start searching the nearby turn points within distance *d*. We compute the kernel densities at each of the turn points and *duplicate them according to the assigned weight*. For example, a turn point with weight = 3 will be considered a total of three times during clustering. The cluster centers are refined until their location converges. Step 3 selects a new seed point that has not been included in any other cluster yet and Step 2 is repeated. The algorithm proceeds until all turn points have been included in clusters, i.e., are represented by a cluster of which its center will become an intersection node. During this node inference process, we preserve movement direction and the respective turn information for each inferred node *N*.

An example of clustering turn points and inferred nodes is given in Fig. 3. In this figure, turn points with different turn angles are classified into three categories and a different weight is assigned to each. The colors ranging from light to dark brown visualize turn points with respective weights. The figure also shows two alternative cluster centers, one computed with and one without considering weights. The weighted cluster center is considerably closer to the ground-truth intersection node location.

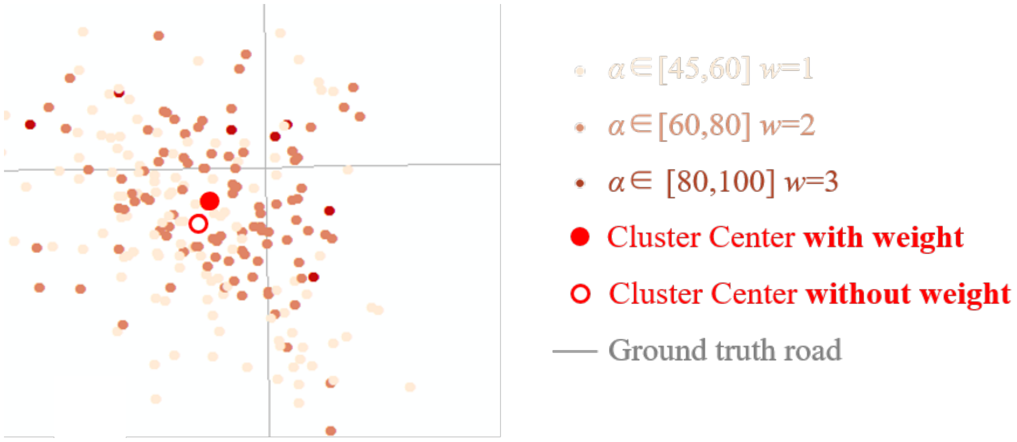


Fig. 3: Clustering turn points - intersection nodes with and without weight consideration

3.3 Trajectory segmentation based on adaptive node attraction

Nodes are inferred from turn points, which are position samples of trajectories. We can now use the trajectories to connect the nodes, i.e., to derive edges in our road network graph that we want to construct. We achieve this by using, in the simplest case, trajectory segments related to turn points, but also newly discovered segments based on so-called split points.

(1) *Turn-point segment aggregation*

Each turn point is subsumed by a node. Thus, turn points can be connected to the node. Consider the example of Fig. 4, which shows a trajectory T_1 with 4 turn points (P_1 to P_4) that are included in the clusters that generated nodes N_2, N_3, N_4, and N_5. We want to assign trajectory segments to node pairs (N_n, N_m). In our example, the three trajectory segments T_1^{1-2}, T_1^{2-3}, and T_I^{3-4} are linked to node pairs (N_2, N_3), (N_3, N_4), and (N_4, N_5), respectively. Also, as a convention, for a trajectory segment to be assigned to a node pair, both of its constituting (turn) points need to be part of a node cluster. Segments are otherwise discarded such as is the case for $T_1^{?-1}$ and $T_I^{4-?}$ in the example of Fig. 4.

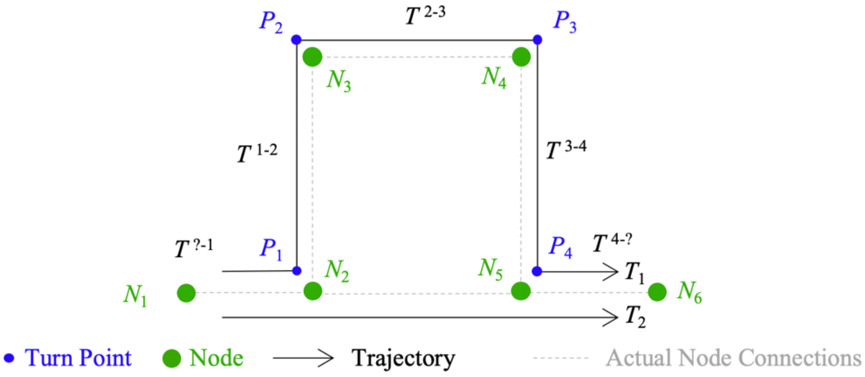


Fig. 4: Turn point attraction

(2) *Split-point segment aggregation*

Following the previous step, not all trajectory segments have been assigned properly to nodes as the *movement generating the trajectory might not include turns*. In those cases, a trajectory "passes" by a node, e.g., passes through an intersection. An example here is trajectory T_2 in Fig. 4. The question now is how we can match such a trajectory to nodes and still include the trajectory segments as part of our constructed map. We propose a gravity-based method that "attracts" trajectories to close-by nodes.

This method uses the number of turn points each constructed node comprises. Based on this count we construct a circle around each node that exudes the importance or *gravity* of this node. The larger a node's gravity and thus its radius, the more it "attracts" trajectories around it. An example range for the radius of a gravity circle is [$0.5d$, $3d$], where d is the search distance used to perform mean shift node clustering.

Finally, we find all trajectory segments that intersect with gravity circles of nodes. For each segment, we compute a split point at the closest distance of the segment to the node to split the trajectory segment into two smaller parts. Fig. 5 gives an example with a gravity circle around a node attracting various trajectory segments that intersect it. Split points are created for each of the intersecting trajectories (empty circles).

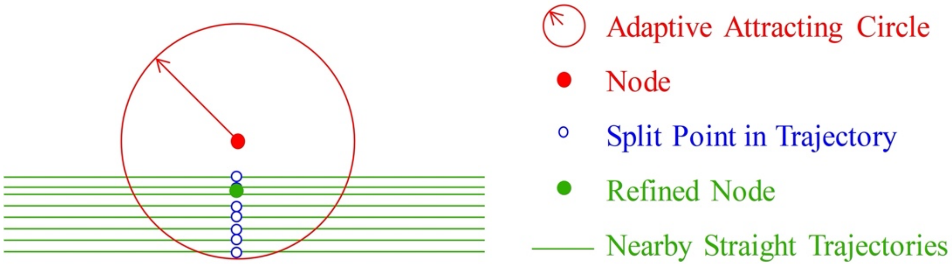

Adaptive Attracting Circle
Node
Split Point in Trajectory
Refined Node
Nearby Straight Trajectories

Fig. 5: Split point attraction

A split point will adjust the location of the respective node it is attracted to. Suppose n turn points were used to compute the location of node N and in addition it attracts m split points, then a new cluster center $N^{'}$ can be derived as a weighted mean shift cluster as shown in Eq. 9.

$$N^{'} = wN + w^{*}N^{*} \tag{9}$$

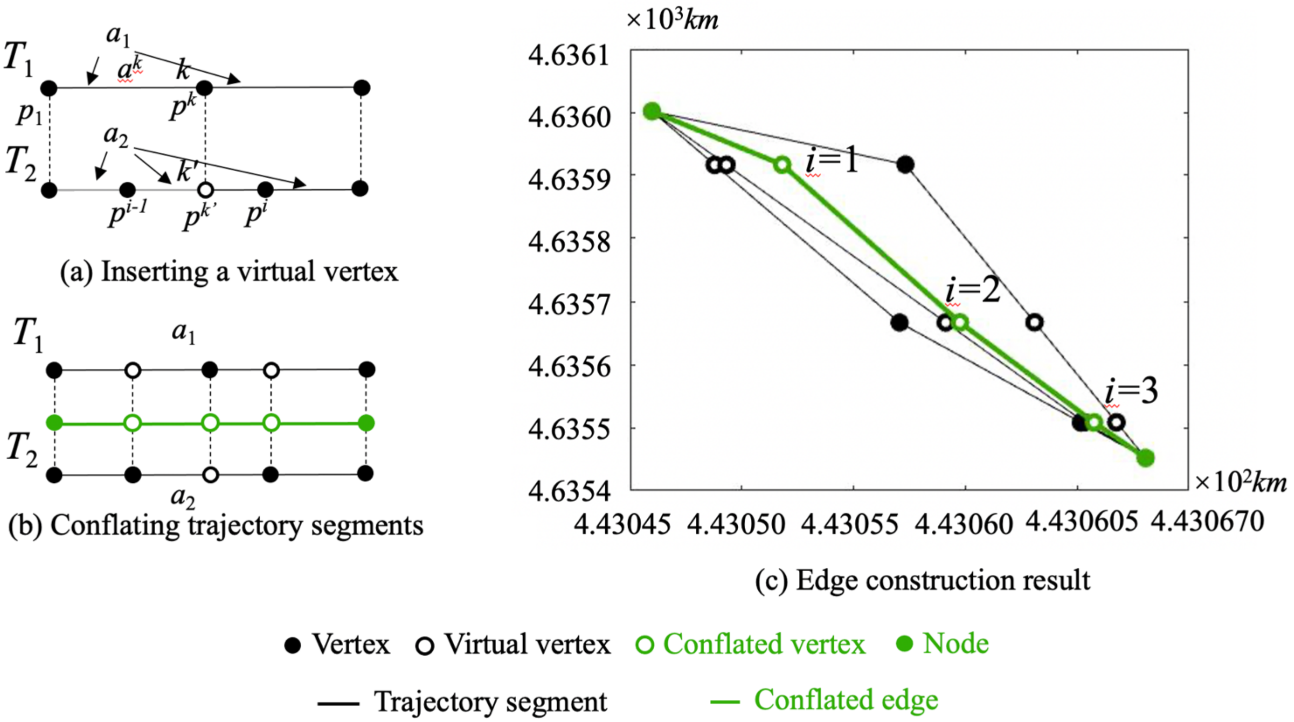
Here, N and N^{*} represent the weighted mean shift clusters of the original turn points and the newly discovered split points, respectively. The weights are defined as $w = \frac{n}{m+n}$ and $w^{*} = \frac{m}{m+n}$.

