AUTOMATIC GENERATION OF ROAD NETWORK DATA FROM SMARTPHONE GPS TRAJECTORIES

By

Zheng Niu

B.Eng. (Hons), Ryerson University, Ontario, Canada, 2010

A thesis presented to Ryerson University

In partial fulfillment of the requirements of

Master of Applied Science

In the program of Civil Engineering (Geomatics Engineering)

Toronto, Ontario, Canada, 2013

Author's Declaration

I hereby declare that I am the sole author of this dissertation. This is a true copy of the dissertation, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

_____

I further authorize Ryerson University to reproduce this dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

I understand that my dissertation may be made electronically available to the public.

AUTOMATIC GENERATION OF ROAD NETWORK DATA FROM SMARTPHONE GPS TRAJECTORIES

Zheng Niu

Master of Applied Science 2013

Civil Engineering (Geomatics Engineering)

Ryerson University

Abstract

Keeping road network databases up-to-date is crucial to Geographical Information System (GIS) applications such as vehicle navigation. The vector road centerlines extracted from satellite images or in-car Global Positioning System (GPS) devices are likely to be inaccurate due to costly and labour intensive or long updating circle. The GPS data crowdsourced through smartphones provides an emerging source for refining road map due to its rich spatio-temporal coverage and reasonable level of accuracy. This thesis introduces an optimized methodology to automatically generate road network data from smartphone GPS data without using any reference maps. The horizontal accuracy of the extracted road centerlines, measured as a root mean square of 1.424 m and 1.252 m for curved and straight road segments respectively, is better than that of some existing road datasets. The outcome of this research will provide a new way of generating a more accurate and up-to-date road network databases.

Acknowledgements

This thesis would not have been successfully completed without the efforts and valuable inputs from many people to whom I owe my deepest gratitude. I have been very fortunate with my supervisor, Dr. Songnian Li. I will forever be thankful to Songnian for his continuous support and patience over the last three years, as well as valuable guidance and effort in keeping my research on the right direction. He also gave me the freedom and encouragement to seek various solutions for resolving problems without objection.

I also own many thanks to my best friends who helped me through the bad times during the thesis research. I would like to acknowledge special thanks to Dr. Waiyeung Yan, Chenfeng Xiong, and Annie Chow. Yan has given me constructive and meticulous comments which were an enormous help to my undergraduate and graduate research since 2010. Advice and comments given by Chenfeng are really helpful in learning and developing practical programming skills of python and web mapping. Annie inspired my final effort to continue and then complete my prototype design despite the enormous pressures.

I have greatly benefited from the education at Ryerson University Geomatics Engineering program, which forms the basis of this thesis. I would like to show my greatest appreciation to all former and current professors in our faculty, especially Dr. Ahemd Shaker, Dr. Ahmed El-Rabbany, Dr. Jinyuan Liu, Dr. Mike Chapman, Dr. Mustafa Berber, and Dr. Said Easa. I also want to thank Robin Luong, Wei Huang, and Dr. Xintao Liu for their generous supports and warm concerns.

Finally, my heartfelt gratitude goes to my wife, Yingni, my parents, and grandparents because of their love, understanding, encouragement and sacrifice in all forms.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

1.1. Research Motivation

GPS data crowdsourced through in-car devices is an emerging source of inexpensive data that can be used to provide real-time traffic information, identify traffic patterns, and predict traffic congestions (Cao and Krumm, 2009). This new data collection method is a type of volunteered geographic information that overcomes the high cost of using the traditional intrusive and non-intrusive on-road sensors (e.g., inductive loops) methods, as well as their limitations in data coverage (Grossman et al, 2005). Over the last few years, studies have been done using in-car GPS devices (Yoon et al., 2007; Young, 2007) and more recently with smart phones (Herrera et al., 2010; Li et al., 2012), to develop data processing techniques, traffic patterns and traffic prediction models, geometric modeling of roads, and generation of road network maps.

The benefits of using GPS trajectory data from smartphones or mobile devices equipped with the GPS receiver and the wireless communication equipment lies in the low cost of data collection, no need for specialized and expensive data collection equipment, potentially massive data collected by road users capturing real-time changes and status of the roads, and rich spatial and temporal coverage. The rich real-time information (e.g. coordinates, traveling timestamp, speed, and direction) of an individual's current location is an emerging source of inexpensive data that can be used for fast updating of a road network database. With proper data processing and analysis in place, it is possible to update road network data in real-time or near real-time (e.g. detecting unknown roads). Although the accuracy of the data is not ideal, ranging from 6-10 m (Haklay and Weber, 2008), the huge number of positioning points form a point cloud which offer excellent samples for statistically significant results.

Keeping the road network database up-to-date is clearly important to many applications based on Geographic Information System (GIS), such as navigation, intelligent transportation system, traffic congestion prediction, emergency handling, as well as traffic surveillance and management. In fact, extracting the road information from the high resolution satellite image has been dominant in updating road network for years

because of the rich multi-spectrum information and stable acquisition of imagery (Park and Kim, 2001; Zhao et al., 2002; Mohammadzadeh et al., 2004 & 2006; Lin et al., 2008 & 2009).

More recently, the GPS trajectory data has been used to extract road geometric data for road network database updating and road maps refinement. This new approach entails a fast, inexpensive way of updating existing road maps and refining road maps with real-time changes (e.g., new roads not showing in the existing road network data). Studies and experiments have been done on extracting road centerlines and other road network attributes such as number of lanes. Some used existing road maps as prior knowledge to find road centerline (e.g., Guo et al., 2007; Zhang et al., 2010). Others examined different ways of extracting road centerline without prior knowledge about the road, such as using Artificial Neural Network (ANN) and GPS trajectories mining methods (e.g., Schroedl et al., 2004; Ekpenyong et al., 2009).

Although researchers have proposed various approaches to generate the road network based on GPS data, these methods are not suitable for generating road centerlines for two scenarios simultaneously, one-way roads and two-way roads. Consequently, studies conducted by (Shi et al., 2009a; Shi et al., 2009b; Limaa and Ferreiraa, 2009; Guo et al., 2010; Zhao et al., 2011; Liu et al., 2012; Karagiorgou and Pfoser, 2012) were restricted to local roads in urban areas because the assumed road width cannot be suitable for both highway and local roads. In this regard, it is necessary to develop ways to use the vast amount of valuable GPS trajectory data in the generation of the bi-directional road centerline, which models each direction of travel as a separate alignment along the same road. Herman (2002) summarized the benefits of using bi-directional road centerline such as: better matching with other data layers; providing a truer presentation of the highway network, and analyzing the road information more effectively (e.g., lane closures pertaining to certain direction of travel).

1.2. Research Objectives

The ultimate goal of this thesis is to develop a cost-effective road network data extraction methodology based on GIS and crowdsourcing GPS data from smartphone users. This will allow mobile technology companies, providing location-based services, to take

advantages of their GPS data collected through mobile navigation applications for developing the road network data acquisition and updating system without matching GPS data to existing road maps.

The biased GPS measurements cause uncertain level of noise that degrades the quality of GPS trajectory data collected by different types of smartphones. As the GPS trajectories are collected regardless of where the smartphone user is (e.g. driving on roads, idling, waiting at traffic lights, or inside buildings), it is challenging to completely filter out such point clouds from raw data. The distribution of positioning data of moving vehicles could be normal, scattered or multimodal within a certain width around a road centerline. However, it is implausible to utilize a fixed value of road width to cluster positioning points belonging to the same road segment, especially at a road split or merge. GPS trajectories that are travelling along two parallel roads are usually overlapped at the road median due to the biased GPS measurements. It is difficult to define a distance threshold to classify them into corresponding road segments. At road splitting/merging or road curvature sections, portions of a GPS trajectory could be offset from a road, even though its predecessor and successor travelled along the same road.

In order to minimize the negative effects of biased GPS measurements collected by crowd-sourced smartphone users and ensure the quality of extracted road centerlines, the objectives of this research are summarized as:

1) To develop an automatic road network data extraction methodology based on GIS and crowdsourcing GPS data from smartphone users without using any digital road map as reference, includes:
    - To develop methods to reduce the volume of smartphone GPS data without affecting underlying road network geometry;
    - To develop methods to distinguish GPS trajectories on nearby parallel roads and to separate GPS trajectories on road segments that have splits of a narrow angle;
    - To develop methods to construct bi-directional road centerlines and topological connectivity at road intersections and Y-split sections;

2) To evaluate the horizontal accuracy of the generated road centerline by comparing with the Ministry of Transportation Ontario (MTO) highway alignment data; and
3) To implement algorithms by developing standalone python script tools that automate the overall workflow from road centerline extraction to the connectivity of road network.

1.3. Limitations

The research reported in this thesis is subject to several possible constraints and limitations listed as follows:

1) Millions of smartphone GPS data points covering the road network in southern Ontario cannot be processed together, due to the memory limitation to running 32-bit PythonWin[1] on the 64-bit Microsoft Windows 7 operating system. Therefore, the GPS data points are tilted in certain region for data processing.
2) Road centerlines are omitted if insufficient smartphone GPS data points were collected on highway ramps and minor roads.

1.4. Contributions

This research makes the following contributions to the overall field of knowledge in this area:

1) The thesis presents a comprehensive understanding of various efforts that have been devoted to extract proper road information from different data sources. It also summarizes the limitations, strengths, and similarities of existing methods.
2) The hybrid method taking GPS trajectory and GPS data point as the basic processing units is developed for generating bi-directional road centerlines without using a reference map. The extracted road network data can support with practical requirements of navigation and linear referencing system, because each extracted road centerline contains geographic location (e.g. starting and ending

[1] PythonWin is the name of interactive development environment for running python scripting programs.

positions) and corresponding attributes (e.g. moving direction and turning direction).

3) The developed research prototype demonstrates the feasibility of automating the process of extracting road network data instead of manual editing the connectivity of the extracted road centerlines.
4) The results obtained from performance testing provide measures of the effectiveness of the proposed alternative solution to road network extraction studies.

1.5. Thesis Organization

The thesis is organized into six chapters, starting with this chapter that presented the overall introduction of research motivation, challenges and objectives, research contributions, and limitations that may affect building a detailed and fine-grained road network database. Chapter 2 reviews published methods on extracting road centerlines by using image processing techniques, clustering algorithms, or the combinations; as well as the polyline merging algorithms that were used to find the representative polyline of a group of polyline segments. Chapter 3 presents the detailed descriptions of algorithms taken to extract the basic road data (bi-directional road centerlines) and construct connectivity of road centerlines without matching GPS data points to a reference road map. Chapter 4 gives some insight on the selection of the study area, the acquisition of the experimental GPS data, accuracy of smartphone GPS data, and statistical analysis of original GPS data. Chapter 5 presents results of the proposed methodology overlaid with digital orthophoto image of 15-cm spatial resolution obtained from Ministry of Natural Resources Ontario (MNR). The comparative analysis is done by quantitatively evaluating the accuracy of the generated road centerlines against the ground truth data, MTO highway alignment data. The effects of GPS data point density on the extraction of road centerlines are analyzed. Chapter 6 concludes this thesis and shares thoughts for future work.

Chapter 2. Literature Review

This chapter presents an overview of existing literature related to road network extraction. Section 2.1 summarizes and points out the common limitations of the existing work in using low/high resolution imagery to extract road data, either semi-automatically or automatically. Section 2.2 provides a detailed review on current clustering techniques used for extracting road data from GPS data; and summarizes these road extraction methods based on the vector data, and provides observations to show the necessity of this research.

2.1. Road Extraction Based on Satellite Imagery

Road network extraction based on satellite imageries has been studied by numerous scientific practitioners. Various efforts have been devoted to extract proper road information from different data sources by employing different preprocessing techniques. These include reducing noise occluding along the road pixels; segmentation of road features with respect to the linear elements or elongated regions with edges; road tracking methods for eliminating road-like pixels; and different grouping methods to connect the traced road segments (Fortier et al., 1999). After data preprocessing, road extraction can be performed based on two categories of approach: semi-automatic and automatic approaches.

The semi-automatic road extraction method usually requires the human intervention to detect road pixels, such as selection of seed points and search direction for initializing the road tracking algorithm, or selection of training area in classification based methods. In template matching method as addressed by Park and Kim (2001), Zhao et al. (2002), and Lin et al. (2008 & 2009), the operator has to choose the starting point and the directional point on each road in order to guide the direction of extracting road centerline. Seed points coarsely descripting the road geometry are also needed to initialize the modified merit function (Dal Poz and Do Vale, 2003) based on the dynamic programming algorithm proposed by Gruen and Li (1997). Mena and Malpica (2005) used the manually selected training area from existing road network database in the process of image binarization to extract road segments. These operator-aid techniques are

capable of handling more complex road geometry (Guo et al., 2008), but are dependent on the natural skills of the operator who identifies the objects in the image as roads (Dal Poz and Do Vale, 2006). With respect to the drawbacks of human errors and labor intensive operations, the fully automated road extraction methods are highly desirable for improving the efficiency of generating and updating road data in GIS (Yun and Uchimura, 2007; Rajeswari et al., 2011).

In automatic road extraction methods, road features are separated from other surrounding objects in image by using approaches such as hypothesis-verification strategy. The fuzzy-based method is used (Mohammadzadeh et al., 2004 & 2006) to identify the road pixels and incorporated the morphological techniques to automatically eliminate the occlusion of roads caused by shadow or vegetation. Hu and Tao (2005) developed the ribbon road detector to separate the roads from other features based on the profile matching and analysis methods as well as a model-based verification strategy. Peteri and Ranchin (2002) presented the street surface reconstruction algorithm integrating the deformable contour models with the multi-resolution analysis for extracting the road boundaries. Dal Poz et al. (2006) adopted a prior road knowledge-based method to automatically extract road segments from elongated road surface detected by the Canny edge detector.

2.1.1. Major Extraction Techniques

Both semi- and fully automatic extraction of road network from satellite imageries has been the subject of extensive research during the past decades. According to Fortier et al. (1999), Mena (2003), and Quackenbush (2004), the major extraction techniques used in various road extraction methods are: road tracking, mathematical morphology, multi-resolution analysis, classification-based methods, and deformable contour models.

Road tracking is a widespread technique mainly used for semi-automatic road extraction based on the operator-selected initial seed points per each road segment in the image. In (Shukla et al., 2002), the anisotropic diffusion technique is adopted to reduce image noise and preserve the edges of image objects. All edges in the diffused image were detected by the Canny edge detector. Then, the path following algorithm starting with operator-selected initial seed points is utilized to derive the road width. At every

seed point, the road width is the distance along the projected line with right angle crossing the orientation of two closest seed points. The road center point was calculated by averaging those two intersecting points on edges of the road. Similarly, Zhao et al. (2002) also used the Canny edge detector to extract road edge pixels from the binary image containing road and non-road pixels. The edge pixels along the straight and continuous road were smoothed by using a 3x3 filtering window. Likewise, Gao and Wu (2004) as well as Mena and Malpica (2005) utilized the median filter in the data preprocessing step for image smoothing. Seed points were then extracted by tracing the road edge pixels. The implementation of the road line tracing requires the input of two control points, the starting point and the directional point at the road intersection or Y-split section. In (Park and Kim, 2001; Lin et al., 2008), the implementation of road tracking is highly dependent on the operator's skills. The starting point and the directional point must be manually chosen on each road in order to guide the direction of extracting road centerline while using the template matching method. The selection of control points directly affects the accuracy of the extracted road lines.

Mathematical morphology, initiated in the late 1960s by Matheron and Serra (2002), aims at extracting geometric objects from the image based on their shape and size. Mohammadzadeh et al. (2004 & 2006) utilized the morphological trivial opening and the granulometry to identify the main road pixels from the classified image, in order to overcome the perturbations caused by nearby features with similar spectral characteristics as road surfaces. The trivial opening is used to repair the classified pixels by removing noise from the road surface and the driveways connecting to the main roads. Instead of extracting road from the classified image, Valero et al. (2010) applied the advanced directional mathematical morphology to directly extract roads from the original satellite image. Their experimental work identified the linear geometrical pixels according to the morphological profile, which was constructed by applying the granulometric analysis to each pixel in original image.

Multi-resolution analysis in connection with the road extraction topic enables to minimize the occlusion of roads caused by other objects in the image. Fortier et al. (1999) addressed that road objects are differently represented in the low-resolution and high-

resolution satellite images. Roads mainly appear as lines in the low-resolution image and as elongated regions with more or less parallel edges in the high-resolution image. A number of road extraction methods combining multi-scale or multi-resolution analysis and deformable contour models were proposed in the 90s, because deformable contour models can be used to speed up the detection of road boundaries in the image segmentation (Mena, 2003). Mayer et al. (1997) combined the multi-scale analysis with the ribbon snake to extract road boundaries from the digital aerial image, while Baumgartner et al. (1996) extracted roads based on the fusion of the extracted line segments from the low-resolution aerial image and the detected edges from the high-resolution image. Following the previous work described above, Peteri and Ranchin (2003) presented the street surface reconstruction algorithm integrating deformable contour models with the multi-resolution analysis and the wavelet transform for extracting the boundaries of road segments and intersections.

Road data in the satellite image can be extracted by using classification-based methods based on texture analysis, wavelet transform, or fuzzy logic system as well. Apart from conventional fuzzy logic classification techniques utilized in the identification of road pixels (Carsten et al., 1997; Chen and Lu, 2002; Yun and Uchimura, 2007), Mohammadzadeh et al. (2004) presented the developed fuzzy logic algorithm based on the previous work done by Melgani et al. (2000). Every pixel in each band (Red, Green, and Blue) of the pan-sharpened IKONOS image is classified into five categories (as known as membership functions) according to its closeness to the mean grey value of the arbitrary user-selected road pixels. Then, the fuzzy "if-then" rules are applied to distinguish the class of road pixels from other non-road classes. According to Mena and Malpica (2005), three different classification methods are combined to extract roads in the regions around a pre-existed road in GIS database. These methods include Mahalanobis distance, describing the closeness of the target pixel to road pixels in the training area; Bhattacharyya distance, analyzing the closeness of the target pixel's neighbors to road pixels in the training area; and the texture cube technique used to analyze the color relation. Zhang and Couloigner (2004) investigated the use of wavelet transform for detecting road junctions and centerline pixels. Gao and Wu (2004) introduced a spatial reasoning-based method into the extraction of road networks in a

large region. Firstly, road pixels are identified by using the unsupervised classification method and then converted to a binary image containing road and non-road pixels. Secondly, spatial filter is applied to remove noises unrelated to the road class. Thirdly, the directional cone search method is utilized to link pixels in the same road segment. Finally, the skeleton of only one pixel wide is obtained for each road by the implementation of the thinning algorithm.

2.1.2. Discussion

The above review indicates that substantial studies have been carried out to semi- or fully automatically extract roads from the satellite imageries. The semi-automated extractions (Park and Kim, 2001; Lin et al., 2008 & 2009) are able to handle more complex road geometry than the fully-automated methods, but are dependent on the natural skills of the operator who sets the objects in the image as roads (Dal Poz et al., 2006) and the assumed constant road width. High resolution stereo satellite image, for example, IKONOS, Quickbird, can overcome to limitation of medium-resolution satellite image for extracting the 2D/3D highway alignments (Shaker et al.， 2010 & 2011); however, semi-automatic operation has to be performed to extract the 3D alignments. The fully automated road extraction methods are highly desirable for improving the efficiency of generating and updating road data in GIS (Yun and Uchimura, 2007; Rajeswari et al., 2011). However, only few automated methods produced satisfactory results with quantitative accuracy evaluation. For example, Mena and Malpica (2005) assessed their results and found that the average Root Mean Square error of 1.2 m was close to the average resolution of the IKONOS image (2 m).

Most studies are still limited to specific study areas and may be futile whenever a different geographic region is involved. They are moderate successful in extracting salient major roads but still cannot eliminate the difficulty of resolving the occlusion of roads caused by clouds and shadows of vehicles, vegetation, or other nearby objects. For example, the extracted road centerlines are off from the actual location or represented as a group of zigzag polylines even with the use of the line smoothing algorithm, such as Douglas-Peucker. The local roads are missing or disconnected to major roads.

On the other hand, most methods were developed based on the user's prior knowledge of the road in the study area, such as the assumption of the constant road width. These methods are not capable of creating accurate bi-directional road centerlines on all roads in a large region. In addition, the source and availability of satellite imageries also affects the road extraction due to its long updating circle and high cost. Table 1 lists examples of the popular satellite imageries utilized by previous studies on road extraction and their spatial resolution, revisit rate (temporal resolution), and minimum cost in the market.

Table 2.1: Sources of sample satellite imageries utilized by previous studies

Satellite Imagery	Spatial Resolution[2]	Revisit Interval[3]	Min. purchase & Cost[4]
QuickBird	0.6 m – 2.4 m	1-3.5 days	25 km^2 – archive ($14-17 / km^2) 100 km^2 – tasking ($20-23/ km^2)
IKONOS	0.82 m – 3.2 m	3-5 days	49 km^2 – archive ($35/ km^2) 100 km^2 – tasking ($35/ km^2)
SPOT	1.5 m – 20 m	2 -3 days	169 km^2 ($6.5/$km^2$)

It is still a long way to apply the satellite image based road recognition to the practical road updating due to the above discussed challenges and limitations. However, the rapid development of GPS and wireless communication technologies provides an alternative data source for extracting road geometric data for road network database updating and road maps refinement. This new approach entails a fast, inexpensive way of updating existing road maps and refining road maps with real-time changes.

2.2. Road Extraction Based on GPS Data

GPS data collected using in-vehicle GPS devices is an emerging source of inexpensive geospatial data. It can be categorized as a type of volunteered geographic information

[2] http://www.geoeye.com/CorpSite/products/earth-imagery/geoeye-satellites.aspx#ikonos
[3] http://www.geoeye.com/CorpSite/products/earth-imagery/geoeye-satellites.aspx#ikonos
http://www.nrcan.gc.ca/earth-sciences/geography-boundary/remote-sensing/fundamentals/1954
[4] http://www.landinfo.com/products_satellite.htm

(Goodchild, 2007) that can be used to provide real-time traffic information, identify traffic patterns and predict traffic congestions. Such data collection method does not require dedicated devices and infrastructure because of the mature GPS and cellular communication technologies. Therefore, this method has lower cost and shorter updating cycle than those collected through satellite images or ground surveying by probe vehicles. Over the last few years, studies have been carried out using in-vehicle GPS devices (Schroedl et al., 2004; Worrall and Nebot, 2007; Zhang et al., 2010; Chen and Krumm, 2010; Jang et al., 2010) and more recently with smart phones (Li et al., 2012) to develop data processing techniques, geometric modeling of roads, and generation of road network maps. The smartphone GPS data are a type of volunteered geographic information that has rich spatial and temporal coverage but comes with lower accuracy than that of in-car GPS devices.

Biagioni and Eriksson (2012a) addressed that existing road map generation methods can be categorized by the types of clustering algorithms: K-means, trace merging, or kernel density estimation (KDE). In K-means based approach, one of partitioned spatial clustering methods in GIS, a user-specified number of clusters together with fixed proximity measurements (distance and directional differences) are used to group nearby GPS points into clusters. In trace-merging based approach, GPS sub-trajectories on the same road segment are incrementally merged to a unique polyline segment representing the road centerline. In KDE-based approach, an integration of density analysis and grid-based rasterization, GPS points within the study area are first quantized into a finite number of cells. Image processing techniques are then used to extract single-pixel road centerline from contiguous dense cells. However, not all published studies on extracting road network from GPS trajectories can be grouped into these categories. For example, the agglomerative hierarchical clustering method is adopted by Worrall and Nebot (2007) and Guo et al. (2010) to generate road centerlines, where each point is treated as a separate cluster and then successively merged with nearby points into clusters. Lee et al. (2007) and Li et al. (2010) used the modified density-based clustering method (DBSCAN) to group together GPS sub-trajectories on the same road based on their connectivity and density. Cao and Krumm (2009) used the simulated attraction force model to relocate every GPS trajectory towards the middle of

roads. It is difficult to categorize all existing methods by their adopted spatial clustering algorithm. Therefore, it is suggested in this thesis to classify early studies by the basic processing unit involved in spatial clustering techniques, including points, polylines, or grid.

This section aims at summarizing road extraction techniques with respect to their specific strengths, similarities, and limitations. Through this comprehensive review, suitable techniques were selected for further development of the proposed methodology for extracting road centerlines from massive smartphone GPS data. These techniques include: data preprocessing, data clustering, and road centerline inference (topological connecting, and geometric merging). Table 2.2 summarizes the literature on road map generation in terms of data source, coverage, processing unit, result type, and evaluation method.

2.2.1. Point-based Method

In point-based road map generation methods, road centerlines are usually retrieved from GPS trajectories by employing point spatial clustering algorithms, including the agglomerative hierarchical clustering and partitional clustering. Work under this category includes studies conducted by Edelkamp and Schrodl (2003), Schroedl et al. (2004), Worrall and Nebot (2007), Zhang et al. (2010), and Guo et al. (2010). The road generation process consists of three main steps, preprocessing GPS points, extracting the road center point from clustered GPS points, and inferring road centerline by linking cluster centers.

The agglomerative hierarchical clustering starts with each point as a separate cluster and then is successively merged with nearby points into clusters (Wang and Hamilton, 2010). Most of earlier works use a specific algorithm and a particular set of threshold values to tackle with the measurement errors caused by the limited GPS accuracy and the low sampling rate, in order to reduce negative impacts on the quality of output data. The Douglas-Peucker algorithm based on maximum distance is frequently used to simplify GPS trajectories by removing redundant points (Limaa and Ferreiraa, 2009; Biagioni et al., 2011). The angle-threshold based smoothing filter is used in the weighted clustering algorithm (Wang et al., 2011). Li et al. (2012) simply reduced the

data size by utilizing a threshold value of the vehicle speed. Liu et al. (2012) only reserved critical points at where the direction of a trajectory changes rapidly. Shi et al. (2009a) and Niehoefer et al. (2009) pruned GPS trajectories according to threshold values of direction difference, acceleration, and/or velocity change between two consecutive points.

To overcome above limitations of data reduction, Guo et al. (2010) introduced a two-step approach to reduce the data redundancy and volume on roads, while preserving the underlying road network. The approach proposed by Guo et al. (2010) starts with a moving-window smoothing process which brings every point closer to the center of the road. Every point is relocated to the new location by averaging the coordinates of its nearby points. And then, the modified distance-based clustering algorithm, which focuses on both topological and geometrical simplifications, is applied to link extracted representative points from smoothed points.

Worrall and Nebot (2007) proposed an exceptional method without removing noise. It compresses the raw GPS data collected from operational vehicles at a mining site by clustering them according to similar position and directions. Their method can only be applied to GPS trajectories following a fixed driving route, where overtaking or changing lanes never happened. New GPS points are directly added into corresponding cluster based on the distance and directional differences, without any further editing. The cluster center is calculated by averaging all GPS points within the same cluster. To form a coherent chain representing a road, cluster centers are linked together based on thresholds of distance and bearing differences, as well as a semantic road-knowledge based rule: one cluster center only has one forward link but could have more than two backward links when there is an intersection. Finally, the regression analysis is performed by using non-linear least square fitting to generate better presentations for straight and arc road segments. However, this method is restricted to the single road for two reasons:

1) Different road geometries require different threshold values used to identify a group of linked cluster centers with similar curvature; and

2) The non-linear least square fitting on an arc requires estimation of center and radius of the circle for connected cluster centers on each road.

The partitional clustering assisted by map-matching techniques starts with user-selected clusters and iteratively reallocated points to clusters (Wang and Hamilton, 2010). Zhang et al. (2010) used GPS trajectories to refine the existing road maps by applying the fuzzy C-means clustering method and a reference map from OpenStreetMap. Firstly, the map matching method based on a threshold value of 20-degree direction change and the profile with 30-m width orthogonal to the existing road is used to identify GPS points on the same road. Then, fuzzy c-means clustering algorithm is applied to separate GPS points from nearby parallel roads in terms of distances to its left- and right-side cluster centers, which are the weighted mean of all GPS points in each cluster. To extract road centerlines at highway ramps, the speed values are also taken into account because the speed of vehicle on highway exits is much slower than on highway (Zhang et al., 2010). The design speed on highway ramp is indeed slower than that on highway. Nevertheless, making use of the speed values of volunteered GPS data as a criterion for separating highway ramps from highways are improbable while the traffic on both highway and ramps is possibly going as 30 or 40 km/h in rush hour. The problem of the reference map assisted methods is that they are restricted to the particular area where there are existing roads. The experimental work conducted by Zhang et al. (2010) showed that the position accuracy of extracted road centerlines is affected by the quality of the original road map and the number of GPS points. Errors in the reference map are propagated to the output by using map-matching method; and fewer numbers of GPS points on an existing road cause the extracted road centerline large offset from its actual location.

Edelkamp and Schrodl (2003) and Schroedl et al. (2004) generated road centerlines by applying spline fitting on differential GPS datasets. All GPS points are partitioned into sequence of road segments of a commercial base map by implementing a map-matching module based on Dijkstra's algorithm. For clustered points belonging to each road segment, B-spline approximation is applied to generate road centerlines. The drawback of this approach is that the number of control points must be defined according to the number of sample points in a cluster and a small scale factor. The locations of

control points must be estimated by human inspection and then adjusted according to the trade-off between the mean offset of all points (on the same road) from the generated road centerlines and the second derivative of curvature of piecewise road centerline segments. It may be a good solution to create a curve to fit actual GPS points on the same road, but cannot match with the actual complex road shape. Therefore, Liu et al. (2012) suggested an alternative and simple way to determine the number of control points and their locations, so as to obtain better fitting performance for various road geometries. The shape-aware fitting is adopted to choose control points where the directions of aggregated GPS trajectories start to change.

In addition to agglomerative hierarchical clustering and partitional clustering methods, Cao and Krumm (2009) proposed an alternative GPS data clarification algorithm for minimizing the effect of GPS noise and clustering GPS points together on the same road, and an incremental graph generation algorithm for capturing the connectivity and geometry properties of the road network. Wang et al. (2011) proposed a weighted clustering algorithm embedded with angle-threshold based smoothing to improve the effective and efficiency of the GPS data clarification algorithm presented by Cao and Krumn (2009). The weight factor, which involves the speed of a point and the direction change over three consecutive points, is introduced into the physical attraction model in order to achieve better convergence of all GPS points along the same road.

In road clarification algorithm, Cao and Krumm (2009) addressed that the final position of each GPS point under the action of attraction and spring forces should not be far away from its original position with respect to the center of the road. This assumption keeps every GPS point from being grouped to the wrong road. Every GPS point is relocated on the direction of the resultant force by considering different roads of opposite directions. The resultant force includes the attraction forces caused by nearby GPS trace segments and the spring force from each GPS point's original position. Roads of opposite directions are differentiated by considering the attraction force of nearby trace segments to the GPS point together with the topological relationship to its nearby trace segments.

In the graph generation algorithm, GPS points along the same road are merged together based on distance amongst them and similar moving direction, but no cluster

center is calculated from the grouped GPS points. Making use of the first searched GPS point as the final road center point reduces the positional accuracy of the generated road centerline. In addition, this method is designed for the extraction of roads from a particular area. For instance, parameters governing the resultant forces in the road clarification algorithm are estimated by performing theoretical analysis on only two road scenarios, nearby roads in similar direction and road split, within their study area. The theoretical analysis is dependent on assumed threshold values for the minimum distance between two parallel roads and the angle between them before splitting. In reality, there are no constant values for those thresholds due to the nature of complex road geometry. Besides, this study did not perform any quantitative accuracy assessment of the derived routable road map before it was tested in answering route planning queries. In order to produce a digital road map that meets with practical requirements such as navigation and mapping, the quality of road map in terms of positional accuracy must be assessed and justified (Willrich, 2002).

After the cluster centers are extracted from a set of GPS points, the aforementioned studies show that the road centerlines can be formed in four different approaches. The first is to link cluster centers together according to geometric relationship (directional difference and distance thresholds) (Jang et al., 2010; Zhang et al., 2010; Li et al., 2012). The second is to combine the semantic rule with geometric relationship into the connection of cluster centers (Worrall and Nebot, 2007). The third is to use B-spline approximation to fit road curves based on control points (Edelkamp and Schrodl, 2003; Schroedl et al., 2004; Wang et al., 2011). The fourth is to perform spatial queries in terms of topological and geometrical relationships amongst extracted center points and GPS trajectories passing through them, and to connect adjacent cluster points (Limaa and Ferreiraa, 2009).

2.2.2. Polyline-based Method

Most polyline-based road extraction methods consist of two steps: GPS sub-trajectory clustering and merging. They have been utilized in many applications, such as animal movement and hurricane tracking, to reveal the underlying trends of moving objects based on their similarity (Lee et al., 2007; Kharrat et al., 2008). Lee et al. (2007)

proposed a trajectory-clustering algorithm (TRACLUS) based on partition-and-group framework to group similar sub-trajectories into a cluster. Each trajectory is simplified and split into a set of polyline segments. Similar polyline segments are grouped into a cluster by using the modified polyline density-based clustering derived from the point density based spatial clustering of applications with noise (DBSCAN) algorithm. Besides inheriting all characteristics from the point DBSCAN algorithm, TRACLUS takes into account the number of trajectories from which polyline segments are grouped into each cluster so as to avoid the generation of the single-trajectory cluster. Moreover, TRACLUS employed the distance function (Chen et al., 2003) in the process of polyline clustering instead of using the Euclidean distance. The distance function consists of three components, parallel distance, perpendicular distance, and angle distance. Nevertheless, this algorithm is primarily developed to discover common trajectories from free moving objects, such as hurricane track dataset and animal movement dataset.

To deal with trajectories collected from the constraint network, such as vehicle GPS trajectories, Li et al. (2010) modified and incorporated TRACLUS algorithm in the incremental trajectories clustering method (TCMM). The perpendicular distance is replaced by the distance between centers of polyline segments because vehicle GPS trajectories on the road are denser than free moving objects. However, both TRACLUS and TCMM are sensitive to the parameter of allowable distance amongst polyline segments. In order to obtain optimal quality of clustering, algorithms have to be repeatedly performed by using different values of the allowable distance. Similarly, Ahmed and Wenk (2012) developed a simple undirectional street-network constructing algorithm by matching GPS sub-trajectories to existing map based on Frechet distance. Unlike methods in (Lee et al., 2007; Li et al., 2010), the performance of this algorithm only involves one precision parameter, which is subject to the measurement error of each GPS trajectory.

Instead of using distance function as the unique criterion to group GPS sub-trajectories, Liu et al. (2012) presented an alternative solution. Polyline segments with similar direction and short geographical distance (Lumelsky, 1985) are initially grouped into a cluster. Cluster refinement is then performed to split the initialized cluster into two

or more new clusters if it contains several different roads. The second step provides an innovative solution to tackle with the complex road geometry as road merging or splitting. However, the results of the study conducted by Liu et al. (2012) demonstrated that the assumed uniform road width prevents this approach from constructing the continuous bi-directional centerlines, especially at road junction or splitting.

In polyline clustering based methods, each extracted road centerline is composed of a set of polyline segments. Achtert et al. (2006) and Lee et al.(2007) addressed that each polyline segment has to represent the overall movement of all sub-trajectories belonged to the cluster. There are three methods populated in extracting representative moving trend from a group of unidirectional moving trajectories. Tavares and Padilha (1995) considered the lengths of each participated polyline segments as weights for defining the orientation and placement of the resulting polyline. Lee et al. (2007) computed the average coordinates of endpoints of the representative polyline segment with respect to the average direction vector of the cluster. However, it did not clearly explain how to select the major axis of a cluster on which the sweep line was performed to obtain intersection points on other grouped polyline segments. Li et al. (2010) modified the approach proposed in (Lee et al., 2007) by considering mean values of length, coordinates of center point, and direction in the cluster. Compared to approaches used in (Lee et al., 2007; Li et al., 2010), the method proposed by Tavares and Padilha (1995) is more appropriate for the complex distribution of polyline segments in a cluster, such as partial overlapping, full overlapping, and none overlapping segments in similar directions. On the other hand, the resulting polyline segment is closer to its true location in the cluster than the other two methods because the centroid of all participated polyline segments in the cluster is calculated by taking their lengths as weights.

2.2.3. Grid-based Method

Grid-based road map generation methods quantize GPS points within the study area into a finite number of cells; and then adopt image-processing techniques to extract single-pixel road centerline from contiguous dense cells. Chen and Cheng (2008) addressed that exploiting morphological operations can help reduce effects of GPS measurement errors on extracted road network. In binary image construction step, coordinates of GPS points

(WGS84) are directly transformed to coordinates of image (row and column) by simply multiplying latitude and longitude by 10^5. Morphological operations, dilation and closing, are used to merge points and then smooth road boundaries in the binary image. To obtain road centerlines, the thinning algorithm is applied to skeletonize the smoothed road pixels. The digital road graph was generated by subtracting image coordinates of pixels by 10^5 to get their corresponding geographical coordinates in WGS84 system. A similar method was proposed by Shi et al. (2009a & 2009b), but the affine transformation is employed in the transformation of geographical coordinates of GPS points. Moreover, morphological erosion and dilation operations are implemented to smooth road boundaries and remove small noise in the binary image (Shi et al., 2009a).

Different from assigning binary numbers to cells based on the number of GPS points in (Shi et al., 2009a & 2009b), the value of each cell can also be presented by scale value that is proportionate to the length of intersected GPS trajectory within this unit square cell (Davies et al., 2006; Davies, 2009; Biagioni et al., 2011). After threshold cell values for identifying road features in binary image, a contour follower is employed to detect road boundaries. Finally, the coordinates of the road center point can be derived by averaging coordinates of nearest two points on the road boundaries.

The road inference based on the image processing techniques still faces the same challenges as the satellite image based road recognition:

(1) Jagged and undirectional extracted road centerlines (Chen and Cheng, 2008; Davies, 2009; Shi et al., 2009a & 2009b; Steiner and Leonhardt, 2011; Zhao et al., 2011);

(2) The size of cell in binary image affecting the recognition of nearby roads with similar direction and accuracy of extracted road centerlines (Zhao et al., 2011; Biagioni and Eriksson, 2012a);

(3) Inaccurate extracted road junctions (Davies, 2009; Zhao et al., 2011); and

(4) Difficulties in distinguishing road interchanges from road junctions (Davies, 2009). For instance, Zhao et al. (2011) performed raster-based road centerline extraction

by integrating map matching, point density analysis (Silverman, 1986) and the automatic vectorization of binary image (Mena, 2006).

The limitations of this kind of work are: only unidirectional road centerlines can be extracted due to the oversized raster cell in point density analysis; and using the uniformed road width in map-matching process is implausible for complex road geometries in a large region. In (Steiner and Leonhardt, 2011), Transverse Mercator projection on the ellipsoid is employed to transform geographical coordinates to planar coordinates, in order to achieve minimal distortions of road boundaries. The Watershed transform is then used to extract road centerlines. According to (Biagioni and Eriksson, 2012b), the two drawbacks of Watershed transformation are: the quality of extracted line segments is highly dependent on a set of initial local minima (Gonzalez and Woods, 2008) and dead-end road centerline cannot be extracted. In addition, grid-based spatial clustering methods are heavily dependent on the grid structure. The finer size of cell, the higher cost of constructing the bitmap; the coarser cell size thus reduces the quality of spatial clustering (Wang and Hamilton, 2010; Li et al., 2012). Accordingly, grid-based road map extraction methods are not considered in this thesis.

2.2.4. Hybrid Method

Based on the review of the aforementioned studies, it is clear that the basic connectivity problem in linking road center points or representative polyline segments has not been properly addressed. Worrall and Nebot (2007) linked cluster centers together based on a semantic road-knowledge based rule: one cluster center only has one forward link but could have more than two backward links when there is an intersection. This approach was only applied to the simple road geometry where all vehicle GPS trajectories were collected from several fixed paths (as discussed in Section 2.2.1). Liu et al. (2012) applied B-spline fitting to generate road centerline segment by segment, but ignored the continuity of extracted segmented road centerlines. Therefore, Cao and Krumm (2009) suggested that the connectivity properties of road network can be captured based on threshold values of distance and directional differences amongst GPS points, as well as the shortest distance to existing road centerlines. Nevertheless, this method cannot be applied to the GPS data from different regions because universal threshold values are

hardly estimated. To overcome this challenge, studies on topological and geometrical relationships amongst extracted road center points and preprocessed vehicle GPS trajectories were conducted by Limaa and Ferreiraa (2009), Li et al. (2012) , and Karagiorgou and Pfoser (2012).

Limaa and Ferreiraa (2009) proposed an approach to automatically extract road network by applying spatial queries based on the topological and geometrical relationship amongst center points and filtered GPS trajectories, without using a base map. In data preprocessing/filtering, Limma and Ferreiraa (2009) applied five filters to eliminate noise and to simplify GPS trajectories, with respect to the characteristics of GPS data (e.g., speed, horizontal dilution of precision, number of satellites, time interval, and data amount). In their experiment, road center point is derived by calculating geographic coordinates of centroid per each 5-m resolution raster cell, which has at least 20 GPS trajectories passing through. The spatial queries in terms of topological and geometrical relationships are performed to connect adjacent center points when at least one GPS trajectory passed them, with the radius of 1.5 m that likely covers one lane width. The road intersection is detected when a center point is an end point for at least three connections. The quantitative evaluation, comparing extracted road centerlines with the reference (commercial) map with 5-cm accuracy, concluded that average offset distance was of 1.43 m with respect to input GPS data of 15-m average accuracy and 77 percentages of extracted road center points matched to the reference map. This innovative approach only generates undirectional road centerline for each road. It is caused by the defined cell size of 5 m in the rasterization process, which covered roads of opposite directions. It is difficult to make an assumption for a uniform road width for all roads since the width of the road can vary along a road. For example, the width of the ramp is narrower than that of highway; and the width of the local road at turning is wider than that of straight road segment.

Li et al. (2012) developed an incremental road network extraction method that is specially designed for low frequency dynamic positioning data from China's National Commercial Vehicle Monitoring Platform (NCVMP). Unlike the early work proposed by Bruntrup et al., (2005), the spatial relationship amongst points and identified road-like

trajectories as well as the semantic relationship amongst any three successive points on the same trajectory are introduced into GPS point classification algorithm. The position of the identified road-like trajectory is gradually updated by calculating the weighted mean value of coordinates of recent matched GPS points and the pre-existing trajectory point. There are 18 parameters in terms of point density, distances and angles involved in this algorithm according to the characteristics of major roads in the study area. This method cannot tackle with the complex road geometric shape as road intersections and ramps so that manual editing was required for connecting extracted road segments.

Karagiorgou and Pfoser (2012) offered a better solution to extract road network in consideration of the connectivity of underlying road network embedded in the vehicle GPS trajectories. This automatic road network recognition algorithm consists of two steps. Given speed and direction difference threshold values, turning points are identified by scanning point by point on every GPS trajectory. Intersection point is obtained from each turning-point cluster grouped based on distance and turning type. And then, sub-trajectories between any two intersection points are merged together to generate positions of representative line segment with associate spatial extent. The representative line segment can be refined by adding more nearby sub-trajectories, which intersect with the spatial extent and if their parent trajectories pass one of two intersection points. However, inferring bi-directional road centerlines amongst intersections is infeasible since all turning points were merged into one intersection point regardless of turning directions.

2.2.5. Discussion

The aforementioned studies reveal various kinds of efforts that have been devoted to extracting proper road information from different GPS data sources by utilizing different spatial clustering techniques. Following by the brief summarization of individual method, its limitations, strengths, and similarity to other methods are discussed. Table 2.2 summarizes the literature on road map generation in terms of GPS data source and characteristics, study area, processing unit, result type, and evaluation method.

Although all these research efforts are relevant to this thesis research, only six out of the 18 papers focus on extraction of bi-directional road centerlines. The bi-directional route system, modelling each direction of travel as a separate alignment, was instituted by

the New York State Thruway Authority in 2001. Herman (2002) summarized the benefits of using bi-directional road centerline such as: better matching with other data layers; providing a truer presentation of the highway network, and analyzing the road information more effectively (e.g., lane closures pertaining to certain direction of travel).

The traditional unidirectional road centerline cannot respect one-way streets and roads of opposite directions simultaneously. Consequently, studies conducted by (Shi et al., 2009a & 2009b; Limaa and Ferreiraa, 2009; Guo et al., 2010; Zhao et al., 2011; Karagiorgou and Pfoser, 2012; Liu et al., 2012) are restricted to local roads in urban areas because the assumed road width cannot be suitable for both highway and local roads. In contrast, Zhang et al. (2010) provided a better solution for extracting bi-directional road centerlines with acceptable positional accuracy. Nevertheless, the quality of resulted roads is easily affected by errors in the base map due to the limitation of map-matching method. In addition, Davies et al. (2006) suggested that the direction be manually appended as metadata to the undirectional road centerline. However, grid-based methods are not considered in this research thesis due to the aforementioned discussion in Section 2.2.3.

As noted in the "Evaluation" column, most of the literature provides the visual inspection by overlaying the generated road map with a satellite image or digital map instead of the quantitative evaluation. In order to produce a digital road map meeting with practical requirements such as route direction and mapping, the quality of road map in terms of positional accuracy must be provided (Willrich, 2002). Only few studies provided the quantitative evaluation to conclude the positional accuracy of the generated road centerline with respect to the relative accurate commercial road map. Zhang et al. (2010) compared the extracted bi-directional road centerlines to digital road map from TeleAtlas dataset of 2 to 10 m accuracy. The research concluded that 27.4% of extracted road centerlines are within 2 m, 61.7% are within 5 m, and 73.9% are within 7 m. Limaa and Ferreiraa (2009) concluded that 77% of extracted undirectional road centerlines matched with the commercial digital road map of 5-cm accuracy provided by InfoPortugal.

As reviewed in Section 2.2, the main challenge in existing methods is how to generate a road centerline capturing both accurate geometry and connectivity. Some factors have not yet been fully investigated. For example, reducing data without affecting underlying road network extraction; spatial clustering algorithms distinguishing nearby parallel roads and road splits are subject to various parameters; topological and geometrical relationships for generating accurate road junctions; and constructing bi-directional road centerlines. Therefore, it is still desirable that a practical approach integrates all the factors together to automatically extracting road network without referencing to a base map or image. Moreover, none of the aforementioned studies performed quantitative evaluation by comparing the results with the actual ground truth maps (geometric road alignment data). Table 2.3 provides a summary of available data processing techniques relevant to this thesis research. This thesis research attempts to fill the current gap by proposing a new methodology, integrating point- and polyline-based approaches, to iteratively infer road network from smartphone GPS trajectories.

Table 2.2: Road map generation literature

Literature	Purpose	Data Source (Accuracy & Sampling Rate)	Study Area	Processing Unit	Map-matching	Result Type	Evaluation
Li et al. (2012)	Road map updating	China NCVMP[5](5 m & 33 sec)	City-wide	Hybrid of Points & Polylines	No	Bi-directional[6]	Visual Inspection
Cao and Krumm (2009)	Routable graph generation	55 Shuttles with GPS loggers[7] (unknown; 1 sec)	Campus-wide	Points	No	Bi-directional	Visual Inspection
Wang et al. (2011)	Routable graph generation	Unknown GPS loggers (unknown)	Town	Points	No	Bi-directional	N/A
Edelkamp and Schrodl, (2003); Schroedl et al., (2004)	Road Centerline Extraction	Unknown GPS loggers (unknown; 1-4 sec)	Small area[8]	Points	Yes	Bi-directional	N/A
Zhang et al. (2010)	Road map updating	OpenStreetMap (6-10 m; unknown)	Small area[9]	Points	Yes	Bi-directional	Quantitative Evaluation[10]
Limaa and Ferreiraa (2009)	Road Centerline Extraction	Unknown GPS loggers (15 m; 1 sec)	City-wide	Hybrid of Points & Polylines	No	Undirectional	Visual Inspection & Quantitative Evaluation[11]
Worrall and Nebot (2007)	Routable graph generation	Unknown GPS loggers on five mining trucks (unknown)	Mining site	Points	No	Bi-directional	N/A

[5] China National Commercial Vehicle Monitoring Platform
[6] Road of opposite directions which can respect one-way streets.
[7] RoyalTek RBT-2300 with SiRF Satr III chipset and WAAS enabled.
[8] It contains a combined length of approx. 20 km of urban and freeway roads.
[9] It contains one highway interchange area and one urban intersection area.
[10] Quantitative evaluation with respect to standard road map from TeleAtlas dataset with accuracy of 2 to 10 m.
[11] Quantitative evaluation with respect to vector road map provided by mapping company (InfoPortugal), with accuracy of 5 cm; Visual evaluation is performed by overlapping with Google Map.

Literature	Purpose	Data Source (Accuracy & Sampling Rate)	Study Area	Processing Unit	Map-matching	Result Type	Evaluation
Guo et al. (2010)	Trajectories Clustering	Unknown GPS loggers (10-20 m; 30 sec)	City-wide	Points	No	Undirectional	N/A
Lee et al. (2007); Li et al. (2010)	Trajectories Clustering	Hurricane track and animal movement (unknown)	Unknown	Polylines	No	Undirectional	N/A
Liu et al. (2012)	Road Centerline Extraction	Unknown GPS loggers on taxis (7 m; 16-61 sec)	Small area[12]	Polylines	No	Undirectional	Visual Inspection & Quantitative Evaluation[9]13
Karagiorgou and Pfoser, (2012)	Road Centerline Extraction	Unknown GPS loggers on taxis (unknown)	Small urban area	Hybrid of Points & Polylines	No	Undirectional	Visual Inspection & Quantitative Evaluation[14]
Ahmed and Wenk (2012)	Road Centerline Extraction	Unknown GPS loggers (unknown; 30 sec)	Municipal area of Berlin	Polylines	Yes	Undirectional	Visual Inspection
Chen and Cheng (2008)	Road Centerline Extraction	Garmin GPS25-LVS (5-15 m[15]; 1 sec)	Certain Roads	Pixels	No	Undirectional	Visual Inspection
Shi et al. (2009a)	Road Centerline Extraction	Unknown GPS loggers (5 m; unknown)	City-wide	Pixels	No	Undirectional	Visual Inspection
Shi et al. (2009b)	Road Centerline Extraction	Unknown GPS loggers on probe vehicle (unknown)	Campus-wide	Pixels	No	Undirectional	Visual Inspection

[12] A selected area of 14.5 km x 14 km in urban area
[13] Quantitative evaluating the accuracy of recognized roads in comparison to OpenStreetMap.
[14] Evaluation of connectivity & shortest-path similiarity to existing map
[15] http://www.wildsong.biz/index.php?title=GPS_receivers; ftp://ftp.tapr.org/gps/garmin/Spek25lp.pdf

Literature	Purpose	Data Source (Accuracy & Sampling Rate)	Study Area	Processing Unit	Map-matching	Result Type	Evaluation
Zhao et al. (2011)	Road map updating	China TTIC[16] (unknown; several mins)	Certain Roads	Pixels	Yes	Undirectional	Visual Inspection
Steiner and Leonhardt (2011)	Road Centerline Extraction	Unknown GPS loggers on Taxis (1.2-1.9 m; 15-90 sec)	Small urban area	Pixels	No	Undirectional	Visual Inspection
Biagioni et al. (2011); Davies et al. (2006)	Road map updating	Unknown GPS loggers (4.25-8.5 m; unknown)	County-wide	Pixels	No	Directed graph[17]	Visual Inspection[18]

[16] Transportation and Telecommunication Information Center
[17] Undirected road graph with manually appended direction metadata
[18] Compared with digital road map created by UK Ordnance Survey

Table 2.3: Summary of data processing techniques

Type of Classification	Literature	Data Clarification	Data Classification	Data Extraction	Centerline Inference
Point-based	Edelkamp and Schrodl (2003; Schroedl et al., (2004)	Map-matching	Partitional spatial clustering based on minimum distance between points.	Iteratively averaging clustered points to get cluster center, if new point is added.	B-spline fitting based on user-selected control points
	Worrall and Nebot (2007)	N/A	Simple clustering based on similar position and similar direction.	Averaging clustered points to get cluster center.	Linking cluster points based distance function and sematic rule.
	Cao and Krumm, (2009)	Simulated attraction forces	N/A	Gradually merging points to existing point without replacement.	N/A
	Guo et al. (2010)	Moving window smoothing	Hierarchical spatial clustering based on minimum distance between points.	Calculating centroid of clustered points	N/A
	Zhang et al. (2010)	Split GPS trajectory by distance and speed	Map-matching & Partitional spatial clustering (fuzzy c-means)	Calculating centroids of clustered points, weighted by degree of belonging to the cluster.	Linking cluster points on each road.
	Wang et al. (2011)	simulated attraction forces	Clustering points attached with a weight.	Gradually merging points to existing point without replacement.	N/A
Polyline-based	Lee et al. (2007; Li et al., (2010)	Approximate trajectory partitioning.	Density-based Polyline Clustering	Extracting representative trajectory from grouped sub-trajectories	N/A
	Liu et al. (2012)	Eliminating noise based on threshold values of direction difference, acceleration, and/or velocity change between two consecutive points.	Basic polyline segment clustering based on geographical distance and direction change.	Clustering refinement for road splitting.	B-spline fitting & Shape-ware fitting for choose control points

Method Classification	Method	Data Clarification	Data Classification	Data Extraction	Centerline Inference
Hybrid	Limaa and Ferreiraa (2009)	Eliminating noise based on threshold values of speed, HDOP, and the number of satellites.	Grid-based method quantized GPS points within the study area into a finite number of cells.	Calculate centroid in each point-occupied cell.	Linking centroids based on topological and geometrical relationships to GPS trajectories.
	Li et al. (2012)	Eliminating noise based on threshold values of speed and direction change.	Clustering points based on their spatial relationship to same road trajectory and semantic relationship amongst three consecutive points.	Weighted mean value of coordinates of recent matched GPS points and the pre-existing GPS point.	Linking cluster points based on thresholds of distance and direction change.
	Karagiorgou and Pfoser (2012)	N/A	Clustering turns to identify Intersection points	Extracted common GPS trajectories passing through at least two intersection points	Iteratively update the road centerline between two intersections until no more new trajectory

Chapter 3. Methodologies

Chapter 2 surveyed the state of the art on road network extraction from satellite image and vehicle GPS trajectories, and summarized crucial factors that were not fully taken into account to automatically generate a road network capturing the accurate road geometry and connectivity. The methodology of automatic extraction of road network in this thesis research identifies GPS trajectories that restricted to the road network without using map-matching technology; therefore, it faces following dilemma. First, positioning data is collected regardless of where the smartphone user is (e.g. driving on roads, idling (in parking lots or traffic jams), waiting at traffic lights, or inside buildings), so it is difficult to completely filter out such point clouds from raw data. Second, the distribution of positioning data of moving vehicles could be normal, scattered or multimodal within a certain width around a road centerline. However, it is implausible to utilize a fixed value of road width to cluster positioning points belonging to the same road segment, especially at a road split or merge.

Taking these uncertainties into account, the objective of this chapter is to overcome five challenges in similar studies, including:

1) Data reduction without affecting underlying road network extraction;
2) Distinguishing GPS trajectories on nearby parallel roads;
3) Separating GPS trajectories on roads that have splits of a narrow angle;
4) Constructing bi-directional road centerlines, and
5) Building topological connectivity at road junctions.

Section 3.1 outlines the general procedures of extracting the road network from smartphone GPS trajectories of moving vehicles. Section 3.2 describes the pre-processing of raw data. The detailed explanations of every step and techniques adapted to this methodology are presented in subsequent sections.

3.1. Overall Workflow

The overall workflow of automatic extraction of road networks data from smartphone GPS trajectories is shown in Fig. 3.1. The basic idea behind this new methodology is:

restructuring GPS trajectories on each road so as to ideally obtain at least one new GPS trajectory in each lane; grouping reconstructed GPS trajectories on the same road segment into one cluster and then merging them into one polyline segment representing the road centerline; and utilizing the topological relationships amongst polyline segments to construct the final road network data.

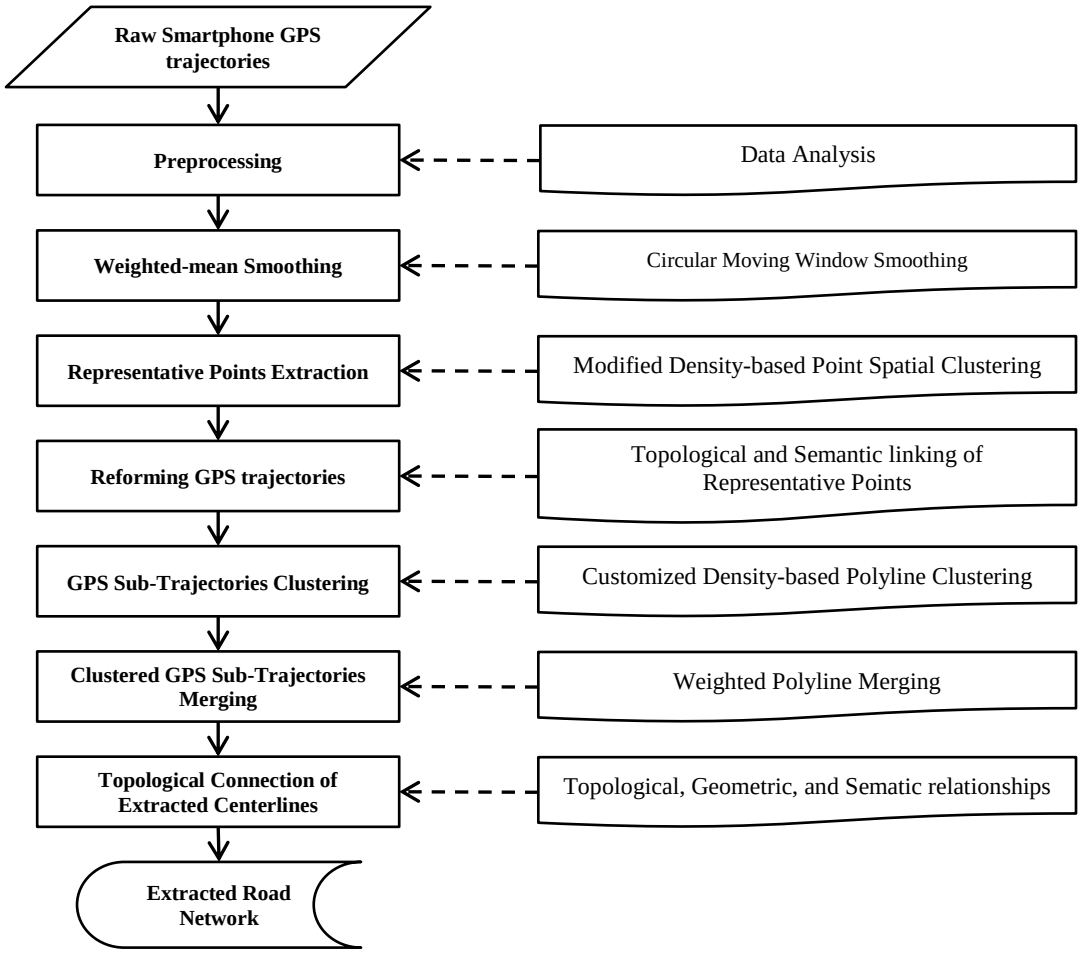


Figure 3.1: Overall workflow of automatic extraction of road network.