3.4 Movement-aware segment conflation

Having discovered all relevant trajectory segments in between intersection nodes, the edges of a road network can be constructed by conflating these segments.

Each edge is represented by a set of segments. We propose a movement-aware segment conflation method based on the so-called “Arc Edit” approach (Lyu et al., 2017). The trajectory segments are conflated by inserting virtual vertices to preserve the movement features in the original trajectory as depicted in Fig. 6. Calculating so-called Accumulated Length Percentages (ALP), makes trajectory segments comparable and will allow us to merge them. Intuitively, the ALP for a given segment provides a one-dimensional distance signature of its vertices with respect to the starting vertex. Having such signatures for a set of segments, we decide on whether virtual vertices need to be inserted to make them comparable.

This method comprises the following three steps to make two trajectory segments comparable and to finally merge/conflate them.



(a) Inserting a virtual vertex

(b) Conflating trajectory segments

(c) Edge construction result

Fig. 6: Trajectory segment conflation based on Arc Edit

(1) *Accumulated Length Percentage* (ALP)

Suppose there is a trajectory $T:\{p_1,p_2,\cdots,p_n\}$ and a sub-trajectory composed of position samples denoted as $T^{(1,k)}:\{p_1,p_2,\cdots,p_k\}$ (such as the segment in Fig. 6(a)), the ALP a^k of $T^{(1,k)}$ is calculated as follows:

$$a^k=\frac{\sum_{i=1}^{k-1}\mathrm{L}(p_i,p_{i+1})}{\sum_{i=1}^{n-1}\mathrm{L}(p_i,p_{i+1})} \tag{10}$$

$\mathrm{L}(\bullet)$ calculates the Euclidean distance between two points. For *T*, the ALP of all of the sub-trajectories can be denoted as:

$$a\ \{a^1,a^2,\cdots,a^n\} \tag{11}$$

Consider the example of Fig. 6(a), which shows a trajectory T_1 that is composed of 3 vertices and a trajectory T_2 that has 4 vertices. ALPs for T_1 and T_2 are computed as a_1 and a_2, respectively.

(2) *Movement-aware vertex insertion*

Suppose there are two trajectory segments T_1 and T_2 with *m* and *n* vertices and ALPs a_1 and a_2, respectively. The unified ALP *a* can be denoted as follows.

$$a\ \{a^1,a^2,\cdots,a^{m+n-2}\} \tag{12}$$

It integrates all individual values from a_1 and a_2 into one sorted sequence, i.e., in ascending order with respect to the movement direction. Consider the example of Fig. 6(a), which shows an a^k (in a_1) with no corresponding value in a_2, i.e., its value is between a_2^{i-1} and a_2^i. Here we insert a virtual vertex $p_2^{k'}$ in T_2 between p_2^{i-1} and p_2^i following Eq. 13. The same applies to points in T_1.

$$p_2^{k'} = p_2^{i-1} + \frac{a^k - a_2^{i-1}}{a_2^i - a_2^{i-1}}\left(p_2^i - p_2^{i-1}\right) \tag{13}$$

(3*) Geometric feature preserving trajectory segment conflation*

Following the movement-aware vertex insertion, each of the two trajectory segments has the same number of vertices with comparable ALPs (temporal signatures) (cf. Fig. 6(b)). As such, each set of corresponding vertices can be conflated based on the average position of each vertex. This conflation process can be illustrated by using the example of Fig. 6(c) using 3 trajectory segments. Virtual vertices are added to the trajectory segments based on Step 2 – Movement-aware vertex insertion. The resulting edge (green) is constructed based on the conflation of vertices in each ALP group (index *i*). In addition, trajectories linking the same nodes, but having opposite movement directions are also conflated by reversing the movement direction of the segments. Here the direction information is stored with the constructed edges.

4. Experiments and discussion

The scope of our experimental evaluation is to assess the map construction result quality in comparison to other methods and by using a range of metrics and datasets. The specific datasets, evaluation methods and comparison methods used in the following experiments are discussed in more detail in Ahmed et al. (2015a &2015b) and were retrieved from http://mapconstruction.org.

4.1 Movement-aware map construction results and quality

This first experiment illustrates the steps of the MAMC algorithm and the data that is generated during each step. The experiment uses data collected in a part of Chicago, IL (cf. experimentation in Biagioni & Eriksson (2012a & 2012b)) and covers an area of about $30km^2$ and includes 889 trajectories generated by university shuttle buses as shown in Fig. 7. As this dataset is not too large and has been widely used in other works, it provides a fitting initial assessment of our method. What follows is a discussion of the results in relation to the steps of the MAMC algorithm.

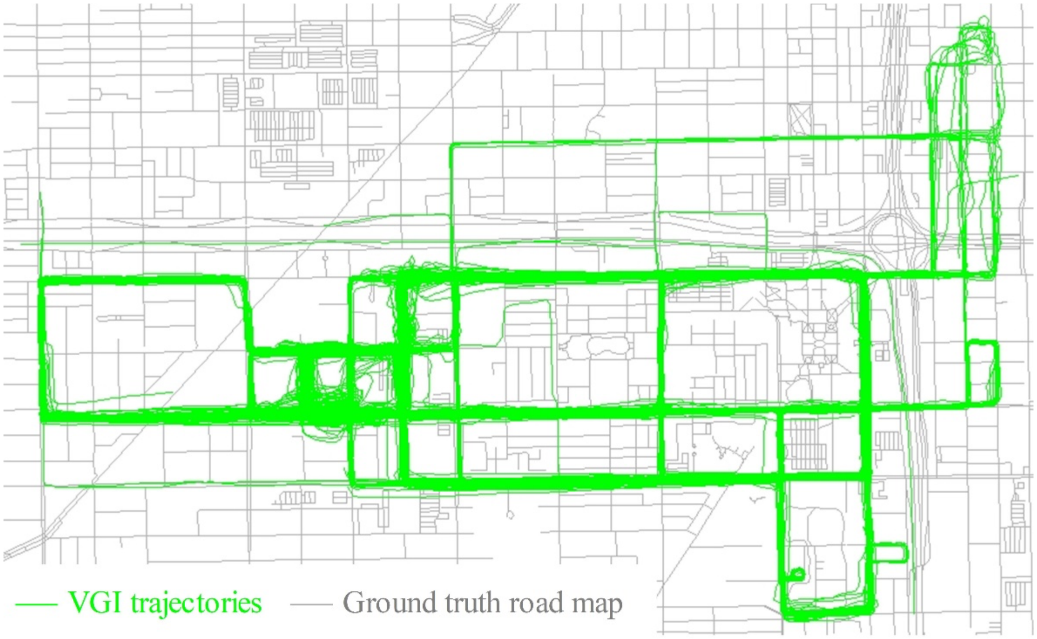


Fig. 7: The Chicago dataset

(1) Node inference

Nodes are derived by clustering turn points, which in turn are trajectory points that fulfill certain turn angle qualities. Only trajectory points with turn angle between 45° and 100° are considered. The movement-aware weights for the turn points are assigned as 1, 4, 16 for $\alpha \in [45°, 60°)$, $\alpha \in [60°, 80°)$, $\alpha \in [80°, 100°]$, respectively and the search distance *d* parameter of the mean shift cluster method is set to be 30*m*. This distance relates to the smallest distance between two parallel road segments. Those weight parameter settings were established following a brief experimentation with different weight values. Although the actual values are not that important, what matters is to choose weights that significantly differentiate between the angles and as such the turn point in question.

The node inference result is shown in Fig. 8. Turn points are categorized into 3 groups with different movement-aware weights. The higher the weight, the closer they are expected to be to the actual intersections. The derived cluster centers are shown as red solid circles and they are located near the actual road intersections with superimposed onto the actual road network. The experiments generated 118 cluster centers. However, we discarded cluster centers with a support of less than 5 turn points since we consider them unreliable. *Finally, 43 intersection nodes are inferred from the cluster centers of movement-aware weighted turn points.* The example shows that all identified nodes coincide with actual intersection locations.