Given a set of smartphone GPS trajectories, the preprocessing step and weighted-mean smoothing algorithm (see Fig. 3.2 (a)) were applied to eliminate extraneous and duplicated data and replace inaccurate data. The modified density-based point clustering method was applied to extract representative points for each lane on the road, as illustrated in Fig. 3.2 (b). Next, representative points belonging to the same lane were

connected based on their topological relationships and directions (see Fig. 3.2 (c)), in order to remain faithful to the underlying road network geometry. Then, the road centerlines were derived by applying the customized density-based polyline segment clustering method (Fig. 3.2 (d)) to merge those reformed GPS trajectories (see Fig. 3.2 (e)) sharing the same geometric attributes on the road. Last, road centerlines were topologically connected together to generate a completed road network.

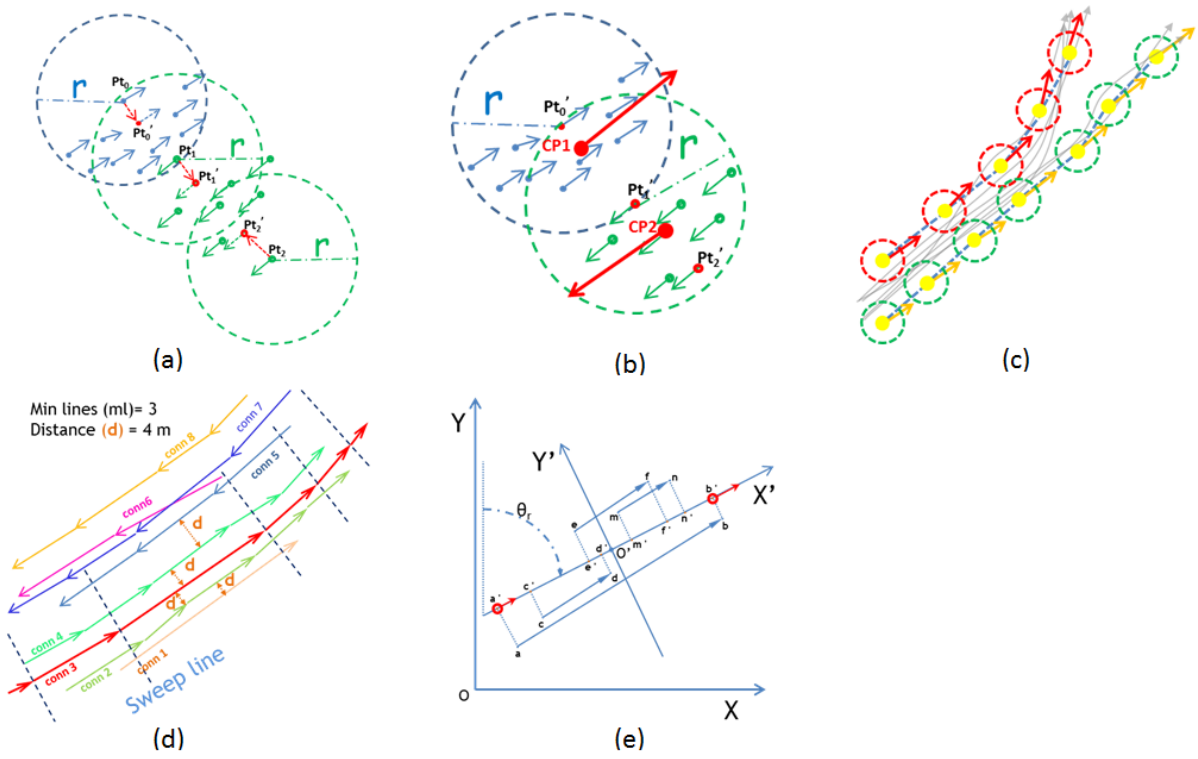


Figure 3.2: Demonstration of road centerline extraction algorithms: (a) weighted-mean smoothing (b) Representative point extraction (c) Linking representative points (d) Polyline spatial clustering (e) Merging clustered polylines.

3.2. Positioning Data Preprocessing

In order to improving the quality of GPS trajectories, road extraction method begins with the data preprocessing step. It is performed to reduce the size of input data without affecting the underlying road network geometry by: 1) filtering out the off-road point clouds, 2) splitting a daily GPS trajectory into a set of individual trips, 3) converging GPS positioning points of each moving vehicle towards the middle of a road by applying the weighted moving circular window smoothing, and 4) eliminating the duplicated positioning points.

Off-road positioning point clouds could be removed by applying the threshold value of speed (1.904 m/s or 6.854 km/hr obtained through data analysis in Section 4.4). It helped to remove noise positioning points caused by slow moving vehicles at traffic lights or traffic congestions as shown in Fig 3.3.

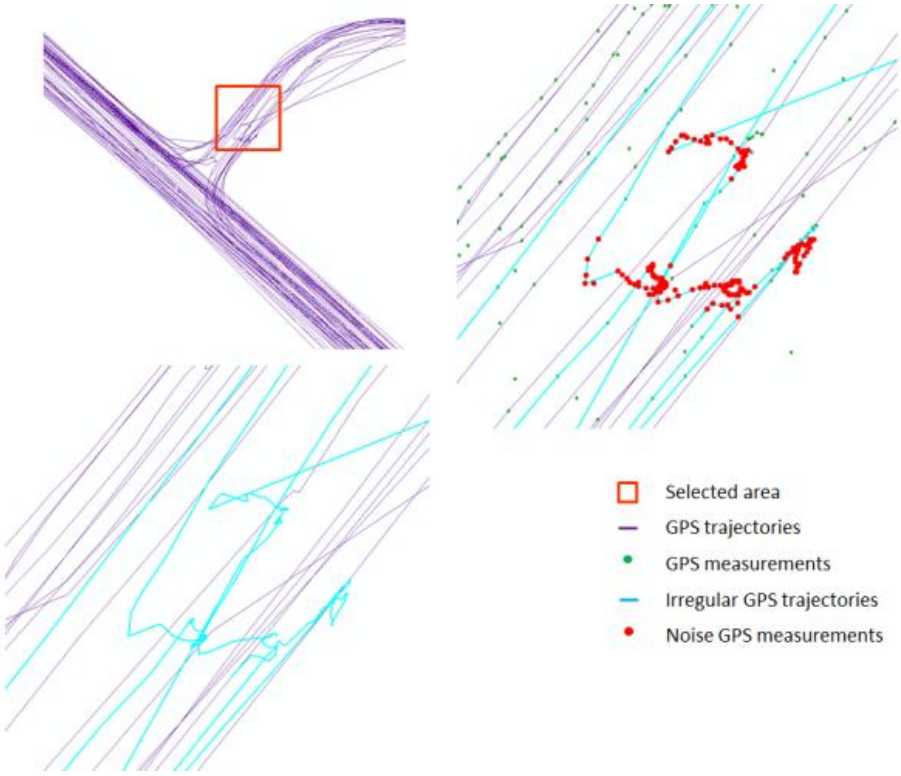


Figure 3.3: Noise positioning points in slow-moving traffic flow.

Unreasonable trajectories may be generated amongst different trips made by the same smartphone users in a same day, as highlighted (in red color) in Fig 3.4. For

example, a connection between a pair of consecutive GPS points is off from the actual road. The road network data is inferred by connecting endpoints of extracted road centerlines based on their topological relationships to GPS trajectories revealing the underlying road network geometry. Therefore, a daily trip travelled by a smartphone user must be split into a set of individual on-road GPS trajectories. In this regard, every trip is subject to two checks, change of driving direction and distance between consecutive positioning points. Each trip is split into discrete GPS trajectories whenever a gap between any two time-stamped positioning points is larger than distance threshold or the change in their moving direction is over direction threshold (100 m and 11˚ obtained through data analysis in Section 4.4).

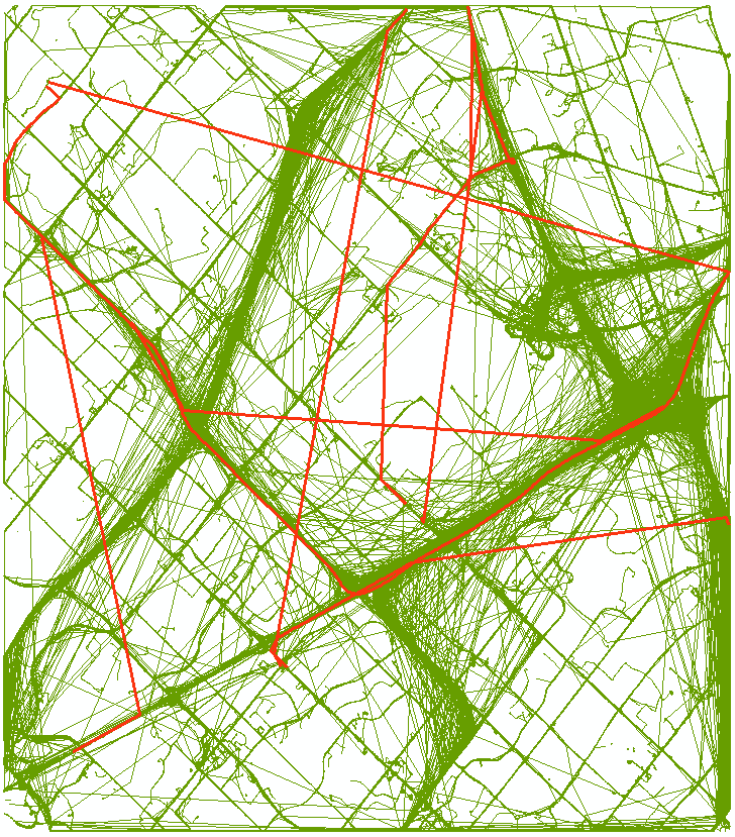


Figure 3.4: Unreasonable connections within raw GPS trajectories.

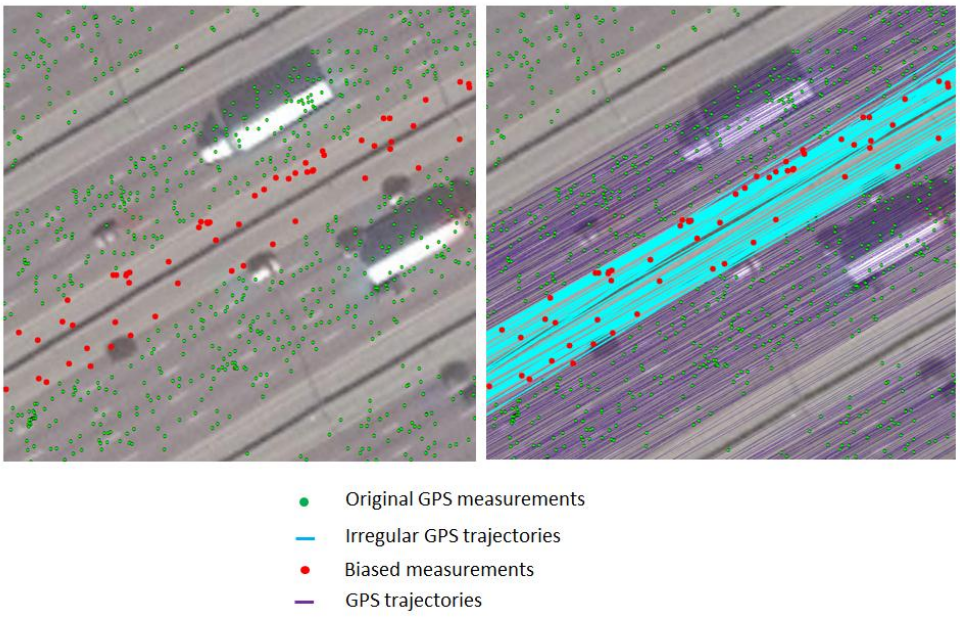


Figure 3.5: Noise positioning points on shoulders due to the biased smartphone GPS measurement.

The distribution of positioning points of moving vehicles could be normal, scattered or multimodal within a certain width around a road centerline, due to the biased smartphone GPS measurements. Some GPS trajectories do not exactly follow the road segment, especially at a road split or merge. Fig 3.5 shows that many positioning points (red-color biased GPS measurements) were located at the median section between nearby divided roadways or road shoulders of an undivided road. To overcome these inaccurate measurements, a weighted moving circular window smoothing is proposed to relocate positioning points towards the center of the road based on the similar algorithm adopted by Guo et al. (2010). In that algorithm, the new position of each positioning point was obtained by averaging all points within its 30-meter circle. However, the algorithm in this thesis research is different in that it uses 4-meter circular window and assigns different weight to the positioning point in terms of its accuracy.

A 4-meter circular window is placed on each positioning points in order to cover the lane width ranges between 3.5 and 3.7 m (TAC, 1999). The new location of the positioning point is changed to the weighted mean of all positioning points in the similar direction (cos(directional change between any two points) > 0) within the circular

window. Fig. 3.6(a) illustrates the performance of smoothing by using Eq. 3.1. As shown in Fig. 3.6 (b), positioning points of opposite directions on parallel roads can be differentiated by using the cosine of the angular difference between two moving directions. It is inspired from the GPS trajectory clarification algorithm adopted by (Cao and Krumm, 2009). In their algorithm, the cosine of the difference between two directions was used to differentiate polyline segments of opposite directions. However, the algorithm adapted in this thesis research focuses on converging positioning points in similar direction toward the middle of road. For example, any neighbor point Pt_j, within the 4-meter circle and moving in similar direction as Pt_0 and cos(direction of Pt_j – direction of Pt_0) > 0, will be used to calculate the new position of Pt_0 by using Eq. 3.2. The moving direction of Pt_0' remains the same as that of Pt_0.

Fig. 3.6 (c) presents locations of a smoothed point by leveraging weighted mean or simple mean. Making use of the simple mean only brings the positioning point to the geometric center of a 4-m cluster. In contrast, taking the accuracy of each point as a weight brings the positioning point close to its neighbor points with high accuracy. In case a positioning point does not have any other points within its circular window, it will be discarded as an outlier point that has no contribution to extracting road centerlines. At the end, the duplicated smoothed positioning points are eliminated. For example, if smoothed positioning points, Pt_1' and Pt_2', are overlapped at the same location in Fig. 3.6 (a), Pt_2' will be removed and the direction of Pt_1' will be substituted by the mean direction of these two points.

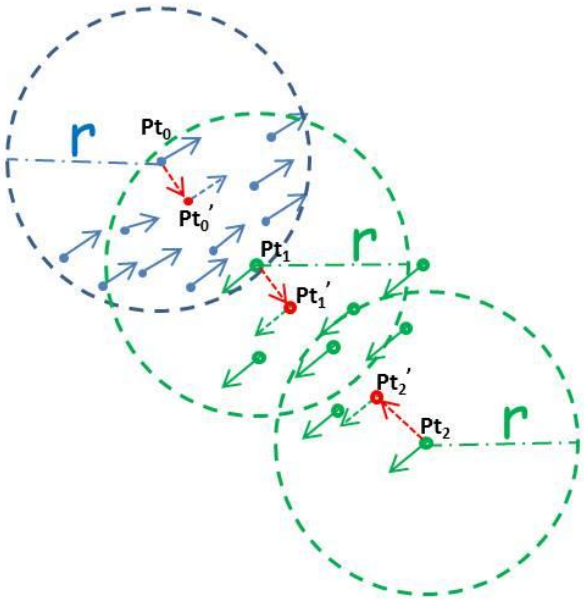
$$if\ \cos\left(direction\ of\ Pt_j - direction\ of\ Pt_0\right) > 0;\ \ they\ are\ in\ similar\ direction$$

$$Pt_0'X = \frac{\Sigma_{j=0}^{n}\left(Pt_jX \cdot \omega_j\right)}{\Sigma_{j=0}^{n}\left(\omega_j\right)}$$

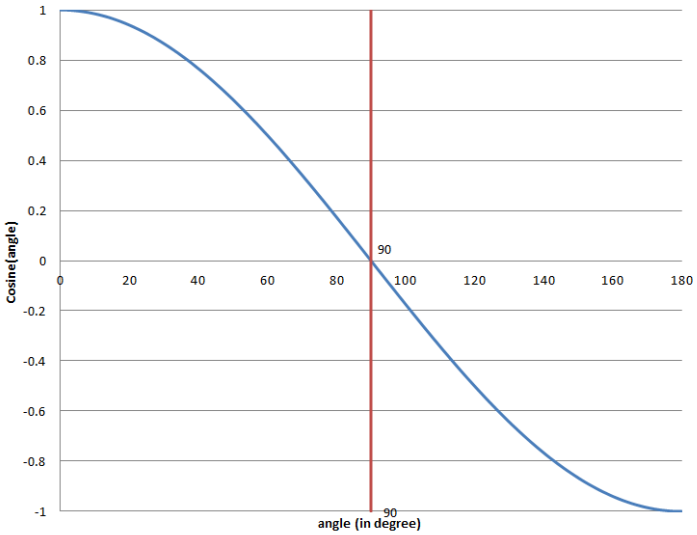
$$Pt_0'Y = \frac{\Sigma_{j=0}^{n}\left(Pt_jY \cdot \omega_j\right)}{\Sigma_{j=0}^{n}\left(\omega_j\right)} \qquad \text{Equation 3.1}$$

$$Pt_0'accuracy = \frac{n}{\Sigma_{j=0}^{n}\left(\omega_j\right)}$$

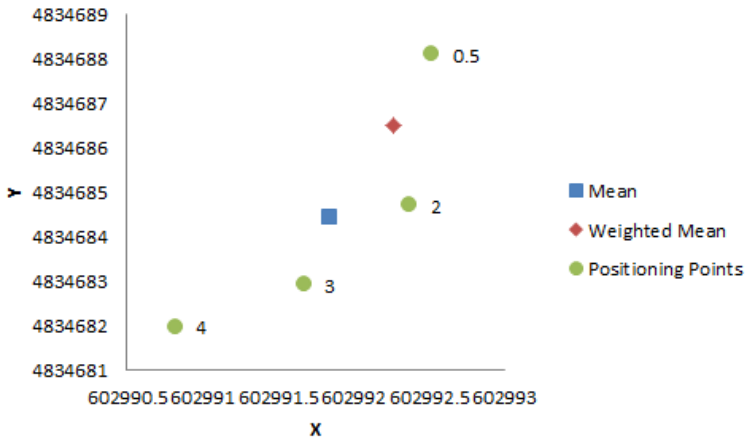
where, weight $\omega = \frac{1}{Pt's\ accuracy}; j = 0,1,2,\ldots,n$


r
Pt0
Pt0'
Pt1
r
Pt1'
Pt2'
Pt2
r

(a)


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Cosine(angle)
0
20
40
60
80
90
100
120
140
160
180
90
angle (in degree)

(b)



(c)

Figure 3.6: Illustration of (a) moving window smoothing algorithm (b) cosine curve of directional change (c) difference between weighted mean and simple mean.

3.3. Representative Point Extraction

After the preprocessing step, a modified density-based point clustering method is adopted to reduce the data size by extracting a smaller set of new positioning points as representative of smoothed points, without affecting the underlying road geometry. This step is similar to the data reduction method adopted by Guo et al. (2010), but takes into consideration the moving direction of individual positioning point and accuracy of individual positioning point as a weight. The smoothed positioning points within a 4-meter (definition of 4-m refers to Section 3.2) cluster must satisfy six properties:

1) A smoothed positioning point can only be represented by one preliminary representative point;
2) One preliminary representative point represents at least one smoothed positioning point;
3) If one smoothed positioning point falls inside of a 4-m circle of any positioning point that belongs a cluster of one preliminary representative point and is within 4-meter buffer of the preliminary representative point, it is also represented by that preliminary representative point;

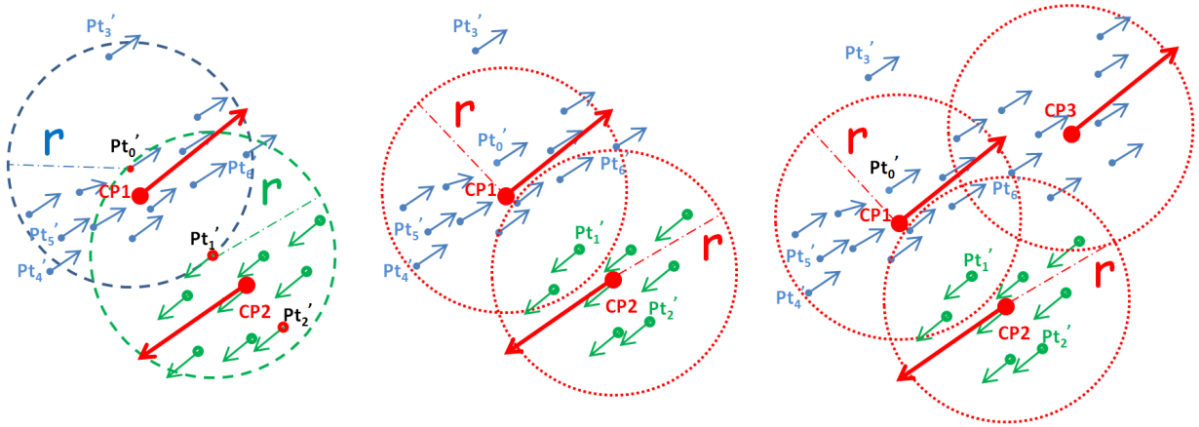
4) If one smoothed positioning point is already represented by another preliminary representative point but is closer to the current preliminary representative point, it thus belongs to the current cluster;
5) If one smoothed positioning point is out of 4-meter circle of current preliminary representative point, it is removed from its current cluster; and
6) All smoothed positioning points within a cluster must have similar directions.

$$Pt_0' direct = \frac{\Sigma_{j=0}^{n}\left(Pt_j direct \cdot \omega_j\right)}{\Sigma_{j=0}^{n}\left(\omega_j\right)} \qquad \text{Equation 3.2}$$

$$\text{where, weight } \omega = \frac{1}{Pt's\ accuracy};\ j = 0,1,2,\ldots,n$$

A more formal description is shown in Algorithm 1 in Appendix A. It consists of two steps, preliminary representative point extraction (Step 1) and refinement (Step 2). Figs. 3.7 and 3.8 illustrate core procedures of extracting representative points in Step 1 and 2. Fig.3.7 (a) demonstrated the 1st and the 6th properties that all nearby unrepresented smoothed positioning points within the 4-meter buffer of Pt_0' are stored together with Pt_0' into one cluster (line 8 to 16 of algorithm 1) except Pt_1', which is in opposite direction. A preliminary representative point CP1 is calculated by using Eqs. 3.1 and 3.2 (in line 18 of algorithm 1). Fig.3.7 (b) exemplifies the 5th property that Pt_3' is outside of the 4-meter circle of CP1 and no longer belongs to the cluster of CP1 (line 23 to 25) after applying Eqs. 3.2 and 3.3. Meanwhile, Pt_4' satisfies the 3rd property that it is within 4-meter circle of Pt_5' where Pt5′ is represented by CP1, so that it is also inside of CP1's cluster (line 27 to 34). Fig.3.7 (c) explains the 4th property (line 20 to 22 of algorithm 1) that Pt_6' is previously grouped into CP1's cluster but is closer to the new preliminary representative point CP3, so that Pt_6' is being represented by CP3. Step 1 is repeatedly implemented until all smoothed positioning points are represented and stored in a temporary dictionary[19]: tempC = {CP1, CP2 …, CPn | n < D}, where each cluster is a set of represented smoothed positioning points CP1 = {Pt_j | 0 < j < D}, D denotes the number of input smoothed positioning points.

[19] Python dictionary definition(http://www.tutorialspoint.com/python/python_dictionary.htm)



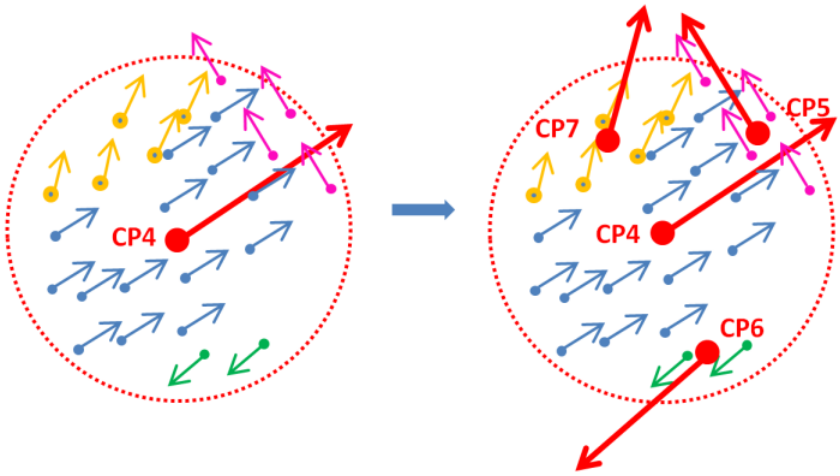
(a) (b) (c)

Figure 3.7: Illustration of extracting preliminary representative positioning point; arrows (in different colors) represent the different moving directions.

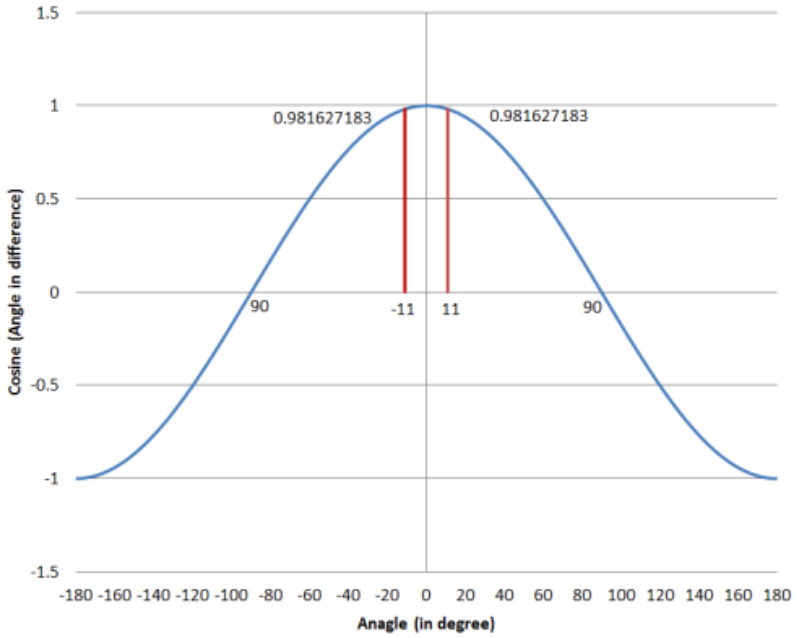
In Step 2, the temporary clusters in tempC are further refined by checking if there are smoothed positioning points moving in different direction to the preliminary representative point in the cluster. If yes, it further splits the cluster into smaller clusters. Fig. 3.8 (a) and (b) illustrate the implementation of cluster refinement based on the cosine of angle difference in Fig. 3.8 (c). For example, the preliminary representative point CP4 representing four sets of smoothed positioning points (in colors of blue, yellow, green, and lavender) can be divided into following sub-clusters:

1) If cos(direct of Pt – direct of CP4) ∈ [cos11˚, cos0˚], point Pt in the blue-color point set is in a direction similar to CP4 on the same road;
2) If cos(direct of Pt – direct of CP4) ∈ (cos90˚, cos11˚), point Pt in the yellow-color point set is in a direction different from that of CP4 but may be on the same road (at road split or turning);
3) If cos(direct of Pt – direct of CP4) < 0, point Pt in green-color point set is on the road of opposite direction; and
4) Point Pt is moving in perpendicular direction to that of CP4 (at road intersection or highway interchange), if the cosine of angle difference is zero.

Finally, new representative points (CP4, CP5, CP6, and CP7) are calculated for each sub-cluster by using Eq. 3.1 and 3.2.



(a) (b)



(c)

Figure 3.8: Illustration of exacting final representative point.