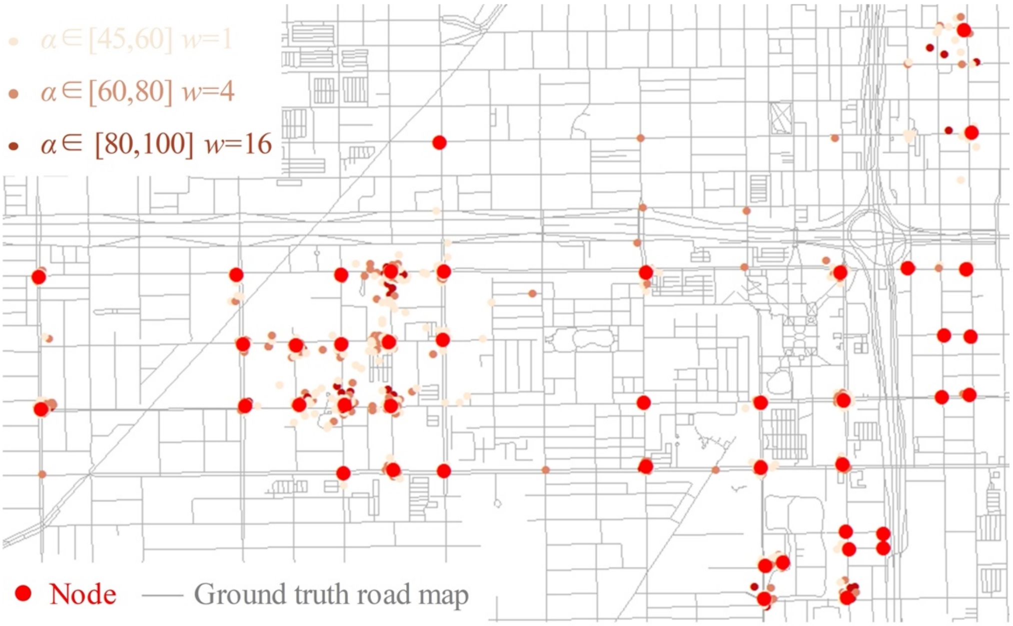


Fig. 8: Node inference with movement-weighted turn points

(2) Trajectory segmentation based on adaptive node attraction

For each inferred node, we record its constituting turn points. We segment the trajectories accordingly using the Adaptive Node Attraction method. In this experiment, the radius of the gravity circle ranges from 0.5*d* to 3*d*, i.e., 15 - 90*m* in our case. These parameters were established empirically and are dictated by the density of the road network. The node locations, their respective gravity circles, and the trajectory segmentation results are shown in Fig. 9. The gravity circles are categorized into five groups shown by the red solid circles, and the trajectories are segmented into 91 different groups, which are shown in different colors. Each segmented trajectory represents a direct connection between nodes. This connectivity information in terms of trajectories is usually not preserved in density-based map construction methods as will be discussed later in this work.

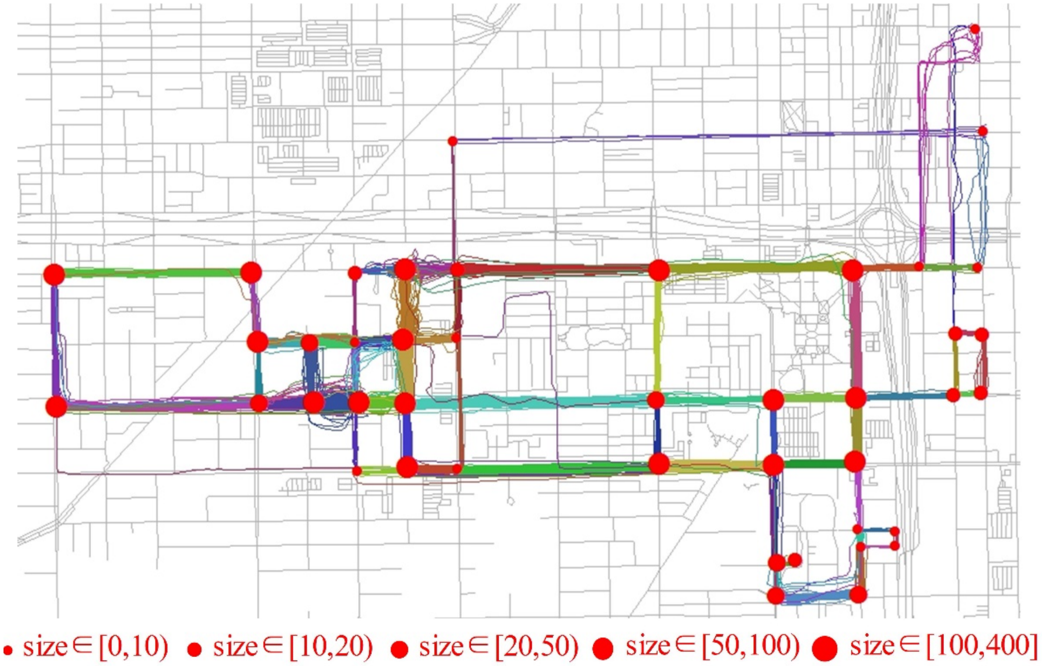
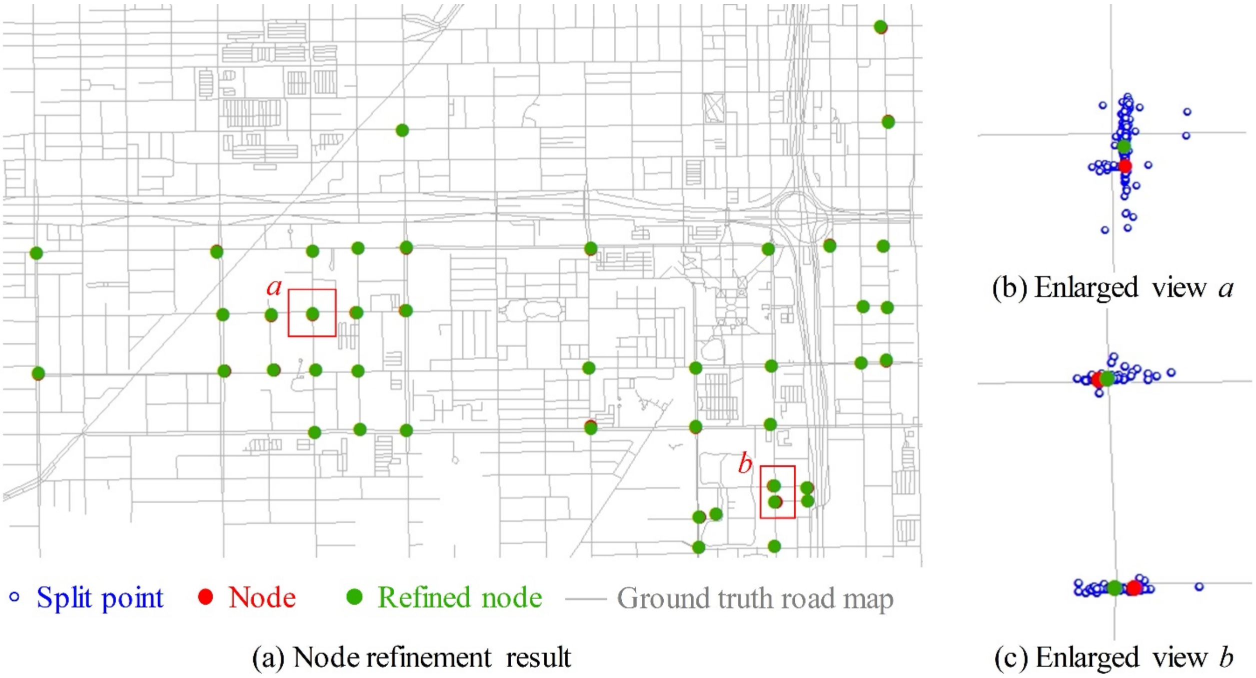


Fig. 9: Trajectory segmentation based on adaptive node attraction

Inferred nodes are further refined by split points derived from adjacent trajectory segments. Fig. 10 gives an example of how nodes (red solid circles) are adjusted (green solid circle) and are then closer to the intersection location.



(a) Node refinement result

(b) Enlarged view *a*

(c) Enlarged view *b*

Fig. 10: Node refinement based on split points

(3) Movement-Aware segment conflation

Based on the movement-aware segment conflation method proposed in Section 3.4, trajectory segments connecting the same node pair are conflated and replaced by the resulting (constructed road network) edge. Our experiment generated 88 viable edges. After eliminating some edges supported by only a few trajectories (< 5), the final generated road network is shown in Fig. 11.

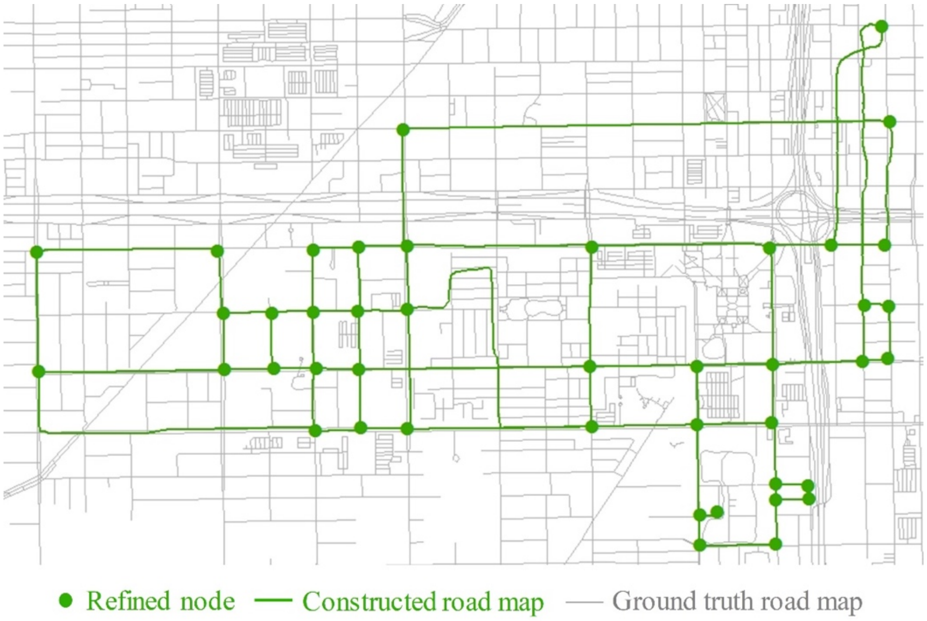


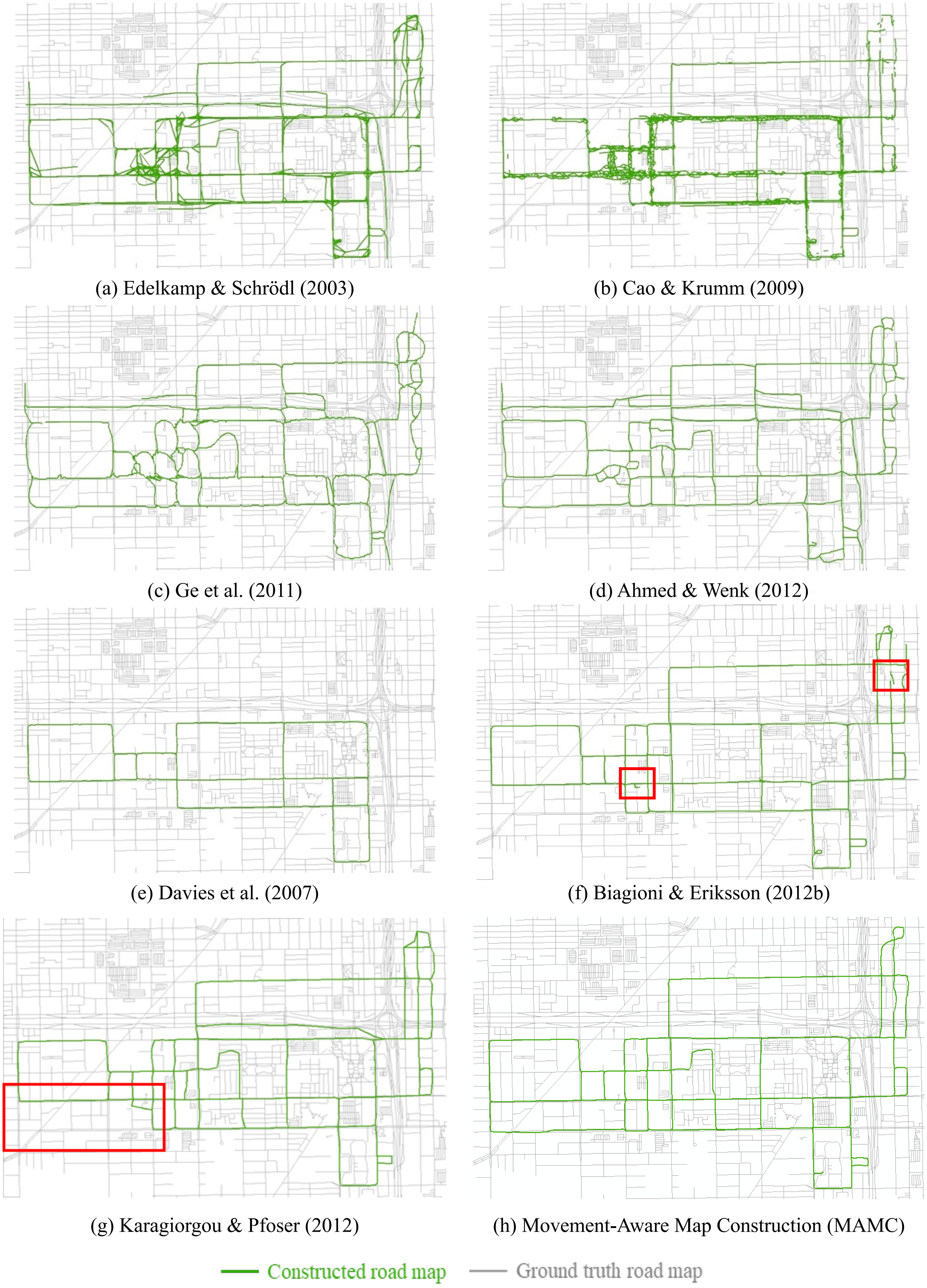
Fig. 11: Map construction based on movement-aware segment conflation

4.2 Assessment of the map construction method

To assess the performance of the MAMC method, we compare map construction results from different well-known methods, such as Edelkamp & Schrödl (2003), Cao & Krumm (2009), Ge et al. (2011), Ahmed & Wenk (2012), Davies et al. (2007), Biagioni & Eriksson (2012b), and Karagiorgou & Pfoser (2012). The methods are summarized in Ahmad et al. (2015a) and Ahmad et al. (2015b). The data and source code is available at http://mapconstruction.org. We re-ran all the experiments to re-generate the map-construction results for the various methods based on parameters provided as part of the source code. Using again the Chicago dataset, several map construction results are shown in Fig. 12.

(1) Visual comparison

The results in Fig. 12(a-b) deviate quite a bit from the ground truth road map as many additional edges are introduced. The approach in Fig. 12(a) relies on a *k*-means approach to cluster the input point set and as such overestimates the intersection nodes. Cao & Krumm (2009), relying on incremental track insertion, in Fig. 12(b) also overestimates the intersection nodes. The problem is mainly caused by data quality and trajectory outliers, which lead to an overestimated set of nodes. Figs. 12(c-d) show good results in terms of topology but inaccurate road network geometries. The result of Fig. 12(c) was generated by a method that tries to capture the topological structure using a *Reeb graph*, however the geometric shape of the road edge was not captured accurately due to the lack of data quality. In Fig. 12(d), the constructed map includes false nodes caused by outliers. However, the result has been significantly improved and the trajectories are better matched and conflated to edges using an incremental track insertion approach. Fig. 12(e) has limited coverage. The map construction results in Figs. 12(f-h) show the best coverage and accuracy in terms of topology and geometric similarity. Figs. 12(e-f) show accurate results with respect to geometry using a density-based sampling approach, which is more robust to outliers. On the other hand, areas of low density are not captured in the final result. The map construction results of Figs. 12(g) and (h) are comparable. The main difference is the accuracy of the intersection node locations. The nodes in Fig. 12(h) are generated based on a modified movement weight and direction approach. In connection with improved trajectory conflation, the map construction result seems to visually outperform that of Fig. 12(g).


(a) Edelkamp & Schrödl (2003)

(b) Cao & Krumm (2009)

(c) Ge et al. (2011)

(d) Ahmed & Wenk (2012)

(e) Davies et al. (2007)

(f) Biagioni & Eriksson (2012b)

(g) Karagiorgou & Pfoser (2012)

(h) Movement-Aware Map Construction (MAMC)

Fig. 12: Comparison with the results of other map construction methods

Tab. 1 provides an overview of the distances between the inferred and ground truth nodes for the results shown in Figs. 12(g) and (h) by reporting the Max, Min, Mean, and Standard Deviation values for the respective methods. The result of Fig. 12(h), MAMC, has a lower mean distance and standard deviation, resulting in an overall better constructed map.

Tab. 1: Distance between inferred and ground truth road intersection nodes [*m*]

Map construction result	Max	Min	Mean	Std Dev
Fig. 12(g)	25.3	0.1	8.2	6.1
Fig. 12(h)	20	0.8	**6.9**	**4.5**

(2) Local similarity measures

Ahmed et al. (2014) presented a distance measure that is based on comparing the local persistent homology of two graphs. The method generates discs (which is an open area *U*) at the sampled positions $x \in \mathbf{X}$ in each road network (R_1 and R_2) with an increasing radius *r* ($r \leq r_{\max}$), which are used to derive the persistence diagrams for different road networks. A local distance signature ψ_r based on the bottleneck distance is computed using a filtration process. The local homology distance metric d^{LH} is denoted as follows.

$$d^{LH}\left(R_1, R_2\right) = \int_0^{r_{\max}} w(r) \int_{\mathbf{X}} \eta(x) \psi_r(x)\, dx dr \tag{14}$$

In Eq. (14), $\eta : \mathbf{X} \rightarrow \mathbf{R}$ and $w : \left[0, r_{\max}\right] \rightarrow \mathbf{R}$ are non-negative weight functions. Finally, the variant d_{var}^{LH} of the local homology metric is computed for the union of the road networks $\mathbf{Y} = R_1 \cup R_2$, with $\left|\mathbf{Y}\right|$ being the total length of the edges in $\mathbf{Y}$, for the area of interest as follows.

$$d_{\text{var}}^{LH}\left(R_1, R_2\right) = \int_0^{r_{\max}} \frac{w(r)}{\left|\mathbf{Y}\right|} \int \psi_r(x)\, dx dr \tag{15}$$

Essentially, this distance measure allows us *to compare spatial graphs in terms of topology and geometric similarity*. We use this measure to construct distance signatures of a constructed network to its ground truth. What will be of interest is not the absolute similarity, but comparing the outputs of the various algorithms to each other.

Here, we only evaluate the three best results shown in Fig. 12(f-h). The similarity of two graphs is calculated using a set of random seed points (locations) distributed over the spatial area covered by the graphs. The graphs are compared in circular areas of a specific radius centered on the seeds (cf. Ahmed et al. (2014)). Each circle can be considered as the local scope of comparison for each graph. The local homology measures then compare the two graphs at each location in terms of their geometric similarity and topology.