3.4. Trajectory Reconstruction

Section 3.3 topologically simplifies GPS trajectories by extracting a smaller set of representatives of original GPS positioning points. In conventional point-based methods, the final road centerline could be generated by leveraging one of four approaches,

including: simply linking cluster centers on the same road together based on geometric relationships (threshold values of direction difference and distance) amongst them; integrating semantic rules and geometric relationships; B-spline approximation based on control points; or spatial queries in terms of topological and geometric relationships between cluster centers and original GPS trajectories. However, as discussed in Section 2.2, they are restricted by the fixed value of road width that is applied to cluster positioning points belonging to the same road segment, especially at a road split or merge.

In order to clarify GPS trajectories on roads of complex geometry, this section aims at restructuring the GPS trajectories on each road so as to ideally obtain at least one new GPS trajectory in each lane. Therefore, the boundary of the road segment (road width) can be outlined by the evenly distribution of reconstructed GPS trajectories along the road. As shown in Fig. 3.9, representative points belonging to the same lane were connected based on their topological relationships to GPS trajectories and semantic relations, in order to remain faithful to the underlying road network geometry. Blue dash-line denotes the new reformed GPS trajectory. Grey solid lines represent the preprocessed GPS trajectories. 4-meter dash-line circle (in color of red and green) of each yellow-color representative positioning point covers the lane width ranges from 3.5 to 3.7 m.

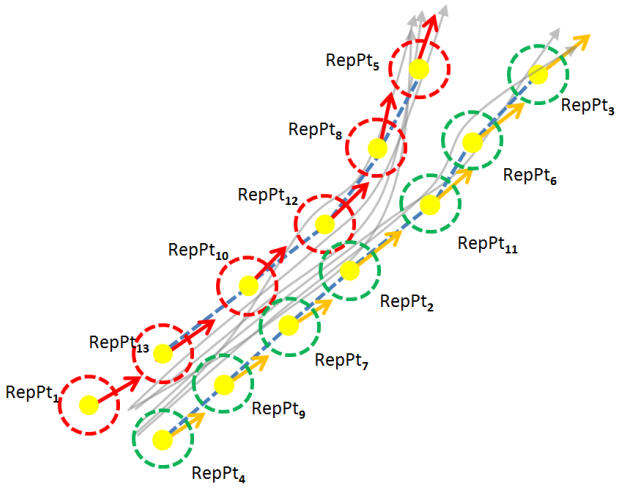


Figure 3.9: Illustration of GPS Trajectory Reconstruction.

The overall logic flowchart of reforming GPS trajectories is represented in Fig. 3.10. Starting with any one of unconnected representative positioning points, its surrounding representative positioning points are checked based on three conditions:

1) Distance to the current representative positioning point must be within 50 m (see Section 5.1 for searching radius);
2) At least one preprocessed GPS trajectory passing through 4-m buffers of surrounding and current representative points; and
3) The difference of the directions of surrounding and current representative points is smaller than 11˚.

If a surrounding representative point satisfies these prerequisites, it becomes a candidate point for constructing new connection from or to the current representative positioning point. If a representative positioning point is selected as the current representative positioning point (e.g. $RepPt_1$ in Fig. 3.9) but has no candidate point, the next representative positioning point ($RepPt_2$) becomes the available current point. The GPS trajectory reconstruction algorithm detects optimal succeeding and preceding points ($RepPt_{11}$ and $RepPt_7$) from candidates of $RepPt_2$ to construct new connections along with their moving directions. New polyline segments (from $RepPt_2$ to $RepPt_{11}$ and from $RepPt_7$ to $RepPt_2$) are generated as shown in Fig.3.9. The algorithm is repeatedly implemented until all representative positioning points are connected.

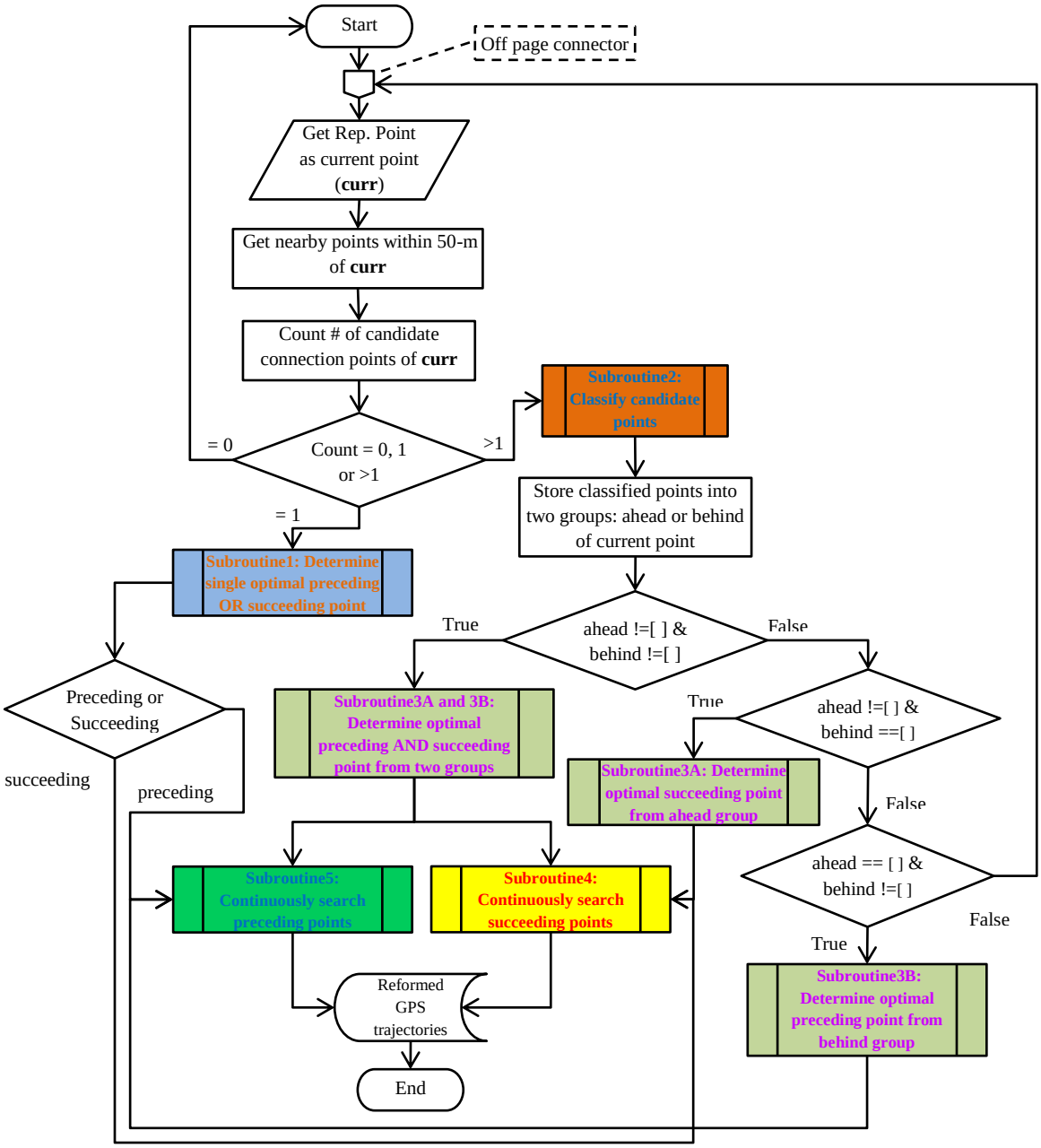


Figure 3.10: Main logic flowchart of reconstructing GPS trajectories.

The GPS trajectory reconstruction algorithm consists of five core subroutines (shown in Fig. 3.10) that are repeatedly utilized to search optimal preceding and succeeding points for reforming GPS trajectories based on their moving directions. Given a set of candidate points of a current representative positioning point, Subroutine 1 is used to determine the single preceding or succeeding point of the current point as the

initial point, if there is only one candidate point; if there is more than one candidate points are found, Subroutine 2 classifies them into two groups: ahead and behind of current point with respect to their moving directions; Subroutine 3 determines the optimal preceding and succeeding points from each group as initial points; Subroutine 4 and 5 are implemented to continuously search and determine rest of connecting points from initial points, until all representative points on the common direction are connected. A more formal description of this algorithm is shown in Algorithm 2 in Appendix A.

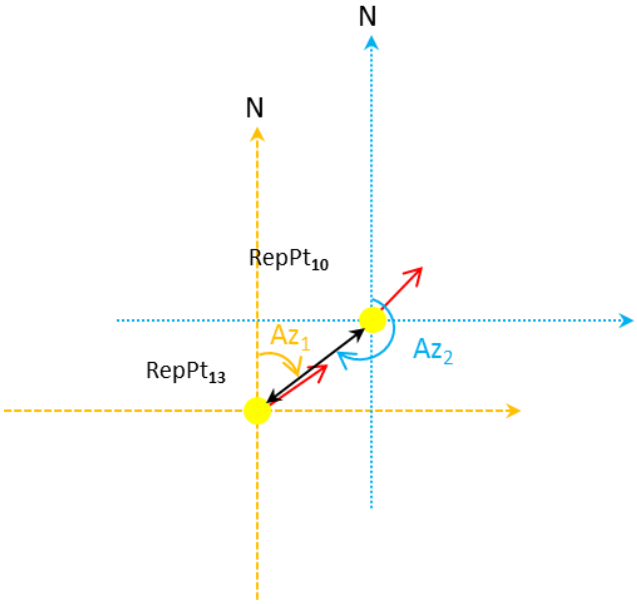


Figure 3.11: Comparison of azimuth and average direction of points.

Subroutine 1 is applied in case there is only one candidate point of the current representative positioning point was found. For example, if $RepPt_{13}$ in Fig. 3.11 is taken as the current point, only $RepPt_{10}$ could be the candidate to which a new connection is constructed from $RepPt_{13}$, because there are two preprocessed GPS trajectories passing through their buffers. In order to construct a new connection following the moving directions of these two points, two azimuths[20] (Az) are calculated by using Eq.3.3: Az_1 is from $RepPt_{13}$ to $RepPt_{10}$ and Az_2 is from $RepPt_{10}$ to $RepPt_{13}$, as shown in Fig.3.11. If the

[20] Azimuth is the horizontal angle measured clockwise from north, in plane surveying (Charles D. Ghilani, Paul R.Wolf, 2002)

angular difference between Az_1 and the average of directions of two points ($RepPt_{13}$ and $RepPt_{10}$) is not over 11˚, a new connection from $RepPt_{13}$ to $RepPt_{10}$ is constructed and $RepPt_{10}$ is recorded as the succeeding point of $RepPt_{13}$. If $RepPt_{10}$ is selected as the current point that has $RepPt_{13}$ as the unique candidate point, the connection is still generated from $RepPt_{13}$ to $RepPt_{10}$ except that $RepPt_{13}$ becomes the preceding point of $RepPt_{10}$. The reformed connection is recorded and its endpoints are marked as "connected representative points", in order to avoid the duplication in later process. The logic flowchart of Subroutine 1 is represented in Fig. 3.12 for reader's easy understanding.

$$Az_{AB} = tan^{-1}\left(\frac{\Delta X}{\Delta Y}\right) + C \qquad \text{Equation 3.3}$$

where, C places the azimuth in the proper quadrant; $\Delta X = X_B - X_A;\ \Delta Y = Y_B - Y_A$

$\Delta X > 0$ and $\Delta Y > 0$	C=0˚
$\Delta Y < 0$	C=180˚
$\Delta X < 0$ and $\Delta Y > 0$	C=360˚

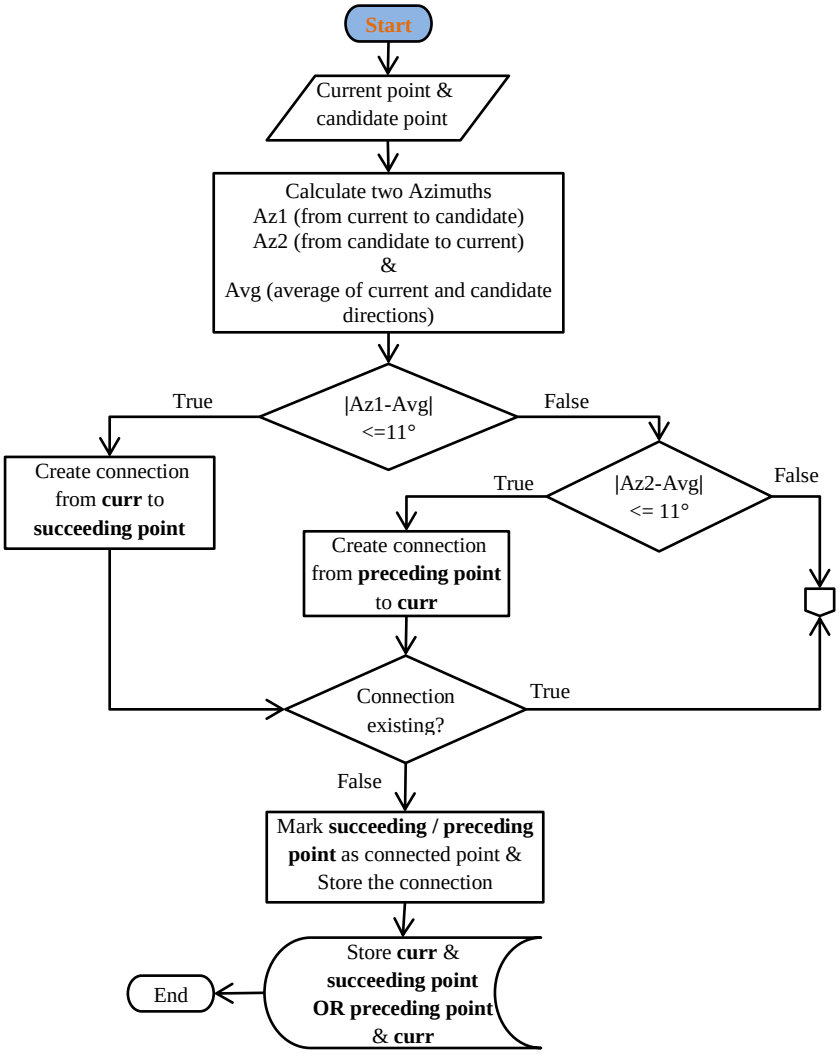


Figure 3.12: Logic flowchart of determining preceding or succeeding point from a single candidate point.

Subroutine 2 is implemented to classify candidate points of a current point based on their precedence relationship in terms of the direction vector between any two distinct points, in case that there are more than one point satisfying aforementioned three conditions in Section 3.4. Eq. 3.4 presents the algorithm of finding one point is ahead of or behind another point with respect to the moving direction and the azimuth. Fig. 3.13

demonstrates the algorithm by using some sample points whose moving directions are within first or third quadrant. For example, $RepPt_{10}$ is selected as the current point and others ($RepPt_{2,\ 4\ 9,\ 12,\ and\ 13}$) are its candidate points in Fig.3.13, because they share four preprocessed GPS trajectories and move in similar directions. Azimuths from $RepPt_{10}$ to each candidate point are calculated using Eq. 3.4 and listed in Table 3.1. The logic flowchart of Subroutine 2 is represented in Fig. 3.14.

Table 3.1: Azimuth from current point to its candidate point

Azimuth (Az)	From	To
1	$RepPt_{10}$	$RepPt_{12}$
2	$RepPt_{10}$	$RepPt_{2}$
3	$RepPt_{10}$	$RepPt_{9}$
4	$RepPt_{10}$	$RepPt_{4}$
5	$RepPt_{10}$	$RepPt_{13}$

In Fig. 3.13, the azimuths (Az_5 and Az_2) are assumed to be about 235° and 85°, respectively; and the moving direction of current point (RepPt10) represented as the red-color arrow is assumed to be 45° in Fig. 3.13 (a) and 225° in Fig. 3.14 (b). With respect to the current point ($RepPt_{10}$), $RepPt_{13}$ is identified as the preceding point and $RepPt_{12}$ is found to be the succeeding point by using Eq. 3.4, in Fig. 3.13 (a). The difference (T) between Az_5 and the direction of the current point is 190°. It is converted to be -170° because of larger than 180°, and is not within the range of (-90, 90). Therefore, $RepPt_{13}$ is classified as the preceding point of $RepPt_{10}$. In the second case as shown in Fig. 3.13 (b), $RepPt_{12}$ is determined as the preceding point of $RepPt_{10}$ based on Eq. 3.4, if Az_1 is assumed to 35°.

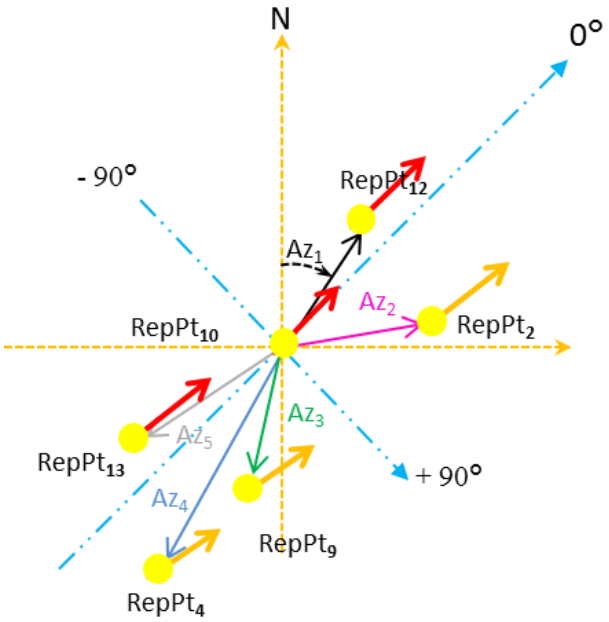
T = Az - direction of current point

If $T > 180^{\circ}$ $T=T-360^{\circ}$ Equation 3.4

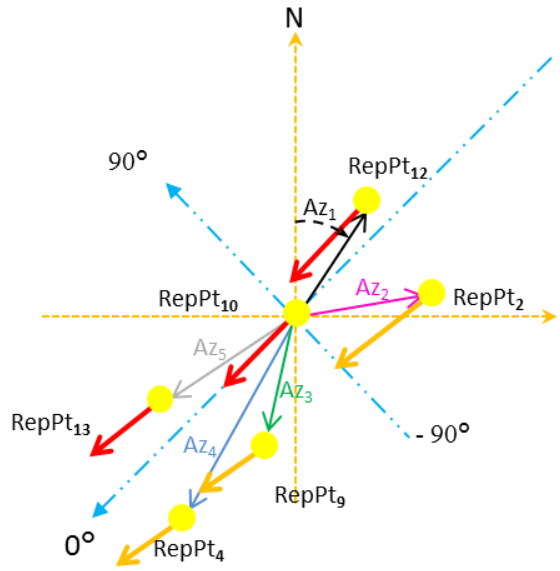
Else $T < -180^{\circ}$ $T=T+360^{\circ}$

If $T > -90^{\circ}$ and $T < 90^{\circ}$ candidate point is ahead of the current point

Else candidate point is behind the current point



(a)



(b)

Figure 3.13: Precedence relationship of candidate points: ahead or behind groups. (a) 1st quadrant and (b) 3rd quadrant.

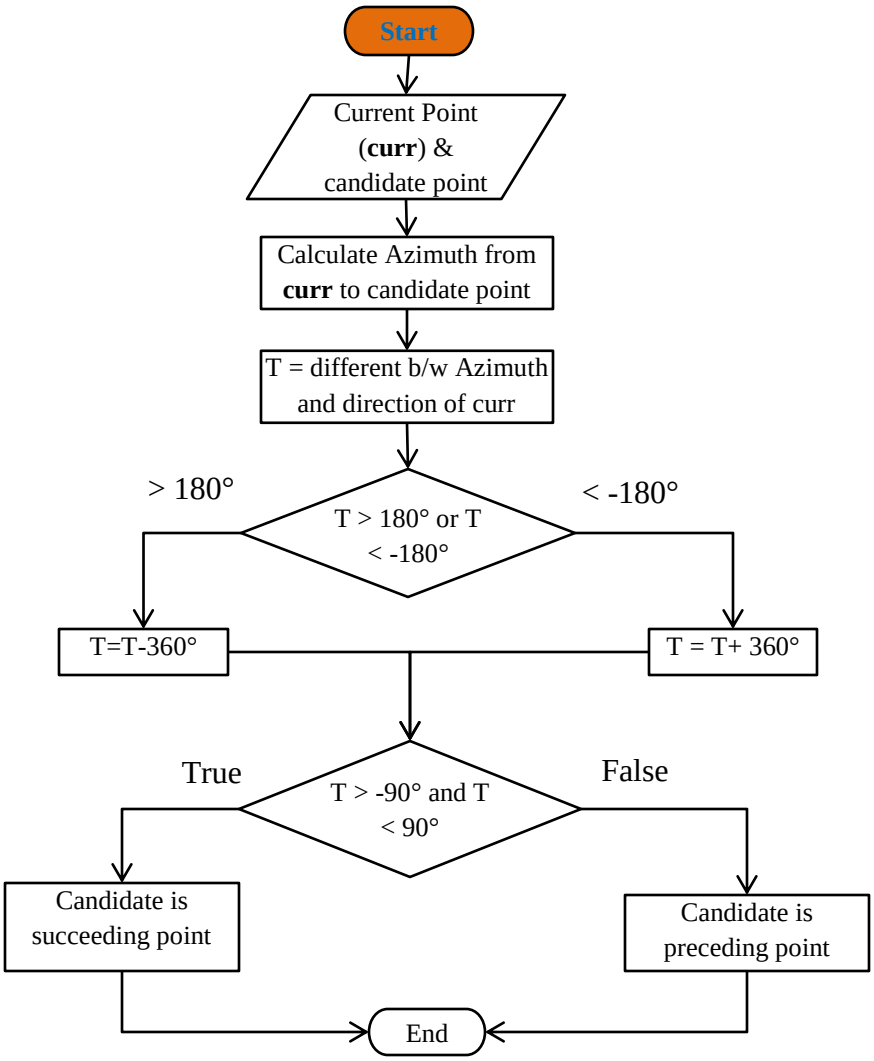
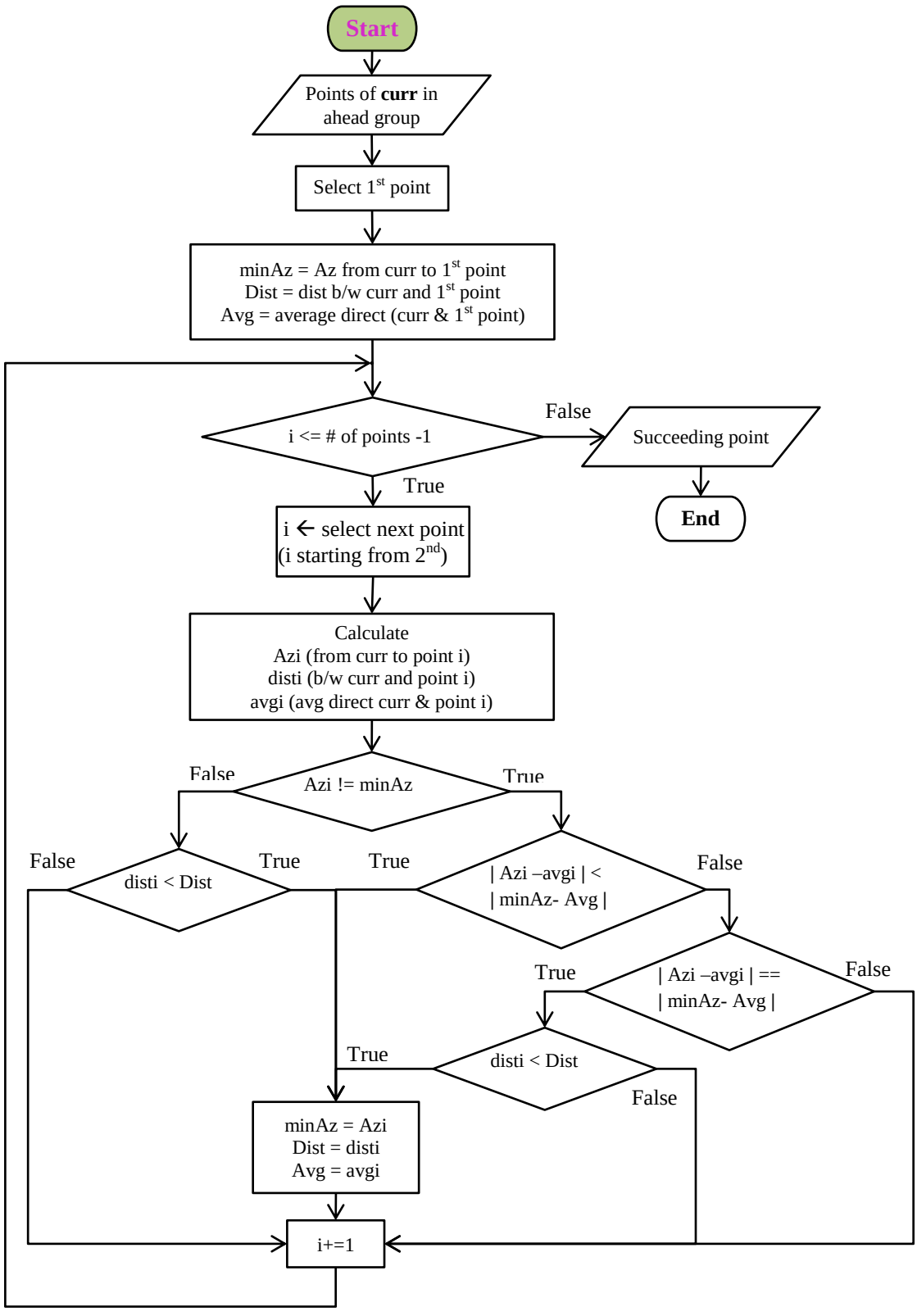


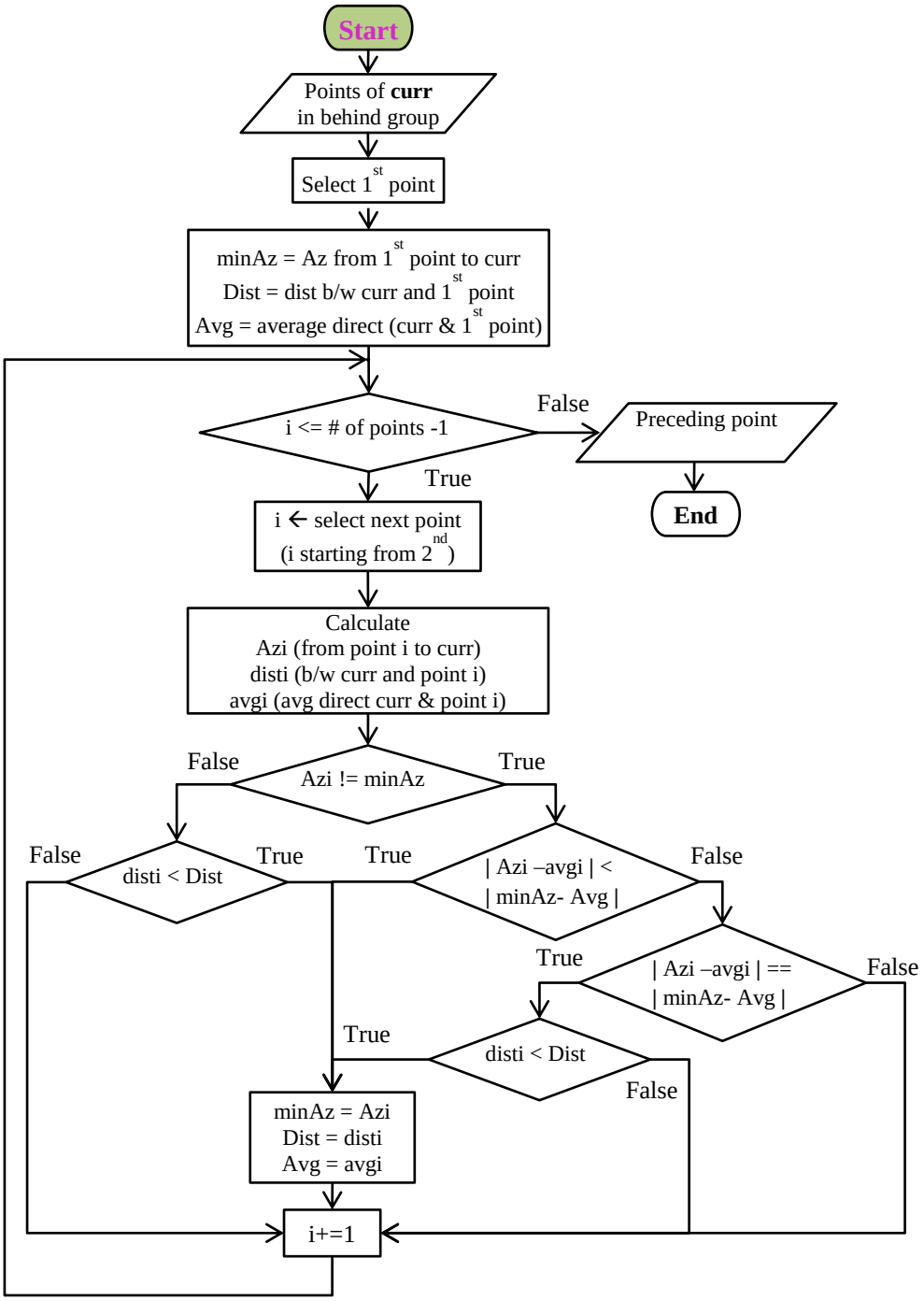
Figure 3.14: Logic flowchart of classifying candidate points of the current point into ahead or behind group

Subroutine 3 is invoked to determine the optimal preceding or succeeding points from candidates classified in ahead or behind group (the output of the Subroutine 2). As shown in Fig. 3.15, subroutine 3A is applied to search the succeeding point while subroutine 3B is required for determining the preceding point. Both of them iterate over the candidate points of any sequence inside a classified ("ahead" or "behind") group until there is one point with minimum difference between the azimuth and the average direction of current and candidate points. The only difference is that the orientation of the

azimuth in Subroutine 3A is from the current point to the point classified in the “head” group, while Subroutine 3B calculates the azimuth from the point inside the “behind” group to the current point. In case there are two points having the same value of azimuth in terms of the current point, the one with shorter distance to the current point is selected as the preceding or succeeding point.



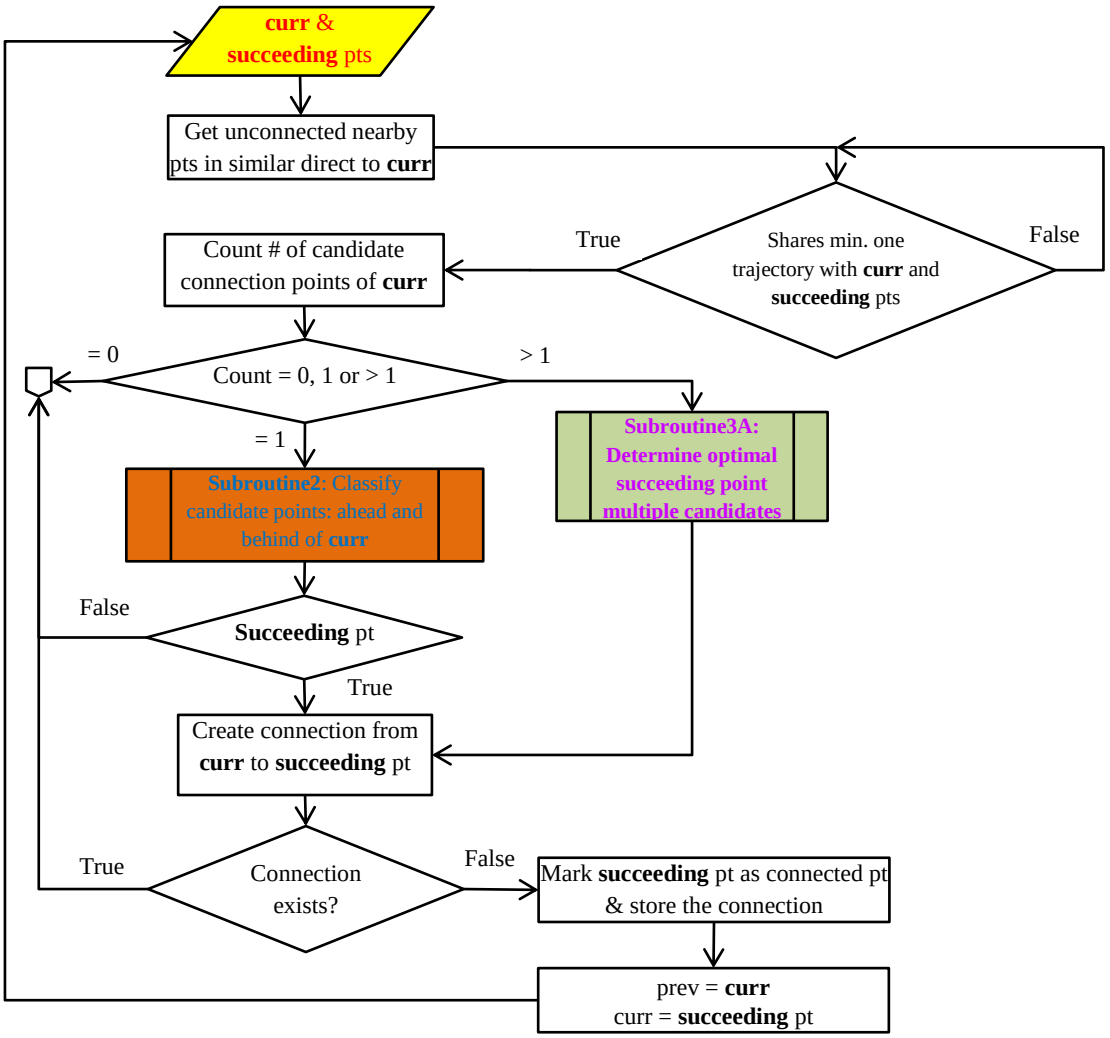
(a) Subroutine 3A



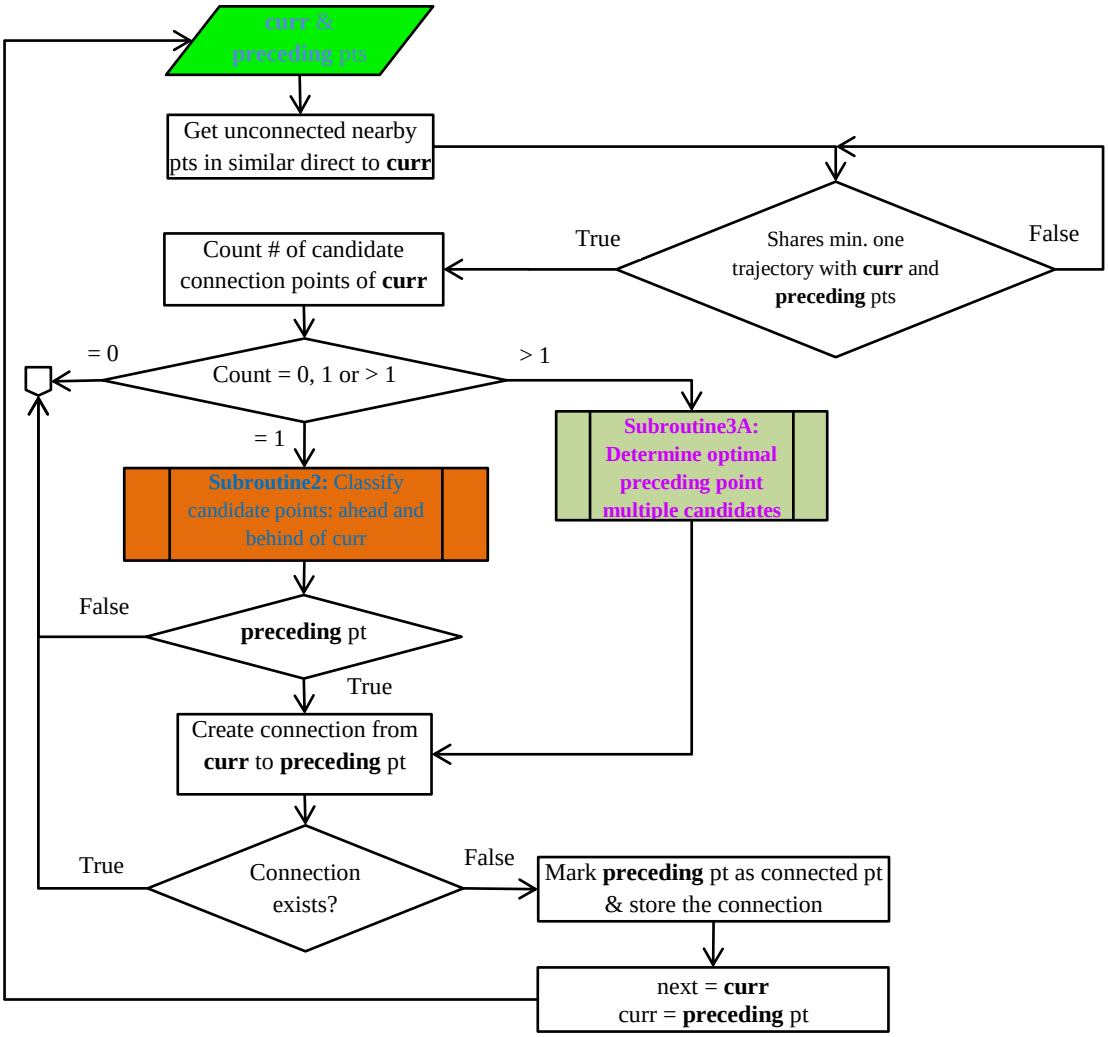
(b) Subroutine 3B

Figure 3.15: Logic flowchart of (a) Subroutine3A determining optimal succeeding point from ahead group (b) Subroutine3B determining optimal preceding point from behind group.

Subroutine 4 and 5 continuously search and connect the optimal preceding and succeeding points, respectively, based on the output from subroutines (1 to 3). In order to obtain optimal connections representing the underlying road network geometry, the second condition is revised to select the candidate point that shares at least one preprocessed GPS trajectory with its two preceding points. Fig. 3.16 represents the logical workflow of detecting sequential succeeding or preceding points from two or three known points obtained from previous subroutines. The performance of searching optimal sequential succeeding and/or preceding points is divided into two cases: bi-directional connection (Case I) and one-way connection (Case II).



Subroutine 4



Subroutine 5

Figure 3.16: Logic flowchart of continuously searching sequential proceeding and/or succeeding points

Case I: Fig. 3.17 illustrates the process of bi-directional search by integrating the Subroutine 4 and 5. Given the current point (Curr) and its preceding and succeeding points (P^0 and S^0); Subroutine 5 takes them as the input to start the backward searching. Fig. 3.17 (b) shows that there are three candidate points (cand1, 2, and3) sharing three preprocessed GPS trajectories (dark-red arrow line) with Curr and P^0. Fig. 3.17(c) represents that the candidate point (cand1) is determined to be the optimal preceding point (P^1) by employing Subroutine 3. The candidate point (cand1) is selected because it

has min difference between the azimuth from Curr to cand1 and average direction of curr and cand1. Subroutine 4 is activated to implement the forward search when the backward search finds no more preceding points, as shown in Fig. 3.17 (d). In case there is only one candidate point sharing preprocessed GPS trajectories with $Curr^{(temp)}$ and P (S^0 and Curr in (a)), Subroutine 2 is invoked to determine whether it is a succeeding or preceding point with respect to Curr. If so, a connection (in dark blue) from S^0 to S^1 is generated.

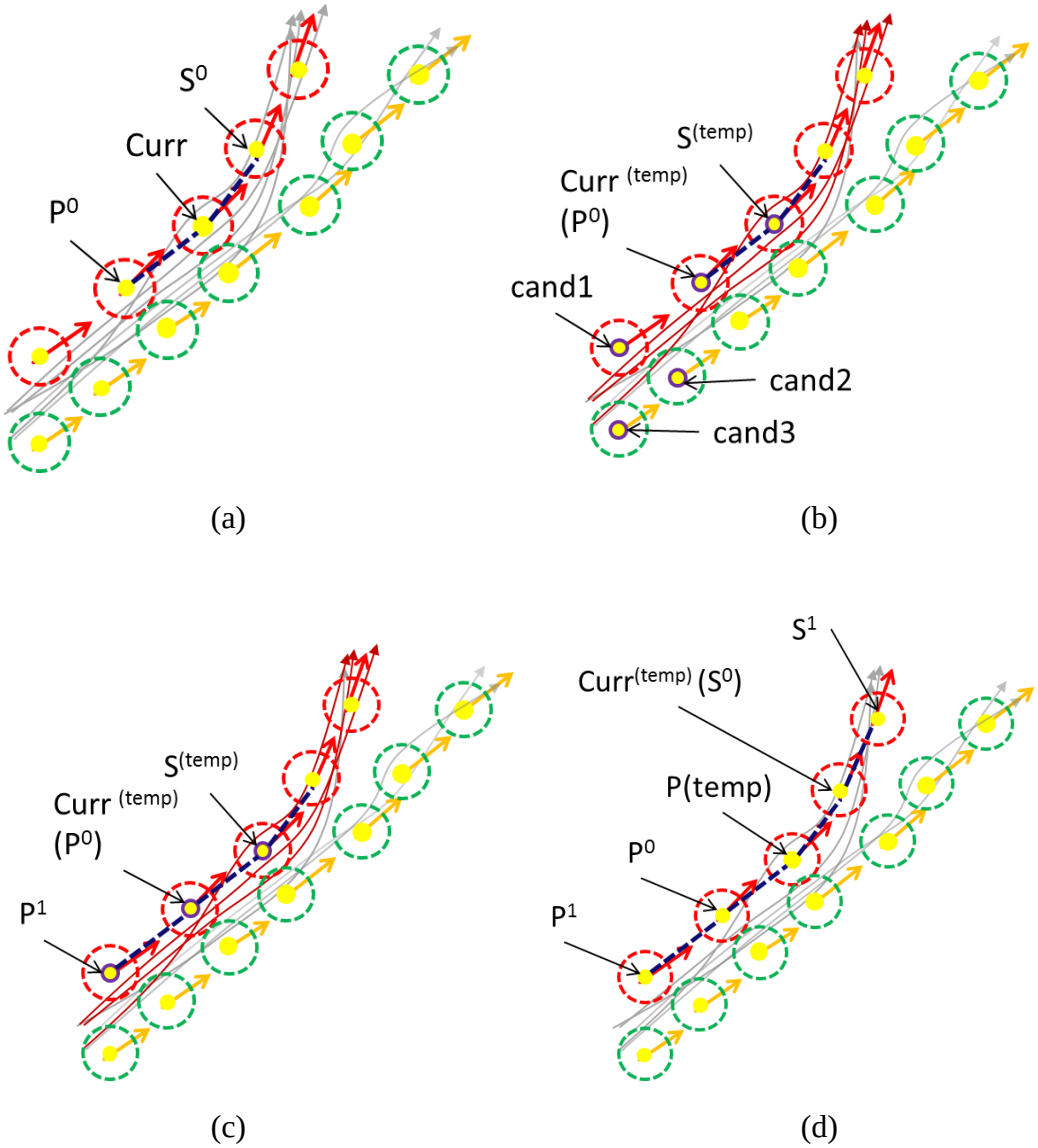


Figure 3.17: Bi-directional connection of preceding and succeeding points

Case II: Fig. 3.18 shows the example of the one-way search by employing the Subroutine 4 when the current point (Curr), an extreme point of the sample area, has no

preceding points along its moving direction. Similarly, the Subroutine 5 is used for backward connecting preceding points from a current point that has no succeeding points. Given a current point (Curr) and its succeeding point (S^0), Subroutine 4 detects four candidate points sharing preprocessed GPS trajectories with them (Fig. 3.18 (b)). Subroutine 3A determines that the candidate point (cand1) is the optimal succeeding point to which a new connection from S^0 can be created, as shown in Fig. 3.18 (c). These two subroutines are repeatedly implemented to generate connections (e.g. S^0 to S^1, S^1 to S^2, S^2 to S^3, and S^3 to S^4) until that there is only one candidate point found (Fig. 3.18 (b-j)). Similar to Case I, Subroutine 2 is invoked again to determine whether the unique candidate point is the succeeding or preceding point with respect to Curr. If so, a connection (in dark blue) from S^4 to S^5 is generated.

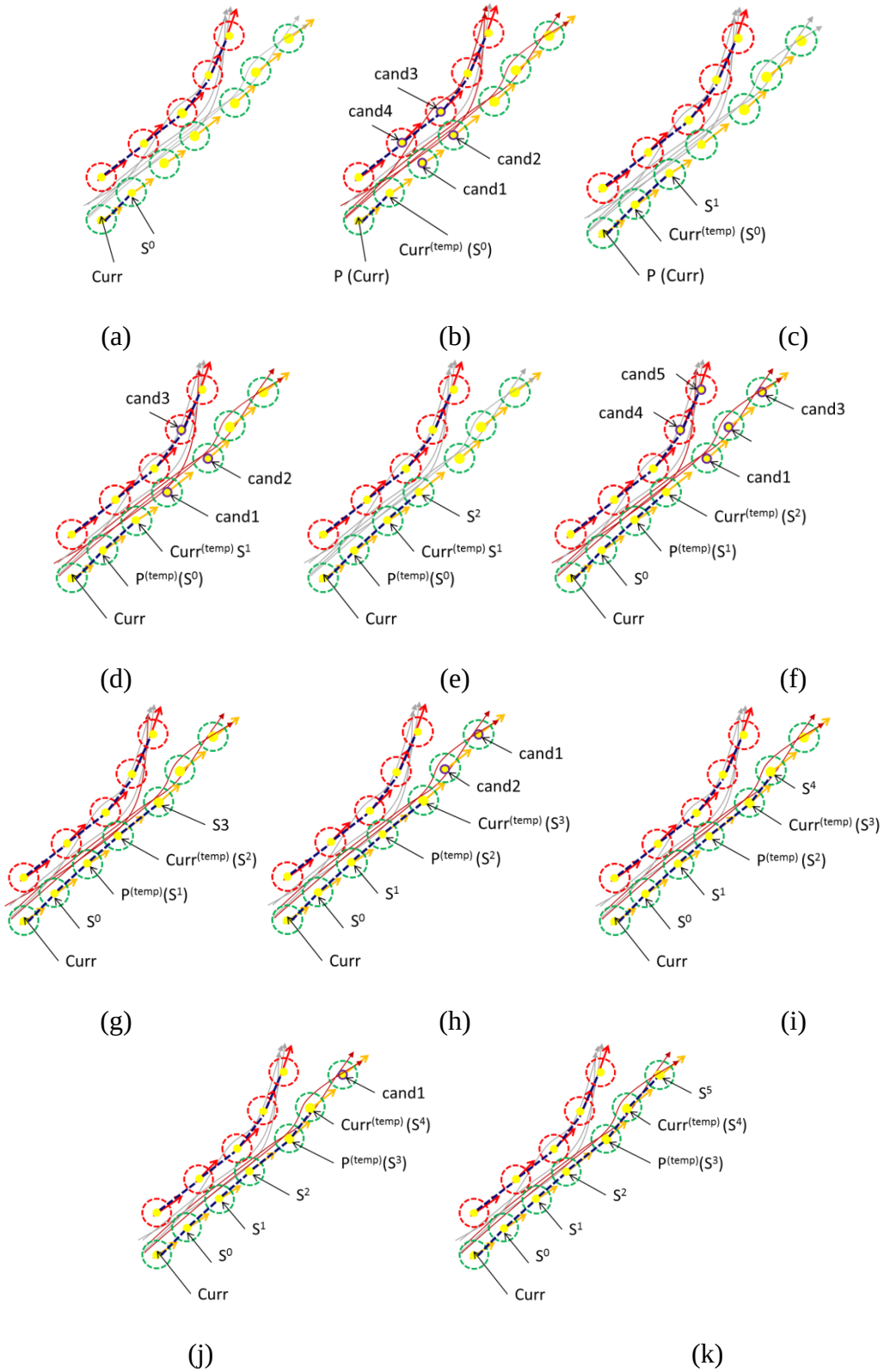


Figure 3.18: One-way connection of succeeding or preceding points

3.5. Polyline Segment Clustering

Section 3.4 reconstructs GPS trajectories by linking representative points based on their topological relationships to preprocessed GPS trajectories and semantic relations. The geometry of road network can be clearly outlined by employing the reformed GPS trajectories, instead of using original GPS trajectories. This section presents a sub-trajectory clustering algorithm that segregates reconstructed GPS trajectories on the same road from others on nearby roads or the same road of opposite direction. The sub-trajectory denotes one polyline segment of the reformed GPS trajectory. One reformed GPS trajectory is composed of a number of sequential polyline segments. The aforementioned polyline-based clustering methods in Section 2.2.2 classify all GPS trajectories on the same road into one cluster. They are sensitive to the parameters of allowable distance amongst polyline segments or assumed road width. For example, the algorithm proposed by Lee et al. (2007) has to be repeatedly tested by using different values of the allowable distance, in order to obtain optimal quality of clustering. The assumed uniform road width utilized by Liu et al. (2012) cannot serve the purpose of constructing the continuous bi-directional centerlines, especially at road junction or splitting.

Unlike similar studies reviewed in Section 2.2.2, the clustering algorithm proposed in this thesis research selects a unique reformed GPS trajectory on each road as the reference; the reference must have the maximum number of polyline segments compared to other trajectories on the same road; and at least one reformed GPS trajectory has the maximum components on each road. Then, polyline segments near each segment of the reference are incrementally grouped into the corresponding cluster. As shown in Fig. 3.19, trace 3 is selected as references because it consists of the six polyline segments (seg. 1 to seg. 6), which is more than others (trace 1, 2, and 4) in the same direction. The allowable distance is determined as the maximum distance between polyline segments of opposite directions that are located at the left-side edges of the two-way road. Within the polyline-segment cluster of seg.3, the distance between polyline segments of trace 5 and trace 4, which move in opposite direction, is larger than that of polyline segments of any two traces in similar direction but is the shortest distance between any two polyline segments in opposite direction. A more formal description is shown in Algorithm 3 in

Appendix A. It consists of two algorithms performing subtasks, including sweep-line algorithm (Subroutine 6 in Appendix A) and recursive polyline searching algorithm (Subroutine 7 in Appendix A).

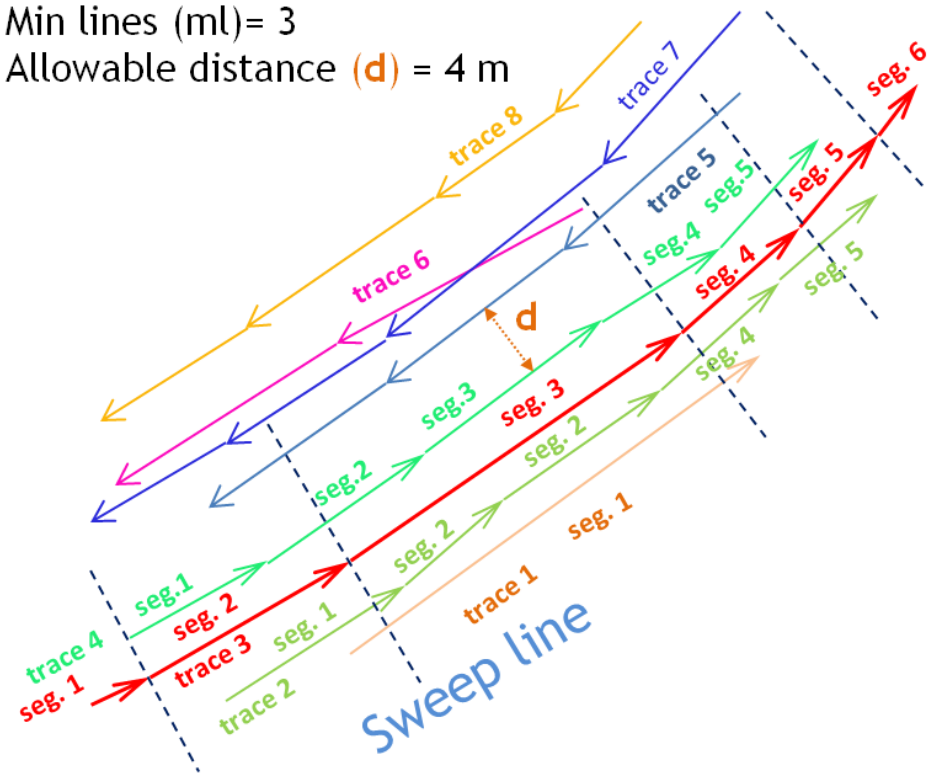


Figure 3.19: Demonstration of polyline clustering algorithm.

The sweep-line algorithm is performed to select portion of nearby polyline segments (divisional polyline segments, Fig. 3.20) within the extent of the reference polyline segment. As shown in Fig. 3.20, two perpendicular lines are placed at starting and ending points of the reference polyline segment. The solid blue dot denotes the intersection points of the perpendicular line and the nearby polyline segments belonging to another reformed GPS trajectory. If the nearby and reference polyline segments are totally overlapping, such as 1st and the Ref Ln Segment in Fig. 3.20, a new divisional polyline segment is generated because the intersection points substitute the original endpoints of the 1st polyline segment. If the entire polyline segment is within the extent of the Ref Ln Segment, such as 4th polyline segment, its endpoints remain unchanged. If

nearby polyline segment, such as 2nd or 3rd polyline segment, and the reference polyline segment are partially overlapping, only one endpoint needs to be replaced by the intersection point. For instance, the ending point of the 2nd polyline segment is substituted by the intersection point while the starting point of the 3rd polyline segment also needs to be replaced. At the end, divisional polyline segments are obtained as shown in Fig. 3.20.

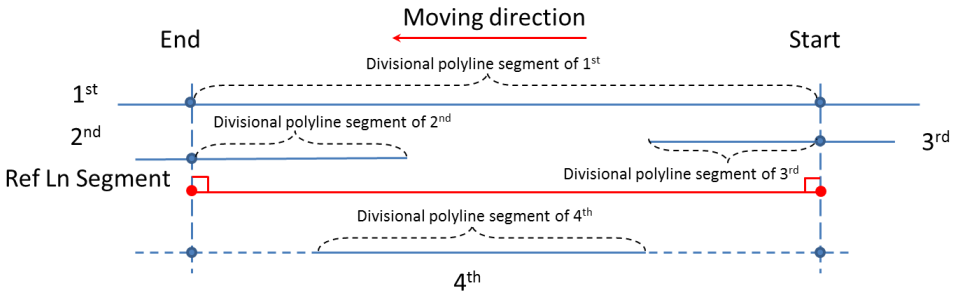


Figure 3.20: Demonstration of intersection points of nearby and reference polyline segments.

The intersection point of any two lines can be found by treating two lines as a system of two linear equations in two variables (x and y). The equations of nearby polyline segment and sweep line perpendicular to the reference polyline segment can be created by using Eq. 3.5 and 3.6. The slope ($m_{\perp}$) and y-intercept ($b_{\perp}$) of the sweep line can be derived based on the moving direction (Azimuth angle, Az) and coordinates of endpoints of the reference line segment. The slope ($m_{nearby\ Ln}$) and y-intercept ($b_{nearby\ Ln}$) of the nearby polyline segment are calculated based on its moving direction and one of endpoints. The coordinates of the intersection point can be obtained by using Eq. 3.7.

$$-m_{\perp}x + y = b_{\perp}$$

Sweep line

$$m_{\perp} = \tan(90^{\circ} - (Az - 90^{\circ}); \qquad \text{Equation 3.5}$$

$$b_{\perp} = y_{Ref\ Ln\ segment} - m_{\perp} \times x_{Ref\ Ln\ segment}$$

$$-m_{nearby\ Ln}\,x + y = b_{nearby\ Ln}$$

Nearby line

$$m_{nearby\ Ln} = \tan(90^{\circ} - Az_{nearby\ Ln}) \qquad \text{Equation 3.6}$$

$$b_{nearby\ Ln} = y_{nearby\ Ln} - m_{nearby\ Ln} \times x_{nearby\ Ln}$$

$$A \cdot X = B \qquad \text{Equation 3.7}$$

$$X = A^{-1} \cdot B$$

$$X = \begin{bmatrix} x \\ y \end{bmatrix}, A = \begin{bmatrix} -m_{\perp} & 1 \\ -m_{nearby\ Ln} & 1 \end{bmatrix}, B = \begin{bmatrix} b_{\perp} \\ b_{nearby\ Ln} \end{bmatrix}$$

The recursive polyline searching algorithm serves the purpose of clustering divisional polyline segment nearby the reference polyline segment according to the allowable distance, threshold value of directional change (11˚), and minimum number of polyline segments inside of a cluster. According to (Lee et al., 2007), the result of polyline clustering is usually affected by the minimum number of polyline segments depending on different data sources or user's preference. The minimum three polyline segments are mandatory for forming a cluster, based on the rule of thumb adopted in the point density-based spatial clustering of applications with noise (DBSCAN).

The reformed GPS trajectories passing through the extent of the reference polyline segment are recorded for later process. For example, Fig. 3.21 show the close-up view of the reference polyline segment seg.3 in Fig. 3.19 and its nearby divisional polyline segments in both directions, which are found by employing the sweep-line algorithm. Starting from the reference seg.3 of trace 3, the algorithm found its closest divisional neighbors (seg.1, 2, 3, and 4 from trace 2; and seg. 2, 3, and 4 from trace 4). There is no more polyline segment on the left side of seg. 3 of trace 3 can be clustered because the distances from polyline segments of trace 5 to those of trace 4 are larger than the allowable distance. On the right side of the reference polyline segment, the divisional seg.1 of trace 1 (in color of dark) is clustered because it has shorter distances to divisional polyline segments of trace 2 than the allowable distance. Meanwhile, the reformed GPS trajectory is recorded if anyone of its polyline segments is divisional into the extent of the reference polyline segment.