In our experimentation, 11424 circles with a radius of 30*m* are generated to cover the trajectories and constructed road network of the Chicago area. To compare the different results, we consider for each network *the number of seed points that have small dissimilarity measures*. We chose a value of < 3000 as a threshold, which is not too strict and preserves a considerable number of locations to compare the various map-construction results. This threshold is an algorithm parameter and has no metric equivalent. A larger number of such points indicates an increased similarity between the constructed and the ground truth network.

*For the three methods of Biagioni & Eriksson (2012b), Karagiorgou & Pfoser (2012), and **MAMC** (this work), the respective number of seed points with such a low score are 405, 436, **459**, respectively. The average dissimilarity* *measures for the three algorithms are 4.98, 5.04, and* ***4.95****, respectively. Both measures give an advantage to MAMC as producing the network with the highest number of seed points with small dissimilarity scores (459) and having the smallest dissimilarity measure.*

A visual representation of the dissimilarity scores superimposed on the constructed map (MAMC) is shown in Fig. 13. Lighter colors show re-constructed road network segments that are similar to the ground truth map. Darker colors indicate dissimilarity in terms of topology and geometry. Overall, areas around intersections are typically shown in darker colors, given that either the topology and/or the geometry (intersection node locations) differ from the ground truth. In examining two specific areas, we see that although road network geometries are similar, missing edges at an intersection contribute to higher measures (blue rectangle). The red rectangle shows a part of the road network that has a low dissimilarity score since the entire network was constructed.

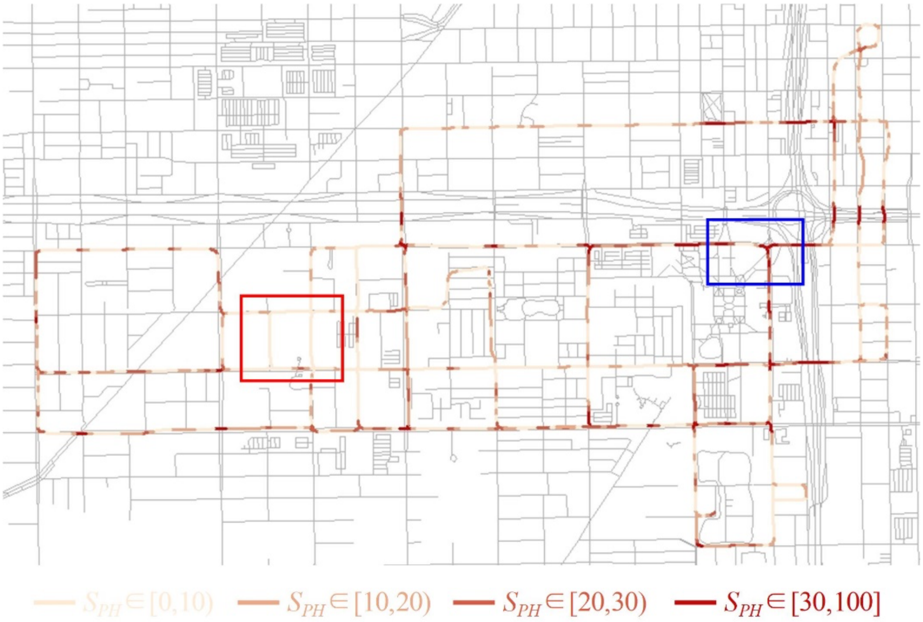


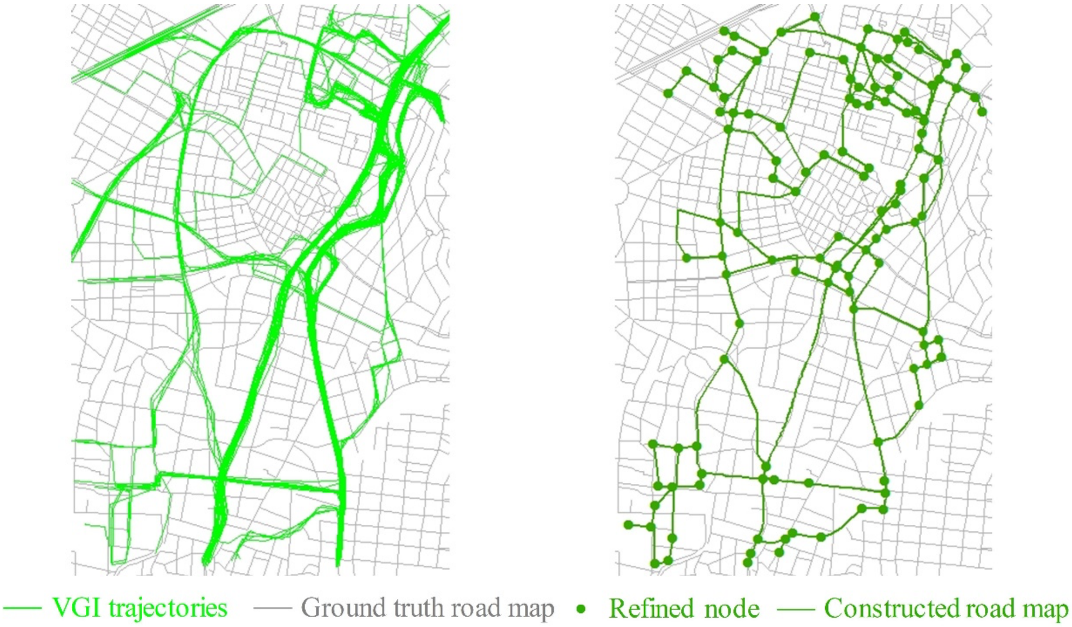
Fig. 13: MAMC result for Chicago – colors showing good (light color) and less good matches (darker color)

4.3 Movement-aware map construction using additional data sources

To validate the scalability and robustness of MAMC we include experiments for three additional datasets. The Athens (small) dataset shown in Fig. 14(a) has an extent of 15km^2 covered by 129 trajectories generated by school buses. The Athens (large) dataset shown in Fig. 15(a) has an area of 170km^2 and is traversed by 120 trajectories generated also by school buses. The Berlin dataset shown in Fig. 16(a) has an area of 36km^2 and is covered by 26831 trajectories generated by a taxi fleet. These three datasets represent a different type of road network at different spatial scales, complexities, and trajectory data sources when compared to the Chicago dataset.

(1) Athens (small) map construction result

The Athens (small) dataset is a bit smaller than the Chicago dataset, but the road network to be inferred is more complex. The road density is higher, which means that a map construction algorithm needs to distinguish between close by, possibly parallel roads.



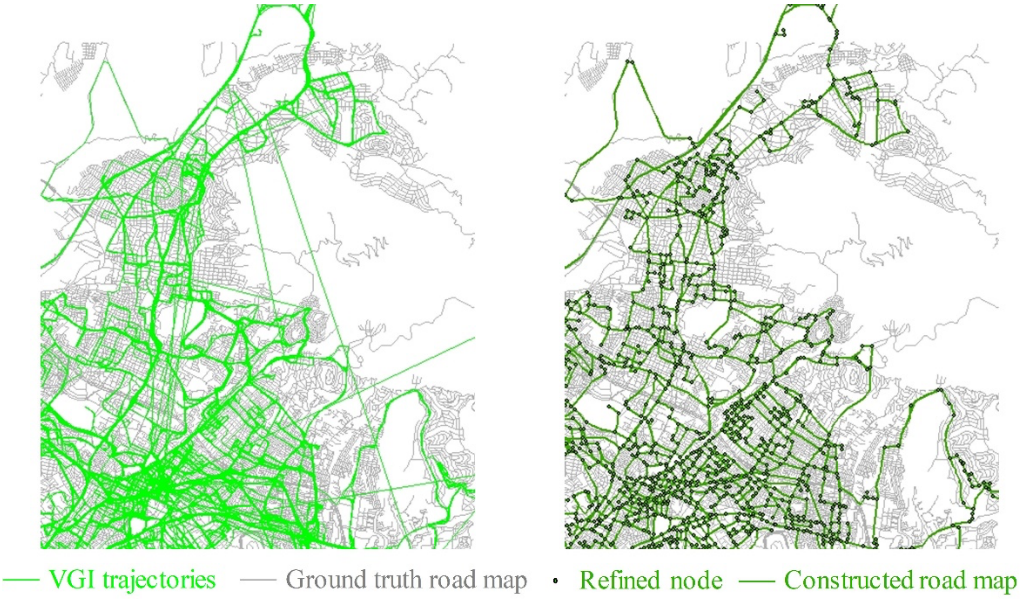
(a) Athens (small) dataset (b) Athens (small) map construction result

Fig. 14: Map construction result of Athens (small) dataset

The parameters of the MAMC algorithm are identical to the Chicago experiment, only the search distance d = 50m. Following turn point clustering, 122 cluster centers are generated, and 122 intersection nodes are inferred. Segment conflation generates 219 candidate edges, of which only 167 edges are preserved. The map construction result for the Athens (small) dataset is shown in Fig. 14(b). With 122 nodes connected by 167 edges, the reconstructed road network generates the portions of the road network covered by the trajectories very well.

(2) Map construction result of Athens (large) dataset

The Athens (large) dataset covers a larger area and a considerably more complex road network. The dataset consists of quite long trajectories, which have few turns, i.e., do not turn at intersections. Turns are essential for our algorithm and as such this dataset presents quite a challenge to our method.



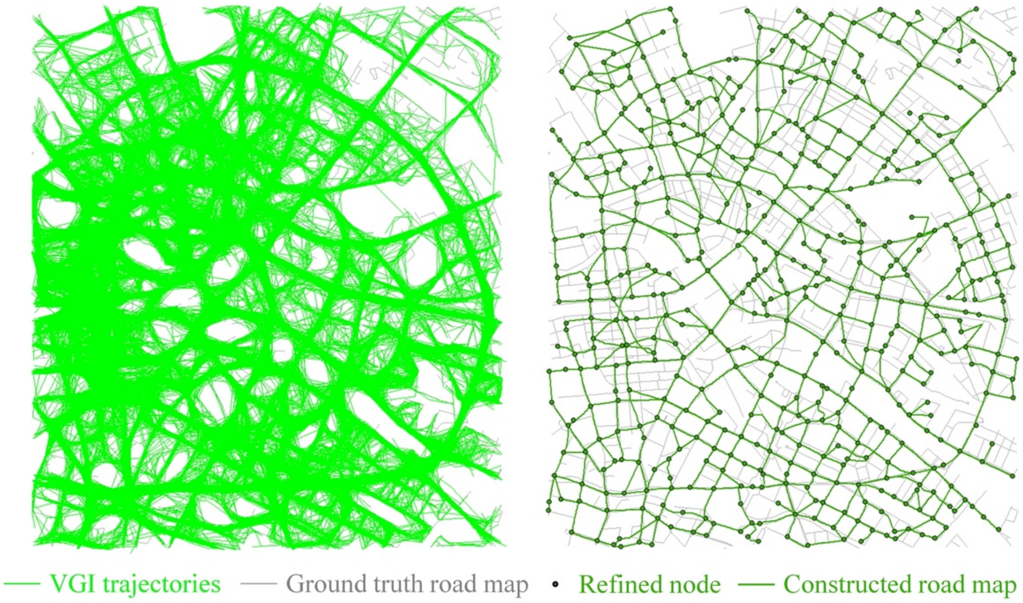
(a) Athens (large) dataset (b) Athens (large) map construction result

Fig. 15: Map construction result of Athens (large) dataset

Using a search distance $d = 70m$ and otherwise unchanged parameters, we have inferred 1039 nodes and 1461 edges. The map construction result for the Athens (large) dataset shown in Fig. 15(b) is impressive given its coverage and accuracy.

(3) Map construction result of Berlin dataset

The Berlin dataset covers an area similar to Athens, but exhibits a more nuanced road network in terms of mix of road categories. The dataset was collected with a high sampling rate (10-30*s*) and has considerable sampling errors. The data quality has a direct impact on the map construction result.



(a) Berlin dataset (b) Berlin map construction result

Fig. 16: Map construction result of Berlin dataset

Using a search distance $d = 70m$ and otherwise unchanged parameters, 517 nodes and 778 edges were created. The map construction result is shown in Fig. 16(b). While not all edges were found and identified correctly, the positional accuracy of the intersection nodes is very high.

(4) Assessment of Map construction results

The map construction results of the Athens (small), Athens (large) and Berlin datasets based on the 7 methods that are mentioned in Fig. 12 are consistent with the results presented in Ahmed et al. (2015b). http://mapconstruction.org/ provides visualizations of the respective results. Beyond visualizations, Tab. 2 provides the local similarity metrics introduced in Section 4.2 for all results. Unfortunately, due to the running time complexity of the algorithm implementing this measure, some similarity metrics could not be calculated.

Tab. 2: Comparison of different map construction results based on local similarity measures ("-" means no result could be calculated)

Dataset	Athens (small)	Athens (large)	Berlin
Edelkamp & Schrödl (2003)	-	-	-
Cao & Krumm (2009)	15.76	-	-
Ge et al. (2011)	13.02	-	-
Ahmed & Wenk (2012)	10.78	-	9.78
Davies et al. (2007)	15.56	-	-
Biagioni & Eriksson (2012b)	12.15	-	-
Karagiorgou & Pfoser (2012)	9.94	-	8.89
MAMC	*9.86*	-	*8.86*

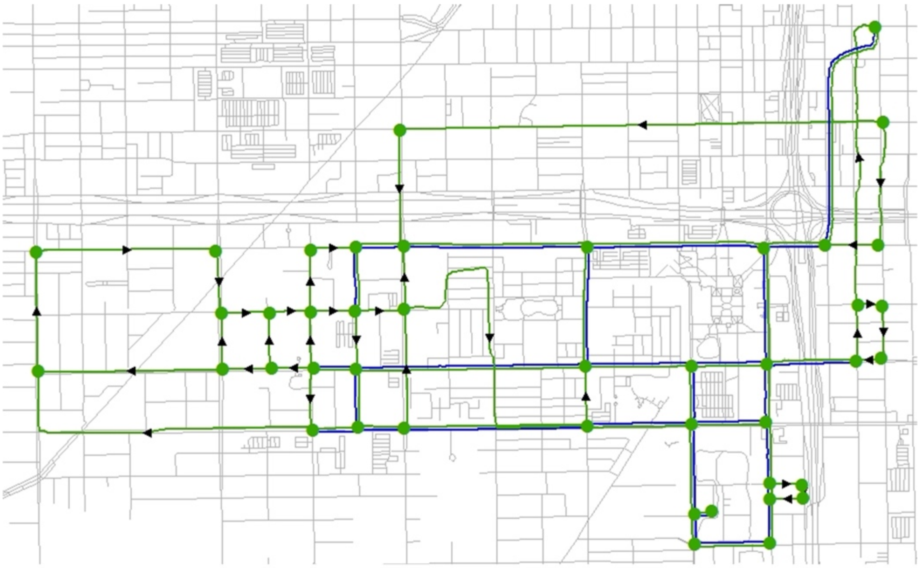
Tab. 2 shows that the average dissimilarity measures for MAMC is 9.86 for Athens (small) and 8.86 for Berlin. In both cases, this is the smallest measure for all algorithms and as such MAMC seems to produce the best overall map construction result.

4.4 Map construction and movement information

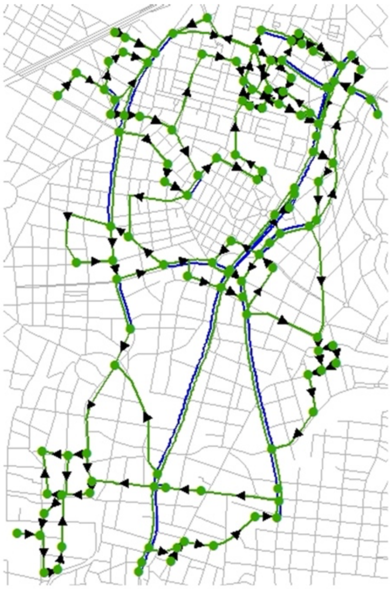
As the map construction method is movement-aware and movement features are utilized in the map construction process, the movement information, such as the edge direction and turn type at a node, is captured as part of the map construction result.

(1) Direction

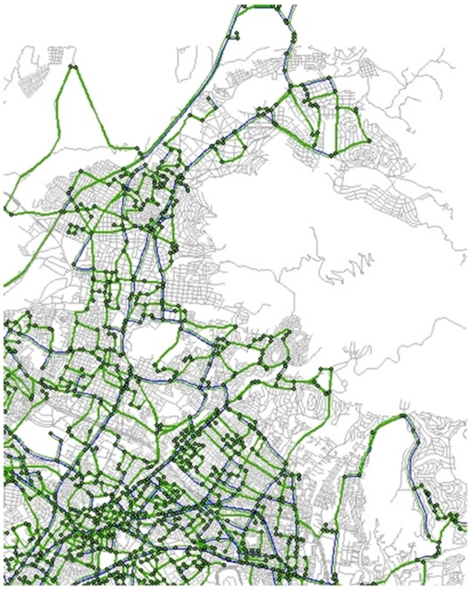
The edge direction is derived from the movement direction of the trajectories. Depending on whether there is uniformity or not with respect to the movement when conflating segments, we have either uni- or bi-directional movement. The edge directions for the various constructed maps are shown in Fig. 17.



(a) Edge direction of Chicago



(b) Edge direction of Athens (small)



(c) Edge direction of Athens (large)