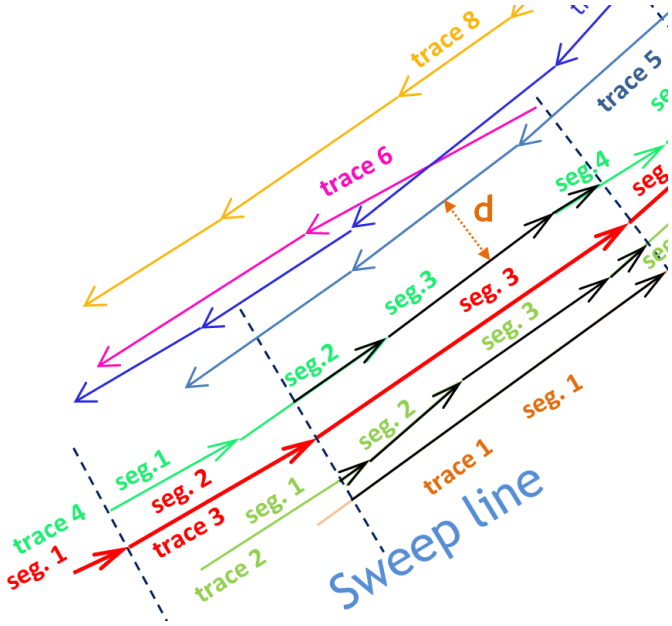


Figure 3.21: The close-up view of reference polyline segment and its nearby divisional polyline segments (dark color).

The distance function employed in the recursive polyline searching algorithm is to calculate the distance between two close polyline segments. The complex distance function consists of perpendicular distance, parallel distance and the angle distance between two polyline segments are already used in the trace clustering algorithm proposed by Lee et al. (2007) and Li et al. (2010). In this thesis research, only the perpendicular distance (shown in Fig. 3.22) is employed because the trajectory reconstruction algorithm converges GPS trajectories on the same road to the middle of the road and the spacing amongst them are less than the lane width.

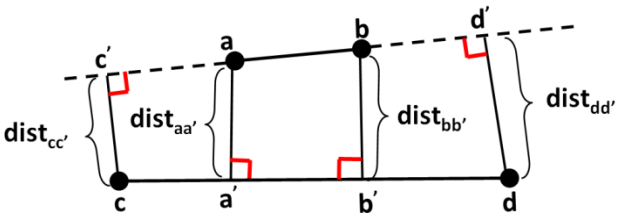


Figure 3.22: Perpendicular distance function for clustering polyline segments.

Given moving direction (Azimuth angle, Az) and coordinates of starting and ending points (c and d) of polyline segment cd, the linear equation of cd based on the

slope and y-intercept of cd can be obtained by using Eq. 3.6. The perpendicular linear equation can be derived according to the fact that the slope of the line perpendicular to cd with slope (m) is -1/m. The y-intercept (b) of the line perpendicular to cd can be found by substituting the coordinates of endpoint (a) into the linear equation in Eq. 3.6. At projected point (a'), these two equations are equal to each other. Therefore, the coordinates of the projected point (a') on the polyline cd can be expressed by coordinates of point (a). The distance $dist_{aa'}$ is the perpendicular distance from endpoint (a) to polyline (cd) that can be solved by using Eq. 3.8. In the recursive polyline searching algorithm, two polyline segments are in the same cluster if perpendicular distances between them are within the allowable distance. As shown in Figure 3.22, polyline segments ab and cd are grouped together if their perpendicular distances ($dist_{aa'}$, $dist_{bb'}$, $dist_{cc'}$, and $dist_{dd'}$) are not larger than the allowable distance. Checking all perpendicular distances between any two close polyline segments make sure that at road split polyline segments on different road are not grouped into one cluster.

Perpendicular distance

$$dist = \frac{|y_p - mx_p - b|}{\sqrt{m^2 + 1}} \qquad \text{Equation 3.8}$$

where, m is the slope of the polyline segment on which point is projected; b is y-intercept of the polyline segment; (x_p, y_p) is the coordinates of the endpoint

3.6. Road Centerline Extraction

Section 3.5 generates the clustered polyline segments at each road segment. This section is to extract the road centerline segment by merging all divisional polyline segments belonging to the cluster. As discussed in Section 2.2.2, the method proposed by (Tavares and Padilha, 1995) is more appropriate for the complex distribution of polyline segments in a cluster, such as partial overlapping, full overlapping, and none overlapping segments in similar directions. On the other hand, the resulting polyline segment is closer to its true location in the cluster because the centroid of all participated polyline segments in the cluster is calculated by taking their lengths as weights. Therefore, the polyline merging algorithm is adopted from the edge line merging used in the area of computer vision (Tavares and Padilha, 1995) with two modifications.

Only two polyline segment distributions, partial overlapping and full overlapping, are considered because sweep lines at starting and ending points of the reference polyline segment divide nearby polyline into two or three segments. The divisional polyline segments are grouped into a cluster by segment. As shown in Fig. 3.23, polyline segments (seg. 3 and 4 of trace 2, seg.3 and 4 of trace 1) are partial overlapping with the reference polyline segment (seg.4 of Ref. Trace); and the polyline segment (seg.4 of trace 5) is full overlapping.

The divisional polyline is involved if its reformed GPS trajectory has similar moving behavior with the reference trajectory. Fig. 3.24 illustrates the approach of selecting the optimal divisional polyline segments from each cluster. At reference polyline segment seg.4, only two reformed GPS trajectories (trace1 and 2) passing through extents of its predecessor (seg.3) and successor (seg.5) simultaneously. Therefore, only divisional polyline segments in color of dark (seg.3 and 4 of trace2, seg.3 and 4 of trace1) and the reference polyline segment are used to calculate centerline segment. In another case, the polyline segment (seg.4 of trace 6) is partial overlapping with the reference polyline segment (seg.4 of Ref. Trace). However, it is not involved in the extraction of road centerline segment regarding the seg.4 of the reference trace (Ref. Trace in Fig. 3.23) because the trace 6 does not have common movement with the reference trajectory. The trace 6 only passes through extents of seg.4 and its predecessor (seg.3). Thus, the divisional polyline segment of seg.4 in trace 6 (dark color) is not used to extract road centerline segment with respect to the reference polyline segment (seg4. of Ref. Trace). But, it will be used to update the position and direction of extracted road centerline segment in later process if it is within the estimated road width. The road width can be estimated by generating a polygon around the extracted road centerline based on its furthest vertexes at four corners.

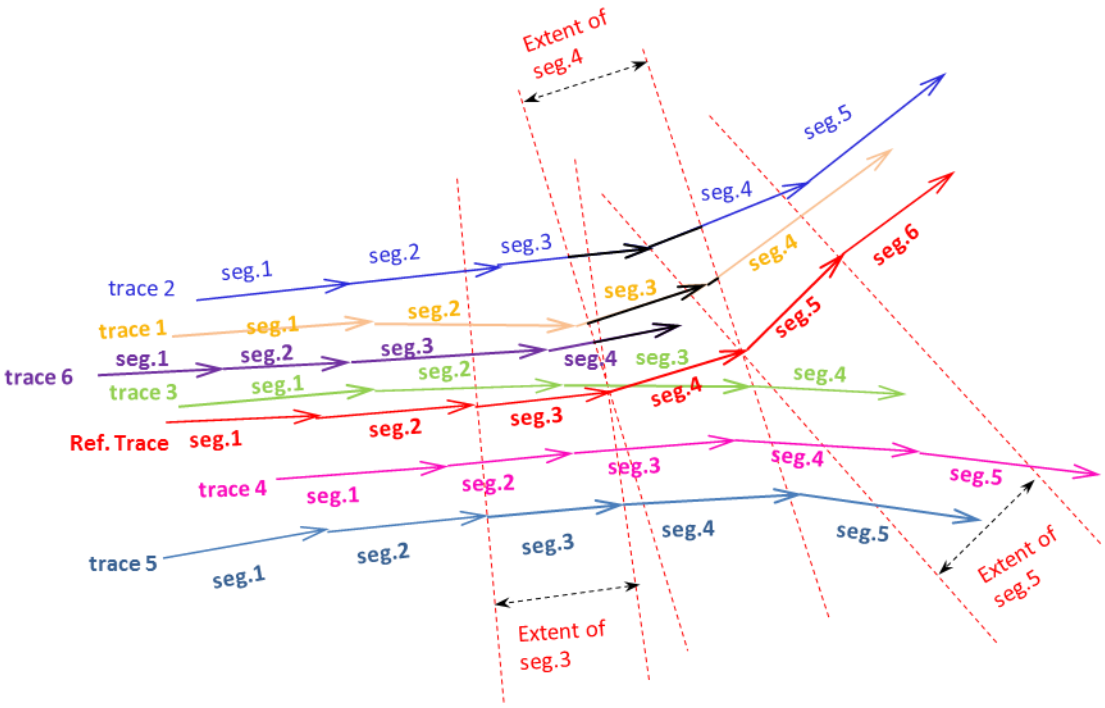


Figure 3.23: Illustration of divisional polylines involved in the polyline merging algorithm.

The road centerline at each cluster is extracted by merging all divisional polyline segments into one representative polyline segment. The algorithm proposed in this section runs in three steps (A more formal description of this algorithm is shown in Algorithm 4 in Appendix A.):

1) Road centerline segment extraction;
2) Road width estimation; and
3) Road centerline segment updating.

Road centerline segment extraction: The direction and coordinates of the road centerline can be derived by merging all divisional polyline segments in the cluster into one polyline. Fig. 3.24 illustrates the polyline merging algorithm adapted from the edge line merging used in the area of computer vision. Given a polyline cluster with n divisional polyline segments; $C_{polyln} = \{polyln_1, polyln_2, \ldots, polyln_n\}$ where each of them contains moving direction (Az) and coordinates of starting and ending points intersected with the sweeping line; $polyln_i = \{x_{start}, y_{start}, x_{end}, y_{end}, direct, length\}$ where i <= n.

Taking the length of the polyline segment as the weight, coordinates of the centroid of the polyline cluster and the overall moving direction of the merged polyline are calculated by using Eq. 3.9 and Eq. 3.10, respectively.

$$Xo' = \frac{\sum_{i=1}^{n} li\ (ix_{start} + ix_{end})}{2\sum_{i=1}^{n} li} \qquad \text{Equation 3.9}$$

$$Yo' = \frac{\sum_{i=1}^{n} li\ (iy_{start} + iy_{end})}{2\sum_{i=1}^{n} li}$$

where, l = length of the polyline; ix & iy = coordinates of start and end points of the divisional polyline segment i.

$$\theta r = \frac{\sum_{i=1}^{n} li \cdot Azi}{\sum_{i=1}^{n} li} \qquad \text{Equation 3.10}$$

$$\theta = -(\theta r - 90°)$$

where, θr = moving direction of the merged; θ = rotation angle from original coordinate system XY to new coordinate system X′Y′; Azi = direction of divisional polyline segment i that is measure as the azimuth angle.

The new coordinate system X′Y′ with x-axis parallel to the overall moving direction and origin at the centroid O' can be defined based on the rotation relative to the XY-coordinate system. Along with the overall moving direction, the starting point of the merged point always has minimum value of x while the ending point has the maximum one. In order to find the minimum and maximum values of x coordinate, all endpoints of divisional polyline segments are projected to the x-axis of the X′Y′ frame by applying Eq. 3.11. The coordinates of starting and ending points in original XY-coordinate system can be determined based on the coordinate transformation in Eq. 3.12.

$$x' = (x - Xo')\cos\theta + (y - Yo')\sin\theta \qquad \text{Equation 3.11}$$

$$y' = 0$$

$$\begin{aligned} &\text{x\_start} = min\_x'\cos\theta - min\_y'\sin\theta + Xo'; \\ &\text{y\_start} = \min\_x'\sin\theta + \min\_y'\cos\theta + Yo'; \\ &\text{x\_end} = max\_x'\cos\theta - \max\_y'\sin\theta + Xo'; \\ &\text{y\_end} = \max\_x'\sin\theta + max\_y'\cos\theta + \text{Yo'}; \end{aligned} \qquad \text{Equation 3.12}$$

As shown in Fig. 3.24, x-coordinates of endpoints (a, b, c, d, e, f, m, and n) can be projected to the x-axis of X′Y′ frame as projected points (a′, b′, c′, d′, e′, f′, m′, and n′). The starting and ending points of the merged polyline, passing through the centroid O' and on the overall moving direction, are defined by two projected points (a′ and b′) that are farther apart on the X′-axis.

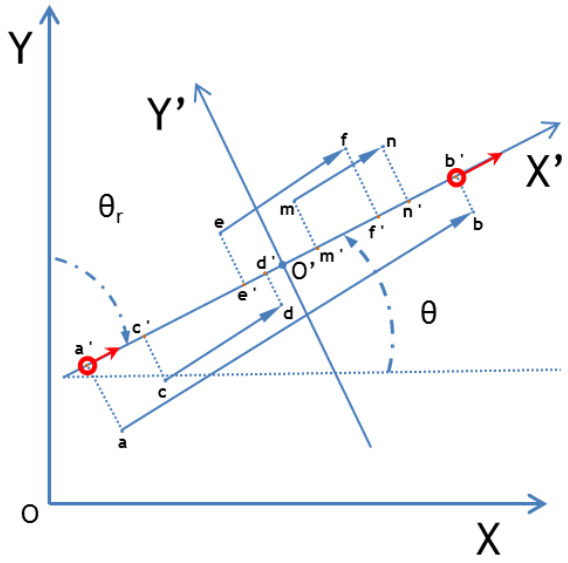


Figure 3.24: Extraction of road centerline segment at each cluster.

Road width estimation: Instead of using assumed constant road width as discussed in Section 2.2.2, the road width at every extracted road centerline segment can be estimated by outlining a polygon based on four vertexes, instead of using the assumed uniformed road width. The vertex is defined as the furthest endpoint on each side of the starting or ending point of the extracted road centerline segment. Fig. 3.25 illustrates the off-line point P (x0, y0) and the extracted road centerline segment with starting point A (x1, y1) and ending point B (x2, y2). In order to determine the side of the oriented line AB at which the point P is located, the definition of the cross product is adapted by using Eq. 3.13 relative to the overall moving direction based on the right-hand rule.

$$a = (x2 - x1, y2 - y1)$$
$$b = (x0 - x1, y0 - y1)$$
$$z = a \times b = \|a\|\|b\| \sin\theta$$
$$z = a \times b = (x2 - x1)(y0 - y1) - (y2 - y1)(x0 - x1) \quad \text{Equation 3.13}$$

If z > 0; P is on left-side of AB

If z < 0; P is on right-side of AB

where, a is the vector from A to B, b is vector from A to P, z is the vector perpendicular to a and b, θ is the angle between vectors a and b within the range from zero to 180°.

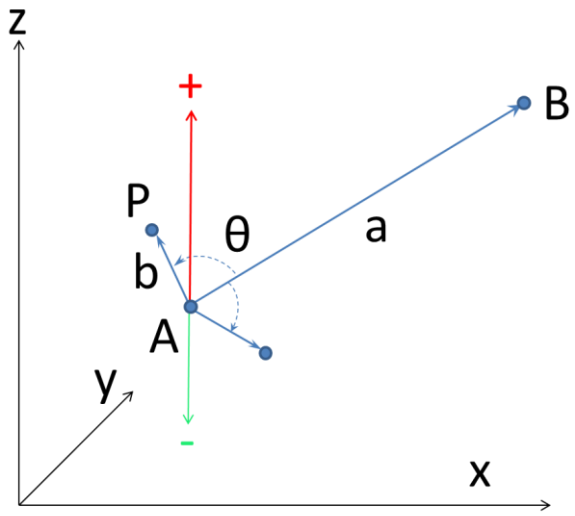


Figure 3.25: Determining left or right-side point of the oriented line segment.

Road centerline segment updating: To perform feature-based topological analysis to identify some unclassified divisional polyline segments within the polygon. If so, updating the coordinates and direction of extracted road centerline segment. Recall the second modification early introduced in this section, the divisional polyline is involved in the polyline merging algorithm if its reformed GPS trajectory has similar moving behavior with the reference trajectory. It results in that the unclassified divisional polyline segment is generated. For instance, in Fig. 3.23, the polyline segment (seg.4 of trace 6) is partial overlapping with the reference polyline segment (seg.4 of Ref. Trace). However, it is not involved in the extraction of road centerline segment regarding the seg.4 of Ref. Trace because the trace 6 does not have common movement with the

reference trajectory. The plumb-line algorithm [21] is utilized to determine if both intersection points of the divisional polyline segment are inside of the polygon. In the plumb-line algorithm, if the vertical line dropped from the point intersects an odd number of sides of the polygon, the point is inside the polygon. Otherwise, the point is not in the polygon. Eqs 3.9 – 3.12 are used again to update the location and direction of the extracted road centerline segment if there is any new divisional polyline segment with both endpoints inside of the polygon.

3.7. Topological Connectivity at Road Intersections

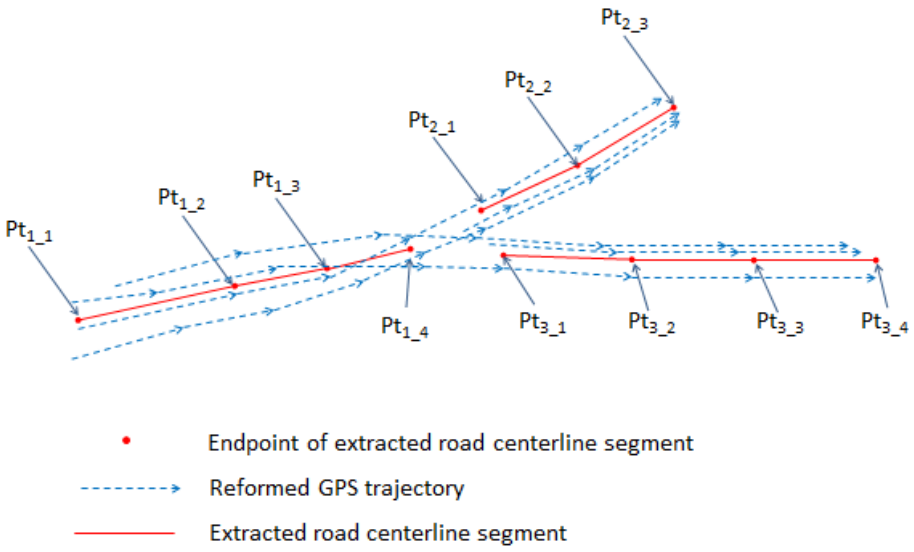
Section 3.6 extracts the road centerline from clustered divisional polylines by segment. On each road, there is only one road centerline that is composed of a number of sequential centerline segments. As introduced in Section 3.5, the reformed GPS trajectory with the maximum number of components on every road is considered to be reference line based upon which the polyline segment clustering algorithm is applied to classify similar trajectories. Extracted road centerlines are disconnected where road is splitting or merging, because relative small change in moving directions of reformed GPS trajectories or sparse distribution of original GPS trajectories. Reformed GPS trajectories have moving directions in common when the road starts splitting apart or merging together. The sparse spatial distribution of vehicle positioning points causes that the number of reformed GPS trajectories is less than the parameter value of polyline clustering algorithm in Section 3.5. Fig.3.26 illustrates two significant scenarios regarding the connectivity of the extracted road centerlines, Y-split and intersection. Every extracted road centerline is composed of a sequential set of endpoints; $CL_i = \{Pt_{i\_1}, Pt_{i\_2}, \ldots, Pt_{i\_j}\}$, where Pts are stored with respect to the direction of CL.

Section 3.7 addresses the topological connectivity of extracted road centerlines in order to create an integrated road network represented by nodes and edges. The intersection of two or more extracted road centerlines are represented as nodes; and connections among neighbor nodes are represented by edges. The identification of nodes is done through spatial query that is implemented at endpoints (starting and ending points)

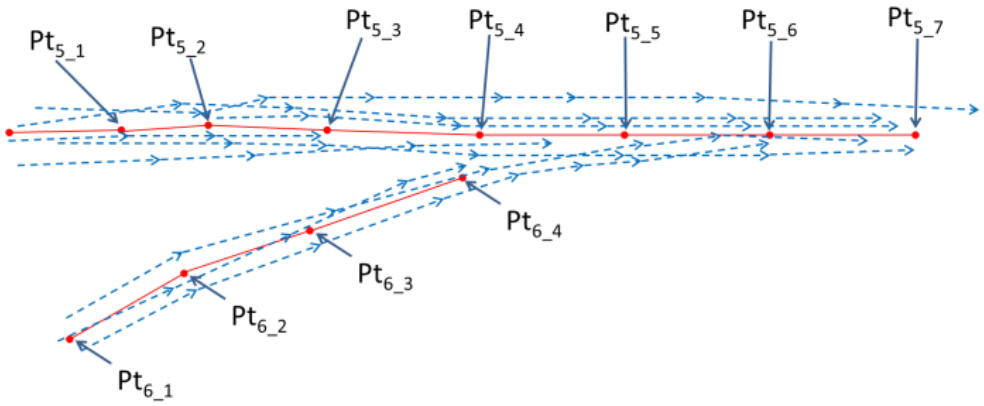
[21] Yeung, A. K., & Lo, C. (2002). *Concepts and techniques of geographic information systems* Prentice Hall.

of the extracted road centerline, respectively. A more formal description of this algorithm is shown in Algorithm 5 in Appendix A. The connectivity between extracted road centerlines are derived based on the semantic road-knowledge based rules of their endpoints and the topological relationship among endpoints of extracted road centerlines and the underlying reformed GPS trajectories.

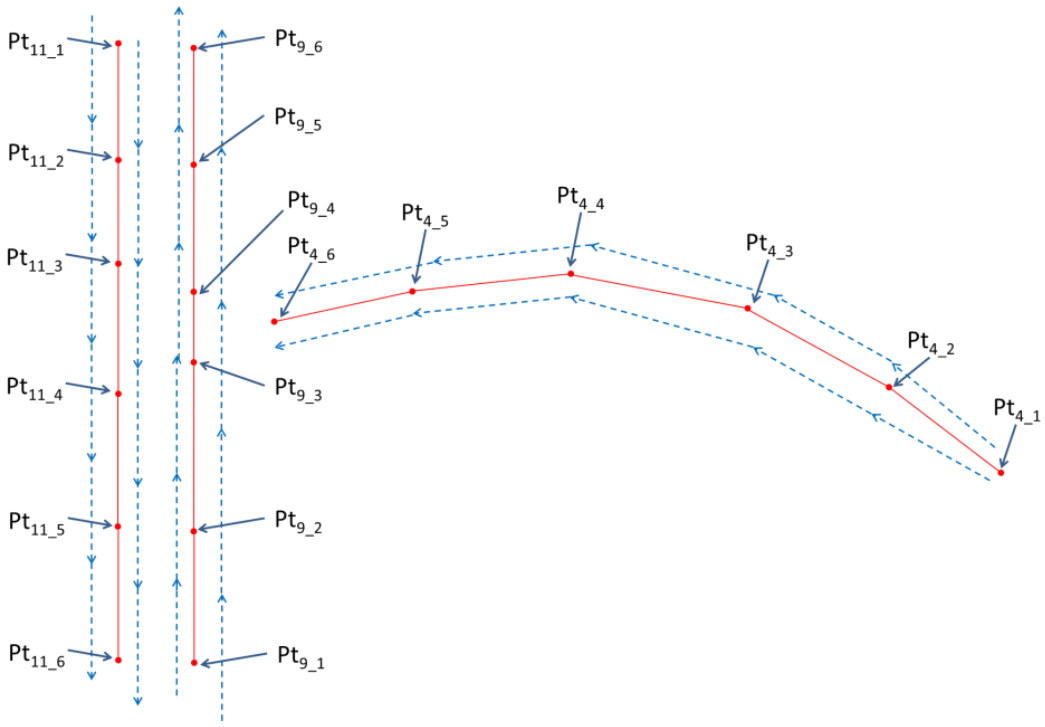
1) The ending point of one extracted road centerline may have one or two forward links to starting points of others;
2) The starting point of one extracted road centerline may have one or two backward links to ending points of others;
3) The ending point of the extracted road centerline of the highway exit ramp has two forward links intersecting with other extracted road centerlines; and
4) The connection is formed if and only if the common underlying reformed GPS trajectories passing through extents of both extracted road centerlines.



(a)



(b)



(c)

Figure 3.26: Illustrations of connectivity of extracted road centerlines.

The geometric relationship amongst endpoints of extracted road centerlines is a prerequisite for linking them together to be an integrated road network. Fig. 3.26(a) illustrates the disconnected road centerlines at road Y-split section. Eq. 3.13 is applied repeatedly at starting and ending points (Pt_{1_4}, Pt_{2_1}, and Pt_{3_1}) to assist in determining the optimal node at where these three extracted road centerlines are intersected. For example,

nearby starting points (Pt_{2_1}, Pt_{2_2}, Pt_{3_1}, and Pt_{3_2}) are within 150-meter range of the ending point (Pt_{1_4}) are classified into two groups, left or right-side of the direction of extracted road centerline CL_1. The threshold value of 150 m is selected in order to compensate the unexpected discontinuity caused by 100-meter distance threshold adapted in preprocessing of original GPS trajectories. In each classified starting-point group, the optimal starting point is derived if the difference between its azimuth angle (from ending point, Pt_{1_4}, to starting point, Pt_{2_1} or Pt_{2_2}) and the direction of ending point (Pt_{1_4}) is the smallest. At each optimal starting point, a reverse search is implemented by using Eq. 3.13 to determine the optimal ending point on CL_1. In case that there are two optimal ending points found by the reverse search, they are recorded as intersection-nodes if their space on CL_1 is over 50 m (see experimental threshold in Section 5.1). Otherwise, the one closer to both optimal starting points is recorded as the node if the distance is within 50 m. Fig. 3.27(b) shows the disconnection extracted road centerlines at road merging section. In this case, Eq. 3.13 is utilized at the ending point (Pt_{6_4}) to search the intersecting node with CL_5. The intersecting node must be one of endpoints on CL_5 which has minimum difference between its azimuth angle (from Pt_{6_4} to itself) and the direction of CL_6.

The most challenge is the connectivity of extracted road centerlines representing the turning at the road intersection. As shown in Fig. 3.26(c), the ending point (Pt_{4_6}) is supposed to be linked to one right-side node on CL_9 and one left-side node on CL_{11}. The intersection node detected could not represent the actual turning information at this road intersection if only Eq. 3.13 is applied. For instance, Pt_{9_3} and Pt_{9_4} could be linked from Pt_{4_6} because they all have minimum change in azimuths comparing to others at each side. Therefore, Eq. 3.13 combined with Eq. 3.14 are used to guarantee that only endpoints of CL9 on right-side of CL_4 and those of CL_{11} on left side of CL_4 are taken into account.

Given last two endpoints of one extracted road centerline (P_1 and P_2, where P_1 is the endpoint P_{4_5} and P_2 is the ending point Pt_{4_6}) and two endpoints (Pt_{9_4} and Pt_{11_3}), Pt_{9_4} and Pt_{11_3} are classified into the right-side group by Eq. 3.13 but on opposite directions of the road. Let Pt_3 denoting either one of them, the vector starting at Pt_3 can be calculated by assuming one close temporary point (Pt_4) on the direction of Pt_3 in Eq. 3.14. After that, Eq. 3.13 takes vectors $\overrightarrow{P_1P_2}$ and $\overrightarrow{P_3P_4}$ to further distinguishes endpoints on the same side but not on the same road. For instance, Pt_{9_4} is on CL_9 and Pt_{11_3} is on CL_{11}.

Finally, only the endpoint on each extracted road centerline and with minimum distance to Pt_{4_6} is deemed to be the intersection node which can be linked from the ending point (Pt_{4_6}).

$P_1 = (P_1x, P_1y)$; $P_2 = (P_2x, P_2y)$; $P_3 = (P_3x, P_3y)$; $P_4 = (P_4x, P_4y)$

If P_3.direct IN $[0°, 90°)$
$P_4x = P_3x + 5 \times \sin(P_3x.\text{direct})$
$P_4y = P_3y + 5 \times \cos(P_3x.\text{direct})$

Else If P_3.direct IN $[90°, 180°)$
$P_4x = P_3x + 5 \times \sin(180° - P_3x.\text{direct})$
$P_4y = P_3y - 5 \times \cos(180° - P_3x.\text{direct})$

Equation 3.14

Else If P_3.direct IN $[180°, 270°)$
$P_4x = P_3x - 5 \times \sin(P_3x.\text{direct} - 180°)$
$P_4y = P_3y - 5 \times \cos(P_3x.\text{direct} - 180°)$

Else If P_3.direct IN $[270°, 360°)$
$P_4x = P_3x - 5 \times \sin(P_3x.\text{direct} - 270°)$
$P_4y = P_3y + 5 \times \cos(P_3x.\text{direct} - 270°)$

where, the constant value is an assumed distance starting from P_3.

Chapter 4. Experimental Data and Study Area

4.1. GPS Data Collection

The real-world GPS dataset was collected by the smartphone application named Traffic Alert developed by the GreenOwls Mobile Solutions Inc. After initial launch of Traffic Alert, over 4,000 users have registered and downloaded the applications for streaming the GPS data to the central database. The experimental data, provided by the GreenOwls Mobile Solution Inc., spans from January 1st to June 15th in 2011 and covers the southern Ontario. Fig. 4.1 shows the spatial distribution of 30,906,455 GPS positioning points of smartphone users (Blackberry, iPhone and Android-based smartphones), who traveled around the south region of the Ontario province, including points related to driving, idling, walking, and biking. Each GPS trajectory is composed of a sequence of time-stamped positioning points which have geographic coordinates (latitude and longitude), direction (azimuth), timestamp, accuracy, speed, and the system-generated identification (SID). The GPS points related to the movement of smartphone users were collected with sampling rate of one second. Most smartphone users never turned off the location access permission to stop the GreenOwls's Traffic Alert application. Consequently, point clouds were generated at some particular locations, such as building (home, office, and shopping mall), parking lots, sidewalks, driveways, traffic lights, and traffic congestions. The SID is randomly generated and automatically assigned to the user at the initialization of the Traffic Alert application. The SID does not change with the update of the operating system of the smart phone. Other attributes of the positioning point, such as accuracy, direction, and speed, are GPS calculations that are uploaded to the central server in real time.