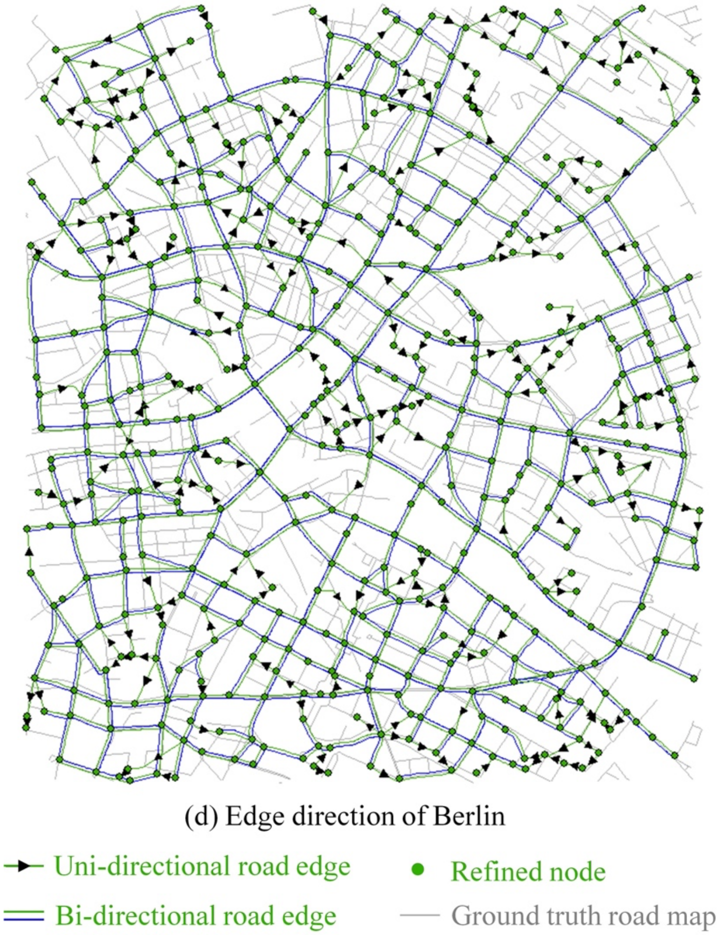
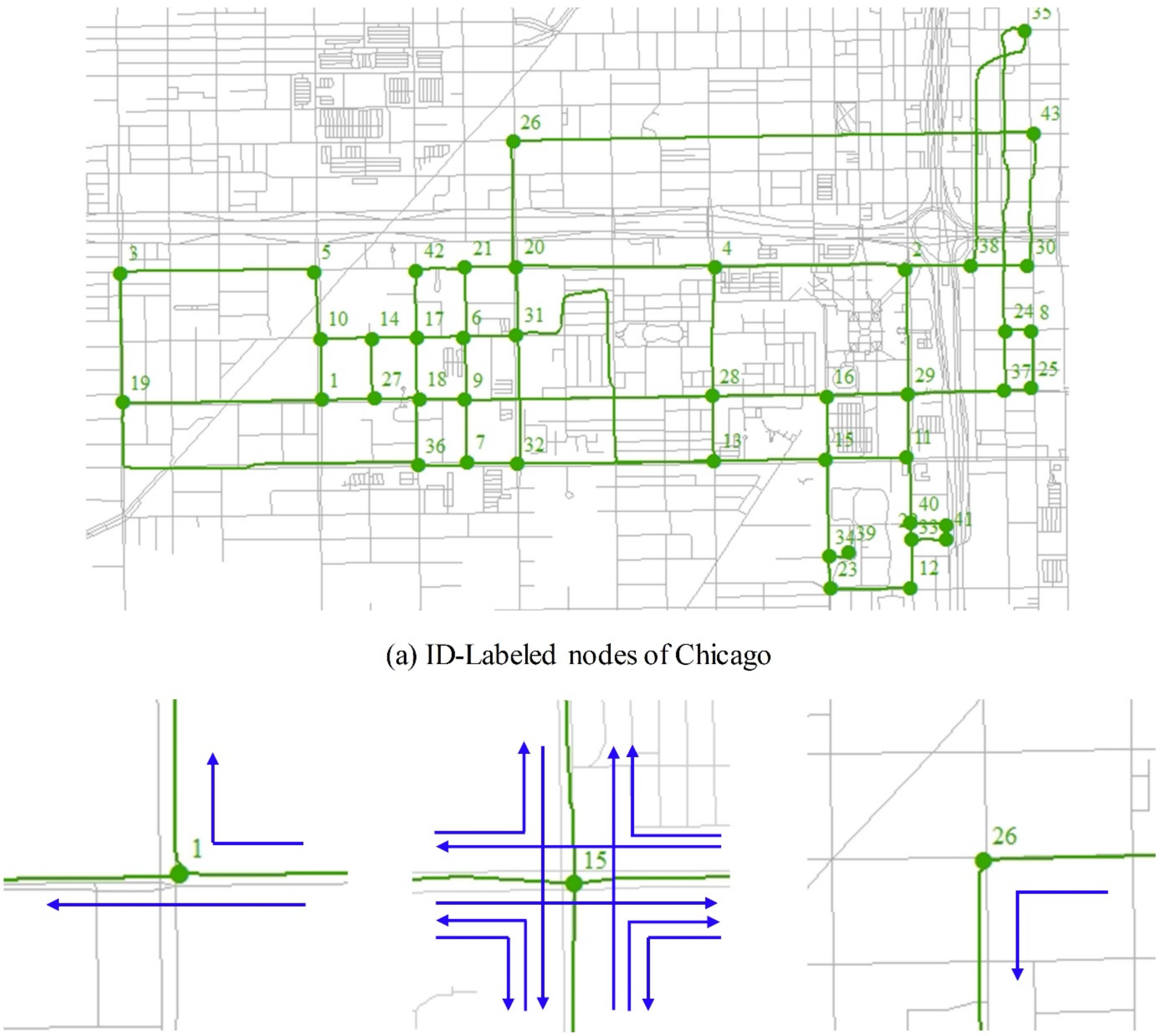

(d) Edge direction of Berlin

Fig. 17: Edge directions in the map construction result

Chicago has 34 uni-directional edges and 28 bi-directional edges, Athens (small) has 118 uni-directional edges and 49 bi-directional edges, Athens (large) has 862 uni-directional edges and 599 bi-directional edges, and Berlin has 200 uni-directional edges and 578 bi-directional edges.

(2) Turn types

Turn types of nodes (is it a 4-way intersection?) are again derived from trajectory information and the respective turns that were observed as part of the map construction process when deriving intersection nodes. Consider here the Chicago map construction result as shown in Fig. 18. The nodes are given unique ids in Fig. 18(a). Three random nodes and their observed turns are shown in Figs 18(b-d). For example, for Node 1, we only observed a right turn driving East. For Node 26 we only observed a left turn. Node 15 is a busy intersection and almost all turns were observed.



(a) ID-Labeled nodes of Chicago

(b) turn types at node 1

(c) turn types at node 15

(d) turn types at node 26

Fig. 18: Turn types of the nodes in the map construction result

5. Conclusions and future work

Location data is a big data by-product of our digital world saturated with mobile computing devices. Trajectory data as its spatiotemporal derivative is collected by a myriad of applications tracking people and goods. Trajectory data is leveraged in a range of contexts such as traffic management, fleet management and logistics services, and even urban planning and has resulted in a range of data mining challenges including map matching, i.e., relating trajectories to road networks, and in our case map construction. Here the goal is to identify the transportation infrastructure that has actually generated these trajectories. This work presents a novel map-construction algorithm that considers the characteristics of recorded movements such as the sampling rate and the position samples in relation to turns as captured in the trajectory data. Our approach belongs to the family of movement-based map construction methods, i.e., methods that aggregate polylines representing trajectories when reconstructing the road network. The method initially constructs intersections from turn points identified in the trajectory data by specifically taking into account the trajectory geometry around a turn point and thus implicitly how far before or after an intersection the location of the vehicle was sampled. This improves the quality of the intersection nodes, which are then further adjusted using adjacent trajectories as they pass by those intersection nodes. Nodes are constructed using a mean-shift cluster algorithm, a density-based clustering technique. We construct the road network edges by clustering trajectory segments connecting intersection nodes using the so-called arc editing method. An experimental evaluation using trajectory datasets from different geographic areas and road networks shows that our MAMC algorithm reconstructs road networks better than existing methods. In our experiments, we observed that the quality of the trajectory data directly impacts the map construction result. Especially movement-based methods such as our MAMC algorithm are affected. In the presence of a considerable number of outliers, MAMC's performance degrades, while density-based methods perform better in these situations. However, those methods do not capture edge direction and turn types. Also, trajectories cannot be too sparsely sampled since MAMC relies on a certain point density to accurately reconstruct intersection node locations.

Directions for future research are as follows. Since node construction is at the core of the algorithm, the accuracy of the inferred nodes directly impacts the map construction result. Node inference assumes a typical 90° turn angle for selecting turn points and also adjusting their location based on trajectory geometry. Although we found that most intersections obey this constraint, relaxing this assumption could yield better results. Another aspect we want to explore is lane information. Preprocessing and filtering trajectories using machine learning methods will help us detect the habits of drivers in an effort to discern lane information and provide a more detail map construction result.

Acknowledgements

This material is based upon work supported by the National Natural Science Foundation of China (Grant No. 41631175), Nanjing University of Posts and Telecommunications Scientific Foundation (Grant No. NY218137), and the National Science Foundation (Grant No. CCF-1637541).

Data and source code availability

The data and source code that supports the findings of this study are available at the following link https://doi.org/10.6084/m9.figshare.12752546. The results are also published at http://www.mapconstruction.org.

References

Aanjaneya, M., Chazal, F., Chen, D., Glisse, M., Guibas, L., & Morozov, D. (2012). Metric graph reconstruction from noisy data. *International Journal of Computational Geometry & Applications*, *22*(4), 305-325.

Ahmed, M., & Wenk, C. (2012). Constructing street networks from GPS trajectories. In *European Symposium on Algorithms* (pp. 60-71). Springer Berlin Heidelberg.

Ahmed, M., Fasy, B. T., & Wenk, C. (2014). Local persistent homology-based distance between maps. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 43-52).

Ahmed, M., Karagiorgou, S., Pfoser, D., & Wenk, C. (2015a). A comparison and evaluation of map construction algorithms using vehicle tracking data. *GeoInformatica*, *19*(3), 601-632.

Ahmed, M., Karagiorgou, S., Pfoser, D., & Wenk, C. (2015b). *Map Construction Algorithms*. Switzerland: Springer International Publishing.

Bhattacharjee, S., Roy, S., & Bit, S. D. (2019). Post-disaster map builder: Crowdsensed digital pedestrian map construction of the disaster affected areas through smartphone based DTN. *Computer Communications*, *134*, 96-113.

Biagioni, J., & Eriksson, J. (2012a). Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation Research Record: Journal of the Transportation Research Board*, 2291, 61-71.