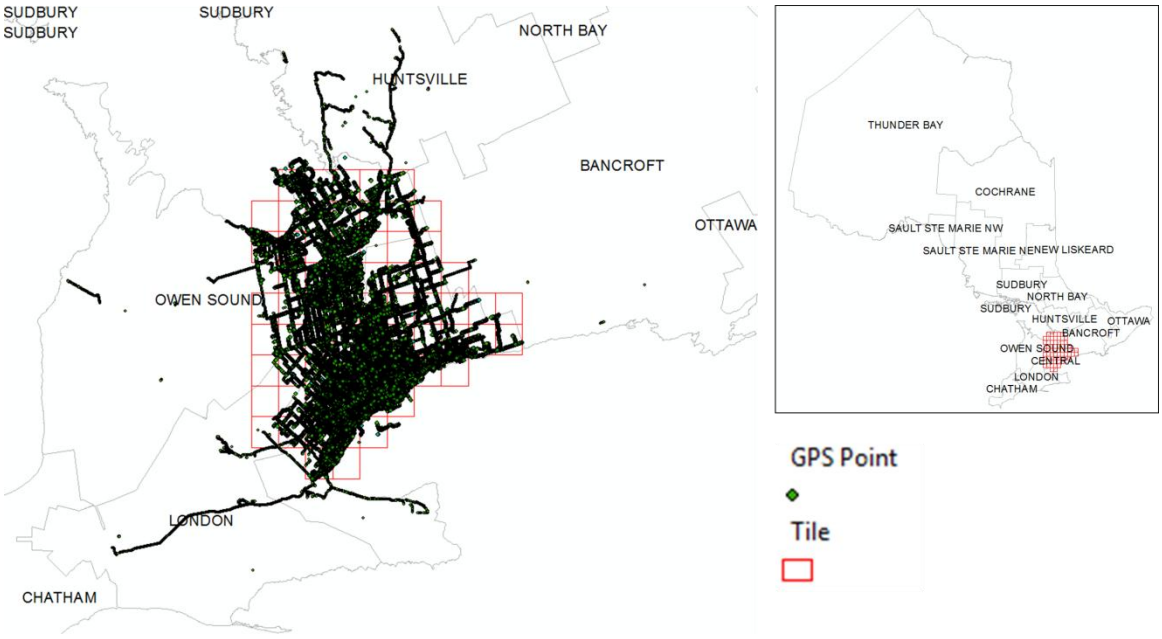


Figure 4.1: Spatial distribution of collected GPS data.

4.2. Case Study Area

The original GPS data retrieved from the central database were stored in comma separated values (.csv) files monthly. Since Environmental Systems Research Institute (ESRI) shapefile is unable to handle such huge amount monthly GPS data because all component files of a shapefile (.shp, .dbf, and .shx) are limited to 2 GB each, the experimental GPS data were split into 67 equal-area tiles covering the majority regions of the southern Ontario as shown in Fig. 4.1. Monthly GPS data in each tile were merged into one feature class in the ArcGIS file geodatabase due to the unlimited file geodatabase size. In the file geodatabase, each feature class can store 4,294,967,295 records. The 11th tile was selected as the study area, because it has the maximum number of GPS points comparing to other tiles. The 11th tile the region in the Great Toronto Area with a geographical range of 79° 34’ 0.89’’ W, 43° 35’ 39.17’’ N, 79° 22’ 44.42’’ W, and 43° 44’ 38.15’’ N. Fig. 4.2 shows that the study area covering the southeastern region of Peel and the west of Toronto as highlighted in the blue rectangular area. The GPS dataset, inside of the tile 11, is composed of 2,026,251 GPS points. The detailed analysis of GPS data is available in Section 5.1.

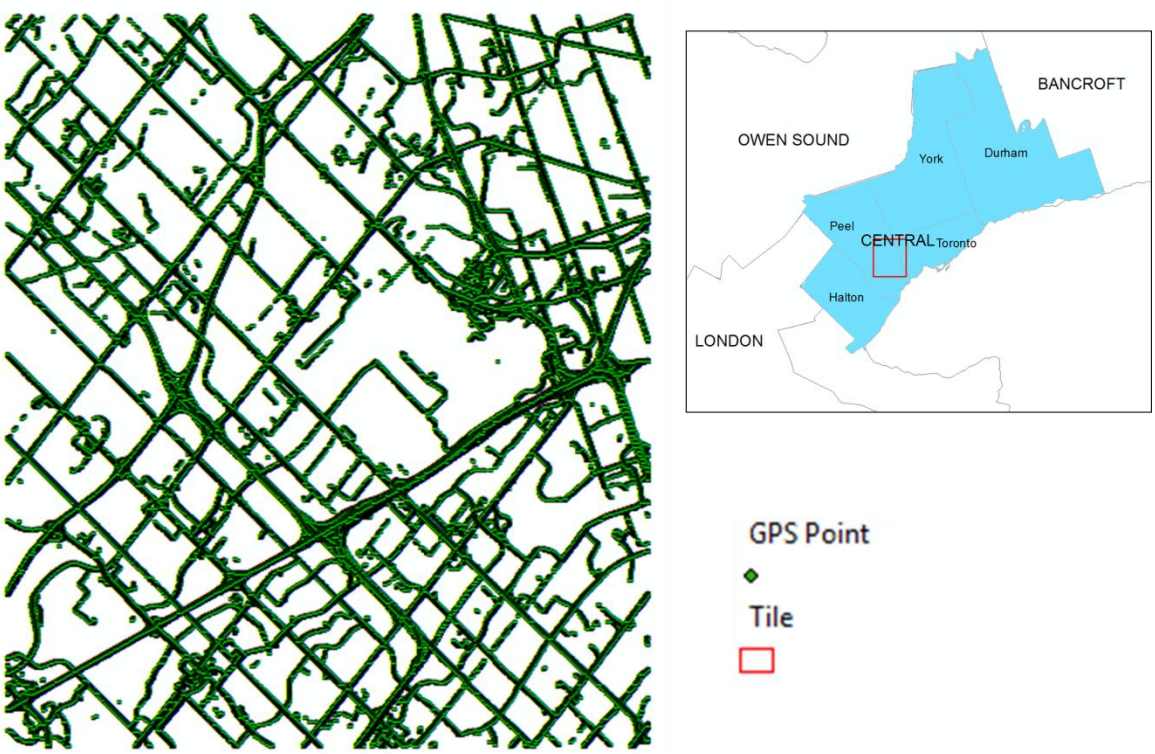


Figure 4.2: Study area and spatial distribution of GPS data of the 11th tile.

4.3. Accuracy of GPS-enabled Mobile Device

The accuracy of GPS data is affected by various factors. For instance, multipath errors could be caused by nearby buildings in local area, the signal reflected by the surrounding vehicles on highway, or/and satellite visibility (satellite geometry and weather condition). Other factors such as ionosphere delay and tropospheric delay were not considered in this project, because accuracy requirements for smart phones GPS cannot match with those of survey-graded professional GPS devices antennas. As has been well known, the more satellites that are available, the less likely to encounter poor Dilution of Precision (DOP) situation. It is difficult to determine the threshold value of accuracy for selecting optimal GPS data, because no standards regarding the accuracy of GPS chips utilized by iPhone and Blackberry have been disclosed by the manufacturers. Accordingly, the parameter of GPS data accuracy involved in the data preprocessing algorithm was selected by considering existing evaluations of GPS-enabled smartphones' accuracy.

To date, little quantitative information is available about the horizontal accuracy of GPS data obtained from the iPhone and Blackberry. No formal studies on the performance of the integrated positioning system (Assisted GPS, Wi-Fi, and Cellular network) of the iPhone were conducted, except a few iPhone blogs having posts based on the users' experiences without any scientific accuracy evaluation. Blackberry locates the user's position by using the combination of Autonomous, Assisted-GPS, and Cellular-network modes.

Table 4.1 lists the available studies of the positional accuracy of smartphone GPS data. The accuracy of Wi-Fi and cellular-network positioning is highly dependent on the density of the Wi-Fi access points and the cell towers. The minimum horizontal errors of Wi-Fi and Cellular-network are reported as 16 and 30 m, respectively. Therefore, Wi-Fi and Cellular network are rarely used for navigation purpose. Unlike the autonomous GPS directly receiving radio signals from satellites, Assisted-GPS establishes a GPS reference network to increase the positional accuracy in case of the weak signals from satellites. However, Assisted-GPS locations are less accurate than those from regular autonomous GPS enabled smartphones. Under the static outdoor testing condition Zandbergen (2009) addressed that the mean horizontal positional accuracy of Assisted-GPS locations is 7.7 m while the regular autonomous GPS can be less than two meters. For real-time traffic analysis, Menard and Miller (2011) concluded that

- "*94.38% of the time*[22] *the iPhone 4 report within 10 m and 50.66% of the time being within 5 m. (12000 sample data points)*" and
- "*93.22% of the time within 50 m and 45.92% of the time within 18 m. (26000 sample data points)*".

As of this writing, only Wiehe et al. (2008) addressed that the approximate six-meter horizontal accuracy of Blackberry's Assisted-GPS data was employed to rack the travel patterns of adolescents. Additionally, (Zandbergen and Barbeau, 2011) evaluated the reliability of Assisted-GPS on other types of smartphones (Sanyo and Motorola) and

[22] manually timed how long vehicle took to travel along a 1.59 km section of roadway

suggested that the smartphone GPS data can be qualified as the source of location information for the location based service applications. According to Canadian Radio-television and Telecommunications Commission (CRTC), a positional error within a radius of 10 to 300 m from the user's actual location is acceptable for most location based services.

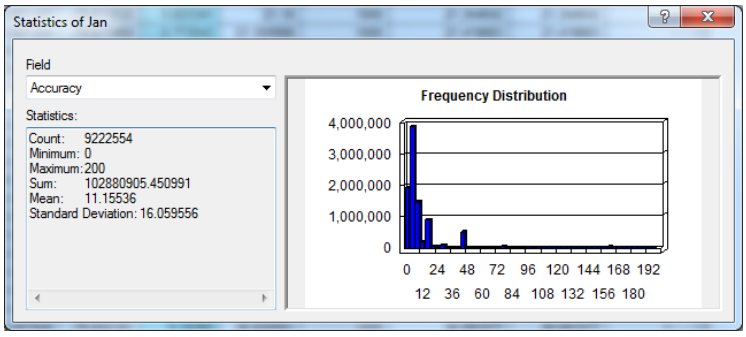
Table 4.1: Studies of the positional accuracy of smartphone GPS data

Mode	**Priority**	**Static outdoor test**		**Dynamic outdoor test**			
		Published	Unofficial	Driving			Walking
Assisted-GPS/Autonomous	***	iPhone 3G: Average Horizontal Median of 7.7 m	10 m	iPhone 3G: (2600pts) 93.22% within 50 m; 45.92% within 18 m of accuracy	iPhone 4: (1200 pts) 94.38% within 10 m; 50.66% within 5 m of accuracy	iPhone 4: (626 pts) 97.77% within 10 m; 58.63 % within 5 m of accuracy	iPhone 3G: (combination of three positioning modes) Accuracy varies from 9 to 47 m
		Sanyo SCP 7050 & Motorola i580: 95th percentiles of the horizontal error distribution from 10.26 to 23.90 m		Sanyo SCP 7050 & Motorola i580: 95th percentiles of the horizontal error varied from 4.04 to 8.51 m			
		Blackberry 7520: horizontal accuracy approx. 6m		Samsung Galaxy S: (652 pts) 97% of all data points within 5 m of accuracy; Motorola Droid X: (665 pts) 80.15% of all data points within 5 m; 95.94% within 10 m			
WiFi	**	iPhone 3G: Average Horizontal Median of 74 m	30 m				
Cellular	*	iPhone 3G: Average Horizontal Median of 599 m	500 m				

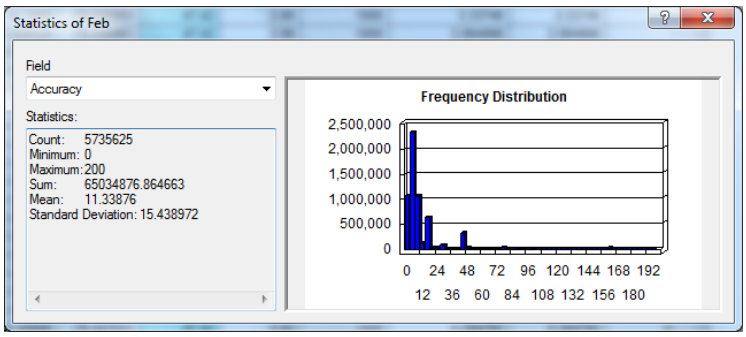
4.4. Raw GPS Data Analysis

The extracted road centerlines could be offset from the road geometry or twisted due to the inherent noise of the raw smartphone GPS data: off-road points (outliers) or point clouds. Therefore, it is necessary to improve the quality of input data as part of the automatic road network extracting algorithm. This section determines optimal universal parameters for data preprocessing based on the statistics of original collected smartphone GPS data, including accuracy and speed.

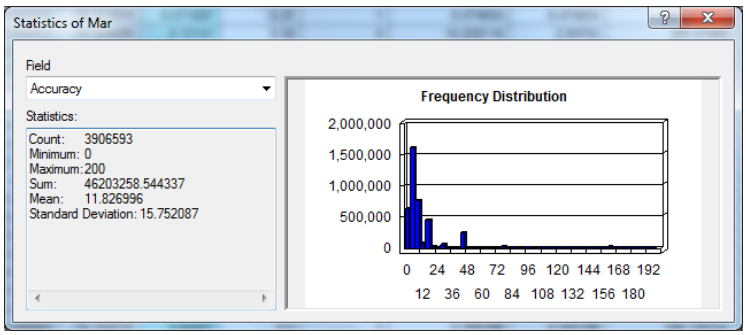
The threshold value of accuracy in the data preprocessing algorithm was developed based on the statistics of monthly raw GPS data accuracy. According to mean values of accuracy in Fig. 4.3 (a-f), it was found that around 72.8% of raw smartphone GPS data are in the range of zero to 11.28 m as shown in Fig. 4.3 (g). As aforementioned in Section 4.3, the dynamic accuracy of smartphone GPS data varies from zero to approx. 50 m depending on the model of smartphone while the static accuracy is up to 7.7 m. 17.2% of the low-accurate collected smartphone GPS data can be eliminated as a result of applying the threshold value of accuracy (11.28 m). Due to the lack of existing literature reviewing the accuracy of smartphone, the threshold value of 11.28m is used for selecting the majority of data input to serve the experimental testing.



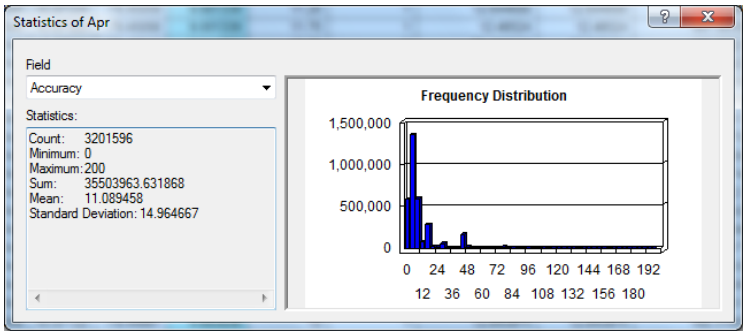
(a)


Statistics of Feb
Field
Accuracy
Statistics:
Count: 5735625
Minimum: 0
Maximum: 200
Sum: 65034876.864663
Mean: 11.33876
Standard Deviation: 15.438972
Frequency Distribution
2,500,000
2,000,000
1,500,000
1,000,000
500,000
0
0 24 48 72 96 120 144 168 192
12 36 60 84 108 132 156 180

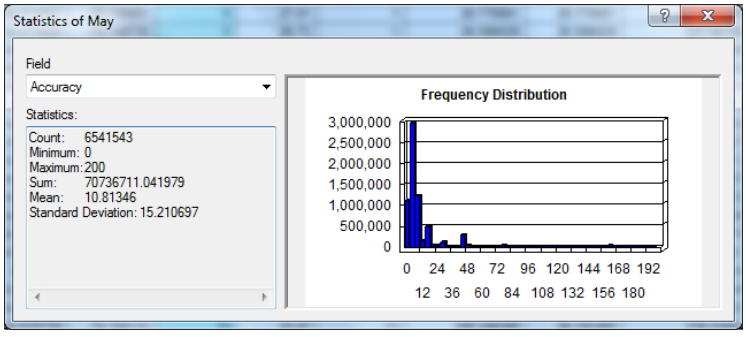
(b)


Statistics of Mar
Field
Accuracy
Statistics:
Count: 3906593
Minimum: 0
Maximum: 200
Sum: 46203258.544337
Mean: 11.826996
Standard Deviation: 15.752087
Frequency Distribution
2,000,000
1,500,000
1,000,000
500,000
0
0 24 48 72 96 120 144 168 192
12 36 60 84 108 132 156 180

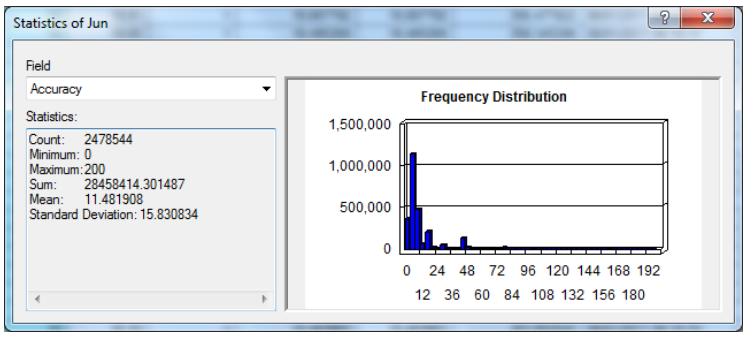
(c)


Statistics of Apr
Field
Accuracy
Statistics:
Count: 3201596
Minimum: 0
Maximum: 200
Sum: 35503963.631868
Mean: 11.089458
Standard Deviation: 14.964667
Frequency Distribution
1,500,000
1,000,000
500,000
0
0 24 48 72 96 120 144 168 192
12 36 60 84 108 132 156 180

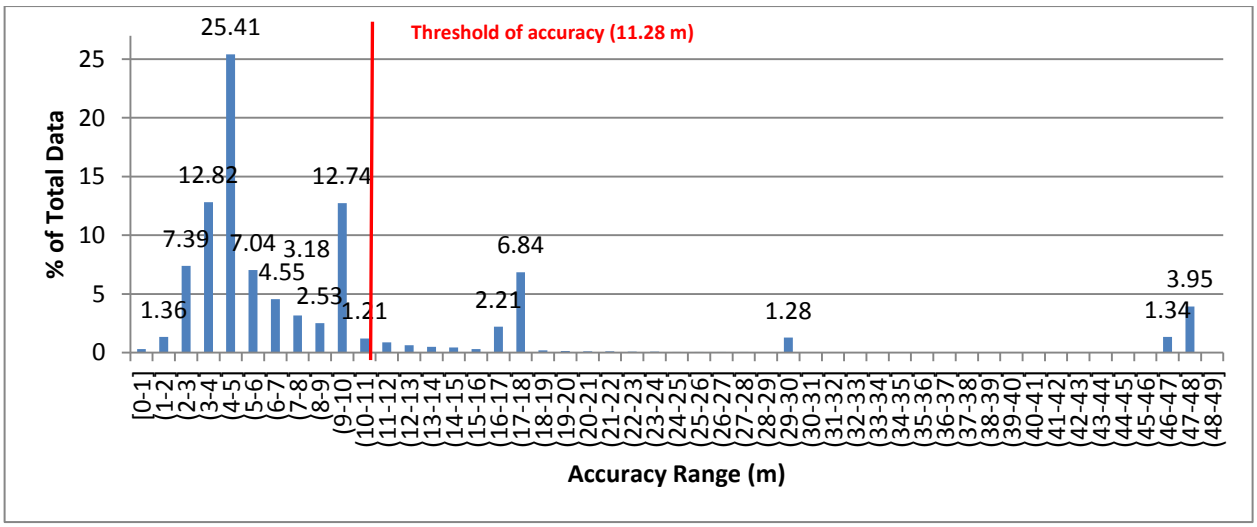
(d)


Statistics of May
Field
Accuracy
Statistics:
Count: 6541543
Minimum: 0
Maximum: 200
Sum: 70736711.041979
Mean: 10.81346
Standard Deviation: 15.210697
Frequency Distribution
3,000,000
2,500,000
2,000,000
1,500,000
1,000,000
500,000
0
0 24 48 72 96 120 144 168 192
12 36 60 84 108 132 156 180

(f)



(f)



(g)

Figure 4.3: Statistics of accuracy of collected smartphone GPS data.

Figure 4.4: Point clouds caused by users' slow-moving activities.

GPS point clouds could be generated because of the slow movement at particular locations, such as building (home, office, and shopping mall), parking lots, sidewalks, driveways, traffic lights, and traffic congestions. These activities can be categorized into two groups, locational or navigational activities. As shown in Fig. 4.4, GPS point cloud is generated due to the slow-moving traffic at the intersection, walking around at the same place, or idling at the parking lot. The patterns of users' GPS trajectories at such places were significantly different from those on roads and highways with normal movement.

In order to get rid of the GPS point clouds, the second filter of the data preprocessing algorithm was developed based on the speed information. A total of 11 datasets are manually sampled tile by tile from the original collected smartphone GPS data; each of sample datasets contained 10,781 GPS points which are densely located at significant places, such as road intersections, parking lots, or buildings. Most of the sampled GPS points were distributed in GTA as shown in Fig. 4.5. Speed distributions of 11 sample datasets were highly correlated to one another (see Fig. 4.6) and were usually skewed to the right. It thus indicated that there are a few large measurements (high

moving speeds) in the sample datasets. The average speed of the sample dataset (1.904 m/s) is selected as the speed filter in the data preprocessing algorithm, because it can remove the majority of noisy points from other sample datasets. Table 4.2 lists the results after applying such filter. For example, there are about 97.6% of 10,781 GPS points in the 2nd sample dataset with speeds less than 1.904 m/s are identified as noise; and overall 96.65 % of GPS points within point clouds are deemed as noise and then can be removed.

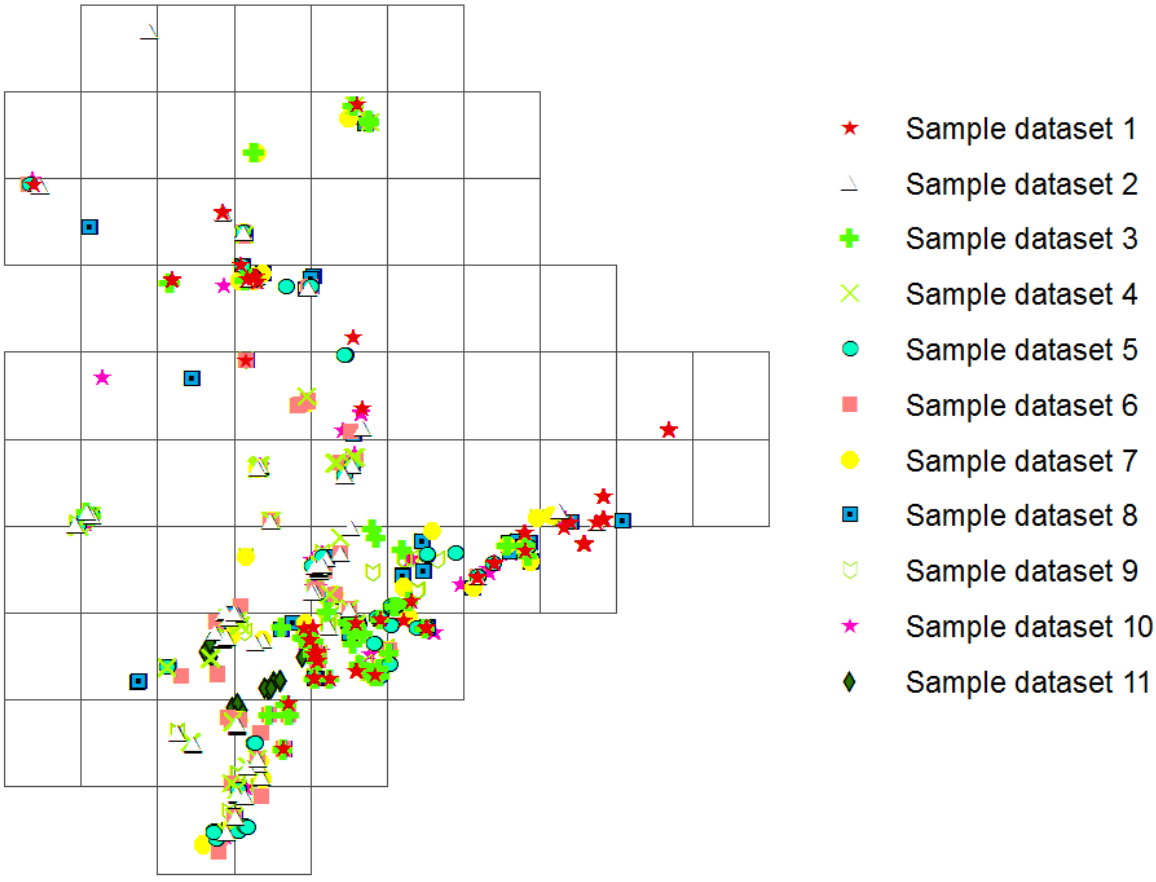


Figure 4.5: 11 sample datasets of point clouds from original collected GPS data

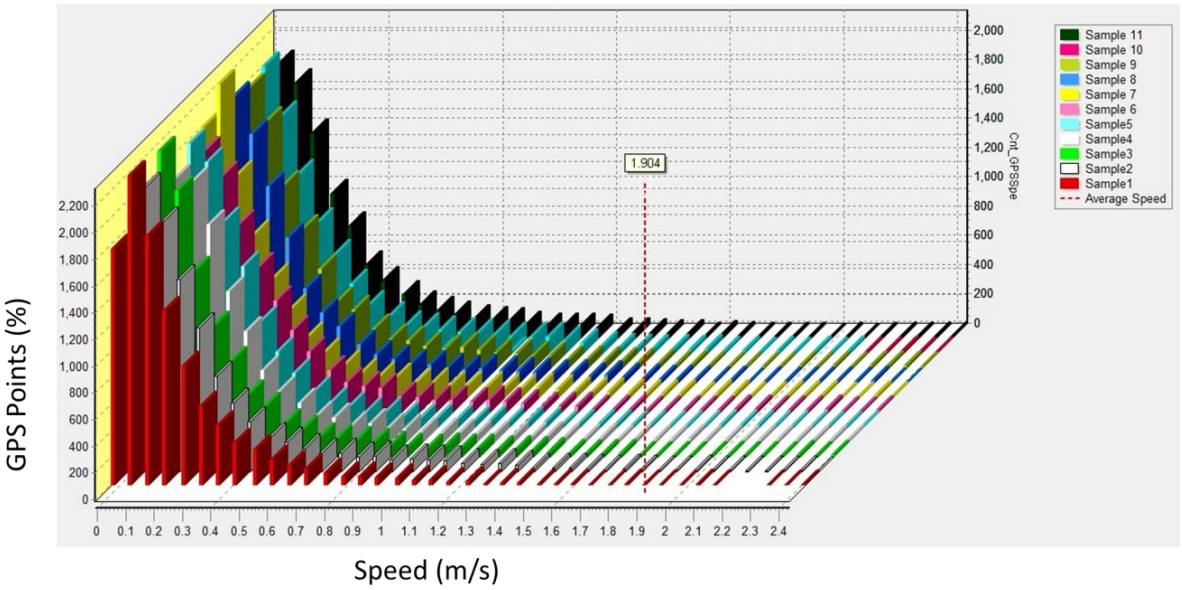


Figure 4.6: Correlations between speeds of 11 sample datasets

Table 4.2: Evaluating the threshold of GPS speed (1.904 m/s) on sample datasets

Sample Dataset	% Removed (≤1.904 m/s)	% Remaining (>1.904 m/s)
1	99.7 %	0.3 %
2	97.6 %	2.4 %
3	97.0 %	3.0 %
4	95.4 %	4.6 %
5	95.0 %	5.0 %
6	95.7 %	4.3 %
7	95.1 %	4.9 %
8	96.5 %	3.5 %
9	96.8 %	3.2 %
10	96.3 %	3.7 %
11	98.1 %	1.9 %
Overall	96.65 %	3.35 %

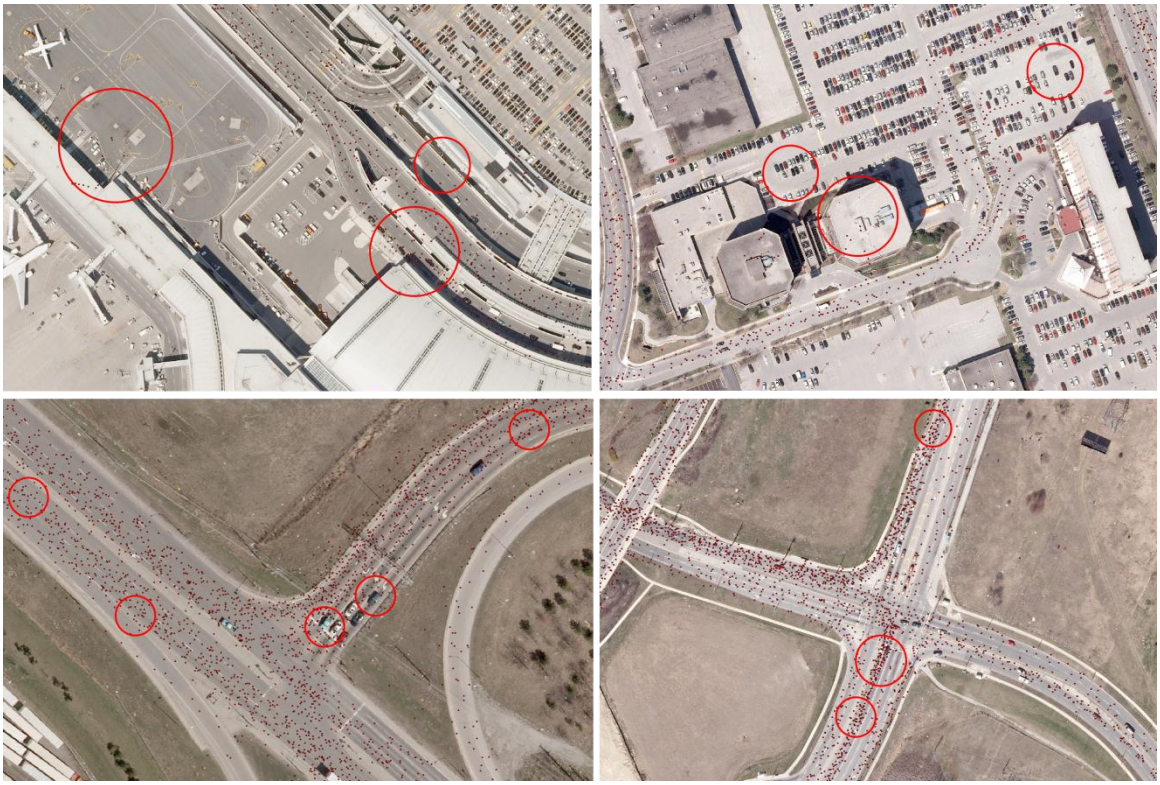


Figure 4.7: Removal of point clouds by applying speed and accuracy thresholds.