Biagioni, J., & Eriksson, J. (2012b). Map inference in the face of noise and disparity. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (pp. 79-88).

Brakatsoulas, S., Pfoser, D., Salas, R., & Wenk, C. (2005). On map-matching vehicle tracking data. In *Proceedings of the 31st International Conference on Very Large Data Bases* (pp. 853-864).

Bruntrup, R., Edelkamp, S., Jabbar, S., & Scholz, B. (2005). Incremental map generation with GPS traces. In *Proceedings of Intelligent Transportation Systems conference* (pp. 574-579).

Cao, L., & Krumm, J. (2009, November). From GPS traces to a routable road map. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems* (pp. 3-12).

Chen, C., Lu, C., Huang, Q., Yang, Q., Gunopulos, D., & Guibas, L. (2016, August). City-Scale Map Creation and Updating using GPS Collections. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1465-1474).

Cheng Y. (1995) Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), 790-799.

Davies, J. J., Beresford, A. R., & Hopper, A. (2006). Scalable, distributed, real-time map generation. *IEEE Pervasive Computing*, *5*(4), 47-54.

Deng, M., Huang, J., Zhang, Y., Liu, H., Tang, L., Tang, J., & Yang, X. (2018). Generating urban road intersection models from low-frequency GPS trajectory data. *International Journal of Geographical Information Science*, *32*(12), 2337-2361.

Edelkamp, S., & Schrödl, S. (2003). Route planning and map inference with global positioning traces. *Computer science in perspective*, (pp: 128-151). Springer, Berlin, Heidelberg.

Efentakis, A., Brakatsoulas, S., Grivas, N., & Pfoser, D. (2014). Crowdsourcing turning restrictions for OpenStreetMap. In *Proceedings EDBT/ICDT Workshops* (pp. 355-362).

Fan, H., Zipf, A., Fu, Q., & Neis, P. (2014). Quality assessment for building footprints data on OpenStreetMap. *International Journal of Geographical Information Science*, *28*(4), 700-719.

Fathi, A., & Krumm, J. (2010). Detecting road intersections from GPS traces. In *Proceedings of the International Conference on Geographic Information Science* (pp. 56-69).

Flanagin, A. J., & Metzger, M. J. (2008). The credibility of volunteered geographic information. *GeoJournal*, *72*(3-4), 137-148.

Fukunaga K, Hostetler L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1), 32-40.

Ge, X., Safa, I. I., Belkin, M., & Wang, Y. (2011). Data skeletonization via Reeb graphs. In *Advances in Neural Information Processing Systems* (pp. 837-845).

Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. *GeoJournal*, *69*(4), 211-221.

Guo, T., Iwamura, K., & Koga, M. (2007, July). Towards high accuracy road maps generation from massive GPS traces data. In *Proceedings of the International Geoscience and Remote Sensing Symposium* (pp. 667-670).

Haklay, M. (2010). How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and planning B: Planning and design*, *37*(4), 682-703.

Haklay, M., & Weber, P. (2008). Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, *7*(4), 12-18.

He, E., Bai, F., Bhagavatula, V., & Hay, C. (2018). Signal reconstruction approach for map inference from crowd-sourced GPS traces. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 472-475).

He, S., Bastani, F., Abbar, S., Alizadeh, M., Balakrishnan, H., Chawla, S., & Madden, S. (2018, November). RoadRunner: improving the precision of road network inference from GPS trajectories. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 3-12).

Huang, J., Deng, M., Tang, J., Hu, S., Liu, H., Wariyo, S., & He, J. (2018). Automatic Generation of Road Maps from Low Quality GPS Trajectory Data via Structure Learning. *IEEE Access*, *6*, 71965-71975.

Karagiorgou, S., & Pfoser, D. (2012). On vehicle tracking data-based road network generation. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (pp. 89-98).

Karagiorgou, S., Pfoser, D., & Skoutas, D. (2013). Segmentation-based road network construction. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 460-463).

Karagiorgou, S., Pfoser, D., & Skoutas, D. (2017). A layered approach for more robust generation of road network maps from vehicle tracking data. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, *3*(1), 3.

Kasemsuppakorn, P., & Karimi, H. A. (2013). A pedestrian network construction algorithm based on multiple GPS traces. *Transportation research part C: emerging technologies*, *26*, 285-300.

Kuntzsch, C., Sester, M., & Brenner, C. (2016). Generative models for road network reconstruction. *International Journal of Geographical Information Science*, *30*(5), 1012-1039.

Lee, W. C., & Krumm, J. (2011). Trajectory preprocessing. In *Computing with spatial trajectories* (pp. 3-33).

Li, H., Kulik, L., & Ramamohanarao, K. (2016). Automatic Generation and Validation of Road Maps from GPS Trajectory Data Sets. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (pp. 1523-1532).

Lyu, H., Sheng, Y., Guo, N., Huang, B., & Zhang, S. (2017). Geometric quality assessment of trajectory-generated VGI road networks based on the symmetric arc similarity. *Transactions in GIS*, 21(5), *984-1009*.

Mariescu-Istodor, R., & Fränti, P. (2018). Cellnet: Inferring road networks from gps trajectories. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, *4*(3), 8.

Mehdipoor, H., Zurita-Milla, R., Rosemartin, A., Gerst, K. L., & Weltzin, J. F. (2015). Developing a workflow to identify inconsistencies in volunteered geographic information: a phenological case study. *PloS one*, *10*(10), 1-14.

Neis, P., & Zielstra, D. (2014). Recent developments and future trends in volunteered geographic information research: The case of OpenStreetMap. *Future Internet*, *6*(1), 76-106.

Neis, P., & Zipf, A. (2012). Analyzing the contributor activity of a volunteered geographic information project—The case of OpenStreetMap. *ISPRS International Journal of Geo-Information*, *1*(2), 146-165.

Ni, Z., Xie, L., Xie, T., Shi, B., & Zheng, Y. (2018). Incremental Road Network Generation Based on Vehicle Trajectories. *ISPRS International Journal of Geo-Information*, *7*(10), 382.

Pfoser, D., & Jensen, C. S. (1999). Capturing the uncertainty of moving-object representations. In *Proceedings of the International Symposium on Spatial Databases* (pp. 111-131).

See, L., Comber, A., Salk, C., Fritz, S., van der Velde, M., Perger, C., & Obersteiner, M. (2013). Comparing the quality of crowdsourced data contributed by expert and non-experts. *PloS one*, 8(7), 1:11.

Shi, W., Shen, S., & Liu, Y. (2009). Automatic generation of road network map from massive GPS, vehicle trajectories. In *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems* (pp. 1-6).

Tang, L., Yang, X., Dong, Z., & Li, Q. (2016). CLRIC: collecting lane-based road information via crowdsourcing. *IEEE Transactions on Intelligent Transportation Systems*, *17*(9), 2552-2562.

Torre, F., Pitchford, D., Brown, P., & Terveen, L. (2012). Matching GPS traces to (possibly) incomplete map data: bridging map building and map matching. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems* (pp. 546-549).

Wang, J., Rui, X., Song, X., Tan, X., Wang, C., & Raghavan, V. (2015). A novel approach for generating routable road maps from vehicle GPS traces. *International Journal of Geographical Information Science*, 29(1), 69-91.

Wang, J., Wang, C., Song, X., & Raghavan, V. (2017). Automatic intersection and traffic rule detection by mining motor-vehicle GPS trajectories. *Computers, Environment and Urban Systems*, *64*, 19-29.

Wang, S., Wang, Y., & Li, Y. (2015). Efficient map reconstruction and augmentation via topological methods. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 1-10).

Wenk, C., Salas, R., & Pfoser, D. (2006). Addressing the Need for Map-Matching Speed: Localizing Global Curve-Matching Algorithms. In *Proceedings of the 18th Int'l Conference on Scientific and Statistical Data Management* (SSDBM) (pp. 379-388).

Worrall, S., & Nebot, E. (2007, December). Automated process for generating digitised maps through GPS data compression. In *Proceedings of the Australasian Conference on Robotics and Automation* (pp. 1-6).

Wu, H., Xu, Z., & Wu, G. (2019). A Novel Method of Missing Road Generation in City Blocks Based on Big Mobile Navigation Trajectory Data. *ISPRS International Journal of Geo-Information*, *8*(3), 142.

Xie, X., Bing-YungWong, K., Aghajan, H., Veelaert, P., & Philips, W. (2015). Inferring directed road networks from GPS traces by track alignment. *ISPRS International Journal of Geo-Information*, 4(4), 2446-2471.

Yang, W., Ai, T., & Lu, W. (2018). A method for extracting road boundary information from crowdsourcing vehicle GPS trajectories. *Sensors*, *18*(4), 1261.

Yang, X., Tang, L., Stewart, K., Dong, Z., Zhang, X., & Li, Q. (2018). Automatic change detection in lane-level road networks using GPS trajectories. *International Journal of Geographical Information Science*, *32*(3), 601-621.

Zandbergen, P. A. (2009). Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning. *Transactions in GIS*, 13(1), 5-25.

Zhang, L., Thiemann, F., & Sester, M. (2010). Integration of GPS traces with road map. In *Proceedings of the second international workshop on computational transportation science* (pp. 17-22).

Zhang, Y., Liu, J., Qian, X., Qiu, A., & Zhang, F. (2017). An Automatic Road Network Construction Method Using Massive GPS Trajectory Data. *ISPRS International Journal of Geo-Information*, *6*(12), 400.