Fig. 4.7 illustrates the results after applying above two filters (speed and accuracy) to the same sample areas as shown in Fig. 4.4. Compared to the raw GPS points, it is clear that the noise in raw GPS points can be suppressed by the preprocessing algorithm. For instance, most of dense GPS points at the airport or closed to the intersection are eliminated because they are noise for the automatic extraction of road centerlines.

The distance threshold was referred to similar studies (Wang et al., 2011; Cao and Krumm, 2009). Instead of determining the best empirical value of direction threshold by a series of experiments (Li et al., 2012; Karagiorgou & Pfoser, 2012; Wang et al., 2011; L. Zhang et al., 2010; Cao and Krumm, 2009), it is inspired by the resolution of heading direction proposed by Liu et al. (2012) and derived based on the definition of degree of curve described by Ghilani and Wolf (2002) (in Eq. 4.1).

$$\frac{D}{360°} = \frac{100}{2\pi R} \; or \; R = \frac{50}{\sin(D/2)} \qquad \text{Equation 4.1}$$

where, D is the degree of circular curve; and R is the radius of circular curve depending on the superelevation and design speed, according to the Geometric Design Guide for Canadian Roads (TAC, 1999). Easa (2002) addressed that the recommended design superelevation is 0.04 for more recent developments of geometric design of urban freeways and high-speed urban streets, which is applicable to the road network in Great Toronto Area. The mean speed of vehicle passing through a circular curve such as the highway ramps is close to 60 km/h based on the data analysis. As shown in Fig. 4.8, GPS trajectories on highway ramps in the 11th tile are sampled for calculating the average driving speed (16.18 m/s or 58.24 km/h).

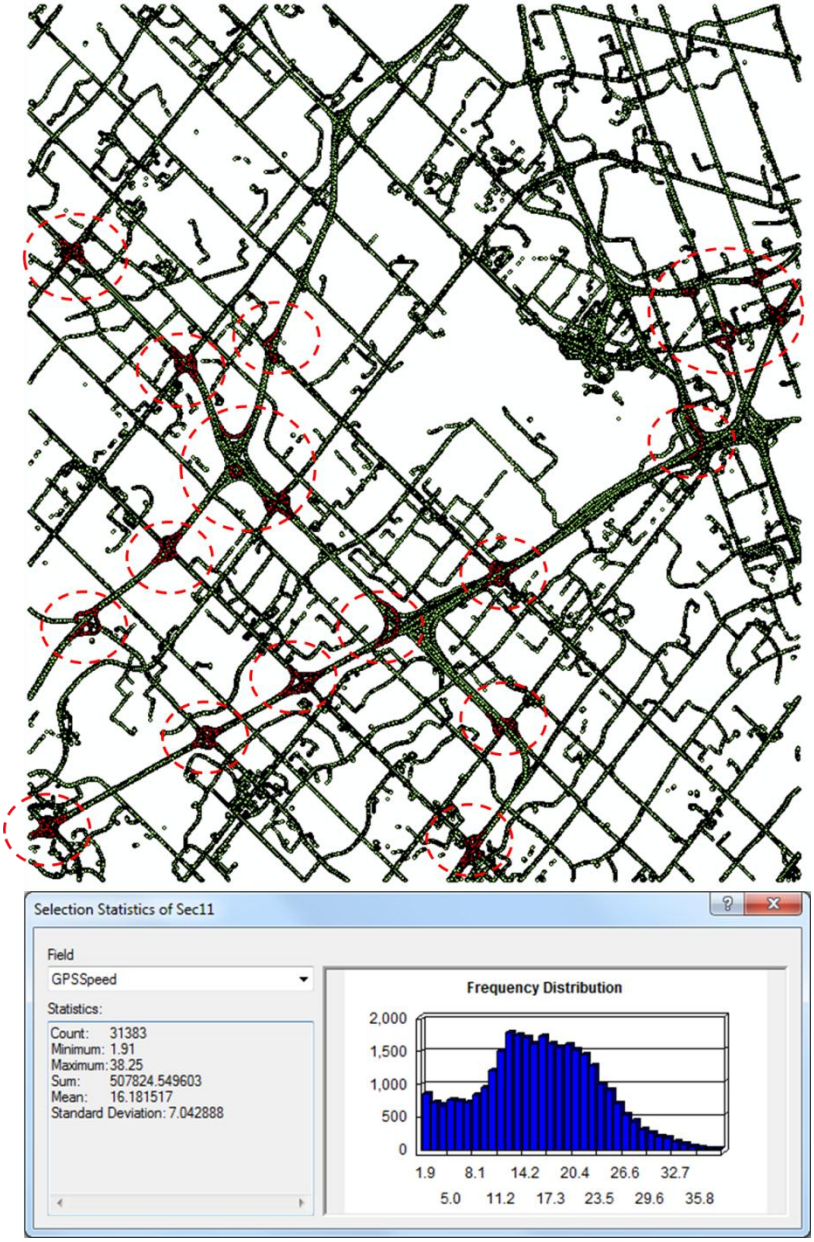


Figure 4.8: Sample GPS trajectories are used for calculating mean speed on highway ramps.

Based on the Table 2.1.2.5 in the Geometric Design Guide for Canadian Roads, it is illustrated that the minimum radius (R) of 150 m is required for superelevation value of 0.04 and speed value of 60 km/h. The value of D is calculated from Eq. 4.1 to be around 11˚. As illustrated in Fig 4.9, if directional change over two consecutive positioning points is larger than 11˚, the GPS trajectories is split into three trips. The first trip

includes Pt_1 and Pt_2; the second trip contains Pt_3, Pt_4, and Pt_5; and the third trip has Pt_6 and Pt_7 (see Fig 4.9).

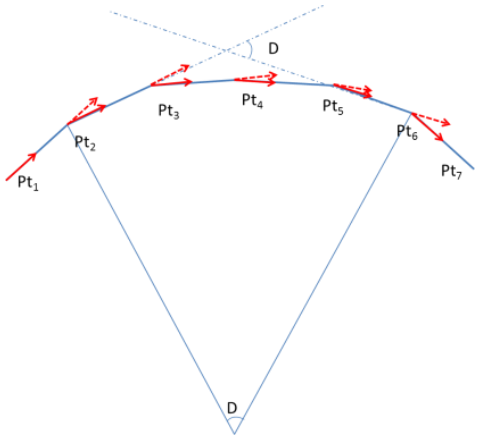
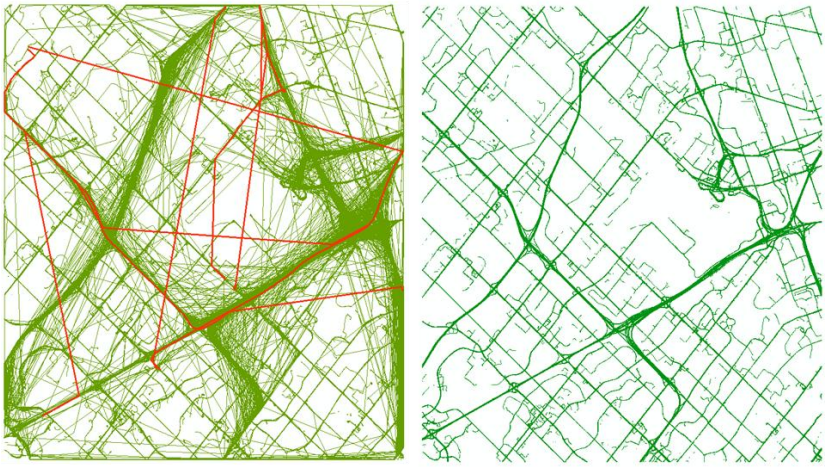


Figure 4.9: A sample GPS trajectory consists of seven time-stamped positioning points with different driving directions.

Fig. 4.10 (a) shows the unreasonable connections between any pair of consecutive GPS points that are off from the actual road network. Fig. 4.10 (b) presents the preprocessed GPS trajectories after data pruning by using four threshold values, including speed, accuracy, direction, and distance. There are 1,810,617 GPS points remaining in the 11^{th} tile after the preprocessing. Obviously, the preprocessed GPS trajectories reveal the better geometry of the underlying road network.



(a) (b)

Figure 4.10: Comparison of raw (a) and preprocessed (b) GPS trajectories.

Chapter 5. Results and Analysis

This chapter presents the results of applying the automatic road network extraction algorithm on the large and rich GPS datasets in different regions. After that, a quantitative evaluation of the horizontal accuracy of the generated road centerlines is provided by comparing them with the MTO highway horizontal alignment data. The chapter then finishes with the section analyzing effects of GPS point density on the accuracy of derived road centerlines.

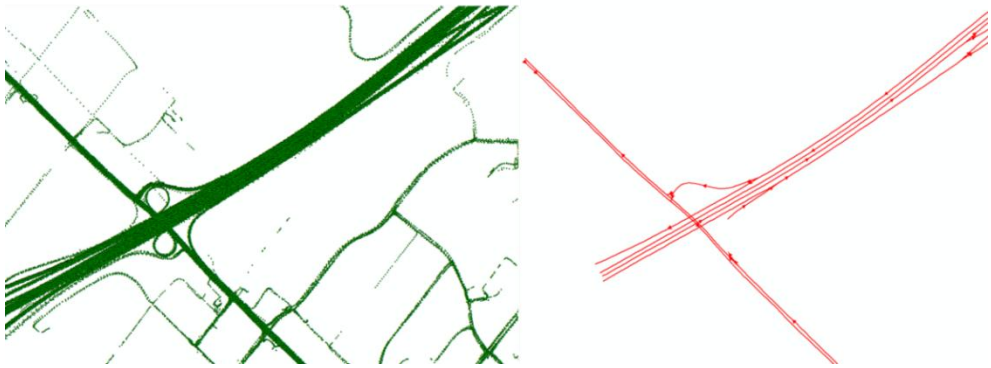
5.1. Experimental Results

This section tests the effectiveness of the proposed road network construction methodology by extracting road centerlines from the testing data in the study area. After applying the data smoothing algorithm, the amount of GPS data is reduced to 84.8 % (1,719,242 out of 2,026,251 GPS points in the 11th tile) of the total raw GPS points. Due to the 2-Gigabyte memory limitation to running 32-bit PythonWin on the 64-bit Microsoft Windows 7 operating system (OS), smoothed GPS data in the 11th tile cannot be completely processed by the later algorithm of representative point extraction, even though a patching application[23] is utilized to allow the OS addressing up to 4-Gigabyte of Random Access Memory (RAM). For the effective perspective, algorithms are implemented based on smoothed data within three typical regions in the study area.

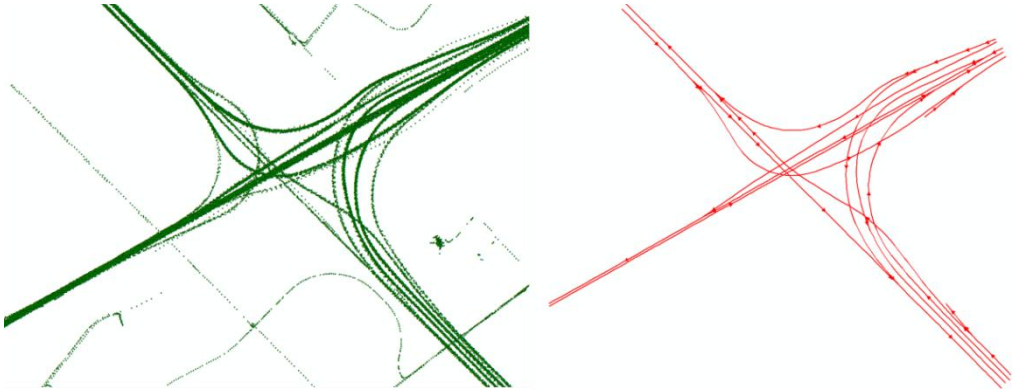
Fig. 5.1 shows the overviews of the collected GPS data and the constructed road network in three different typical regions. Region 1 is selected for testing the proposed methodology because it has typical parallel straight segments as well as intersections of highway ramps and major roads. In Region 2, there are highway interchanges connecting with straight segments. Region 3 is selected for further testing the adaptability of the methodology on additional complex highway than Region 2. It is apparent that centerlines of major roads and highways can be accurately extracted by using the proposed methodology. However, road centerlines of several minor roads and parts of

[23] Increasing the memory limit for 32-bit applications in Windows 64-bit OS: (www.maketecheasier.com/increase-memory-limit-for-32-bit-applications-in-windows-64-bit-os/2011/08/13)

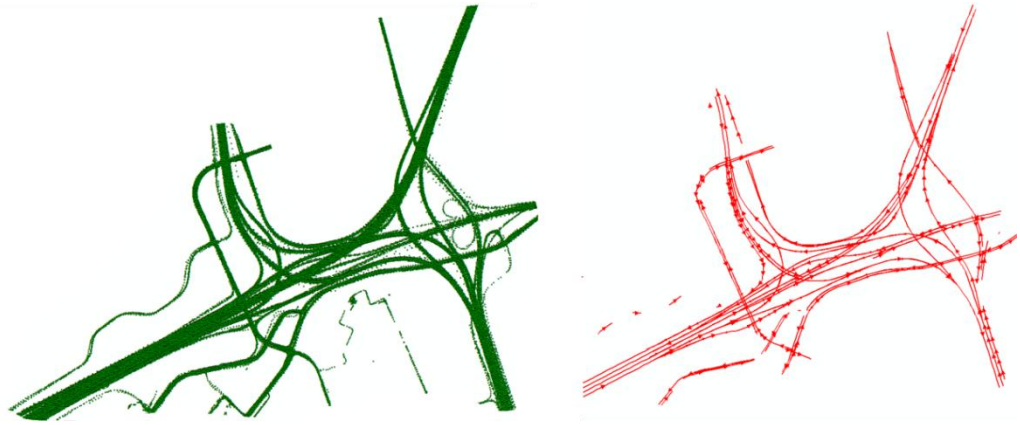
highway ramps are missing due to the fact that the fewer GPS data were collected. The detailed analysis regarding the effect of GPS point density to the result is provided in Section 5.3.



Region 1



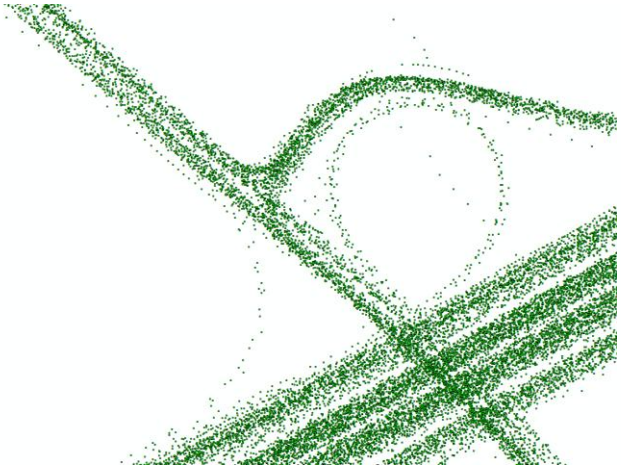
Region 2



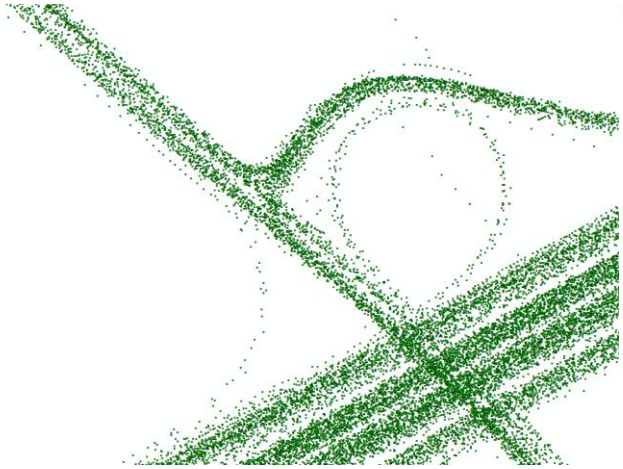
Region 3

Figure 5.1: Overviews of collected GPS data (left) and extracted road network data (right) on three typical scenes.

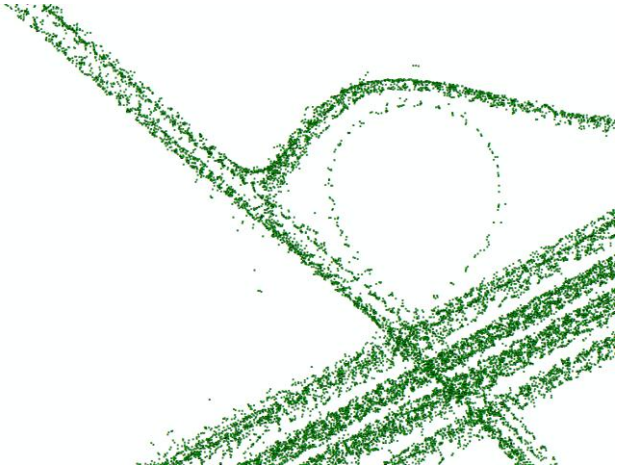
In order to introduce the process of extracting road centerlines more clearly and to better understand the results, Fig. 5.2 illustrates the result of each stage for the special road sections in detail, including the road intersection, straight segment of opposite directions, and Y-split section. After the first two processes (preprocessing and smoothing), duplicated GPS points (overlapping GPS points at the same location) were removed while the remaining GPS points are converged to the middle of the road. Therefore, the spacing among roads can be wider so that GPS points on different roads will be distinguished. However, some outliers of GPS points still remained at the road median section. It is not accurate enough to show the geometric characteristic of the underlying road network. The algorithm of extracting representative points was then applied to reduce the data size by extracting a smaller set of new points as representative of smoothed GPS points, while preserving the geometric shape of major roads. It is clear that the generated road centerlines captured the important direction and connectivity of the road network in each road section.



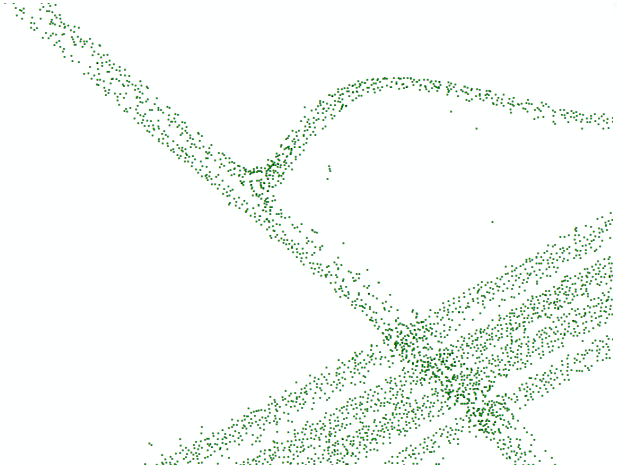
Raw GPS points



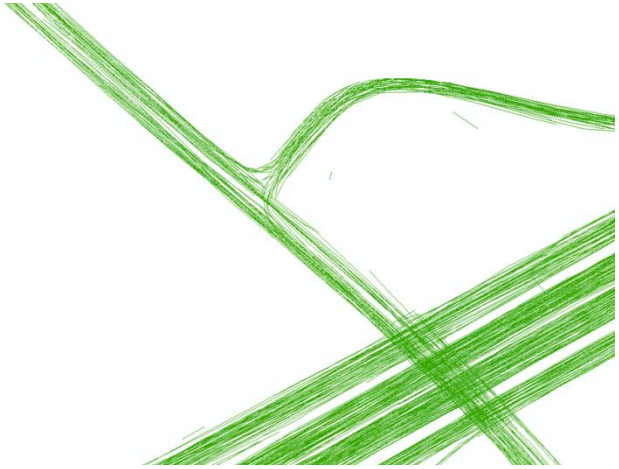
Preprocessed GPS points



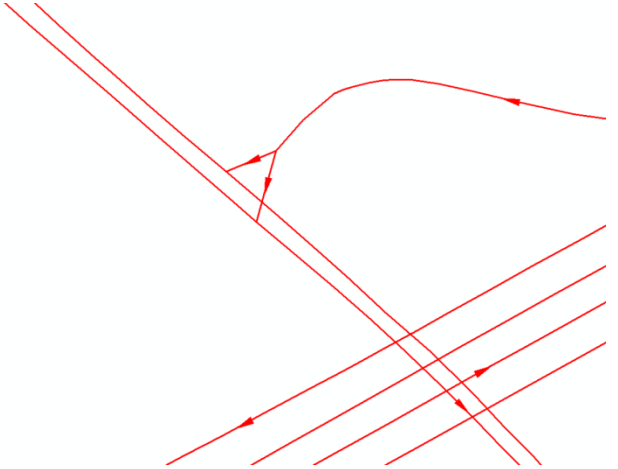
Smoothed GPS points



Extracted representative points

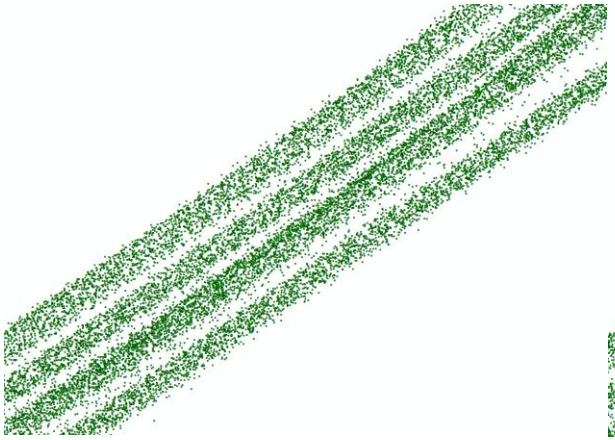


Reformed GPS trajectories

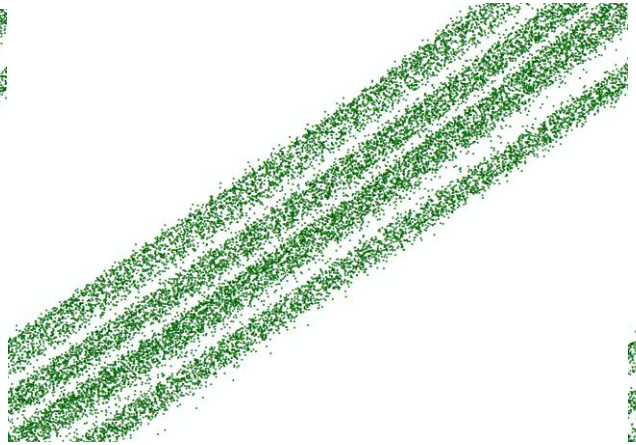


Merged road centerlines

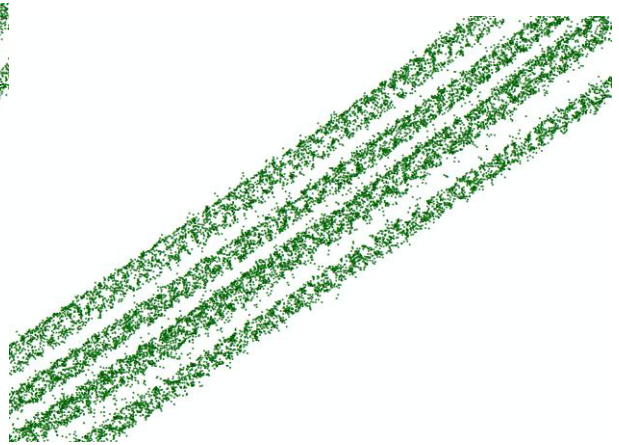
(a) Road Intersection



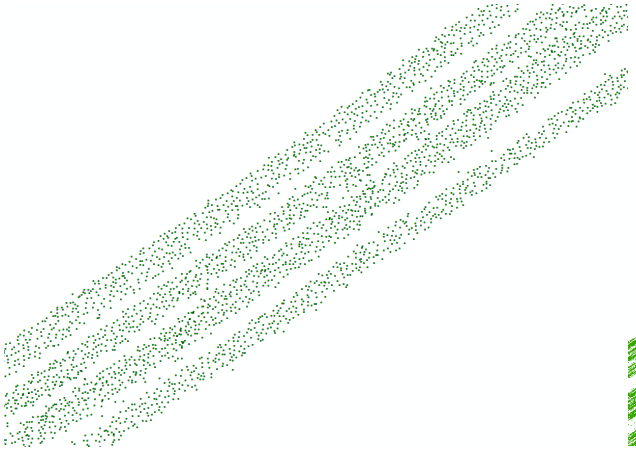
Raw GPS points



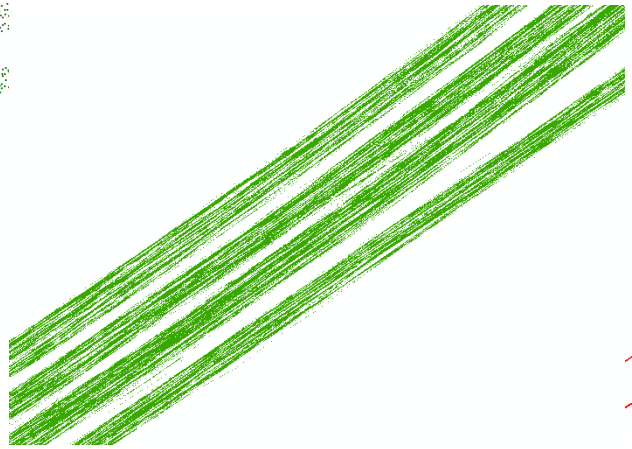
Preprocessed GPS points



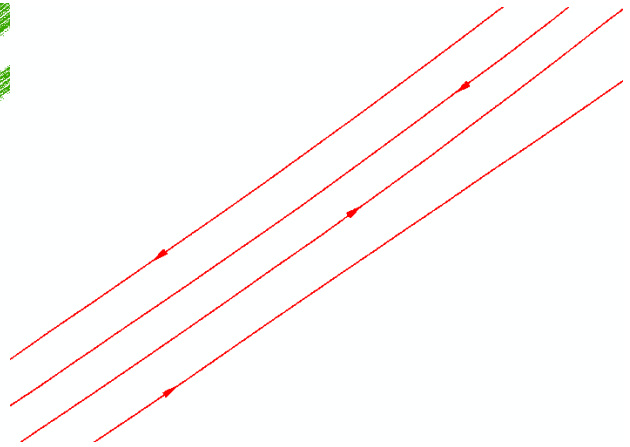
Smoothed GPS points



Extracted representative points

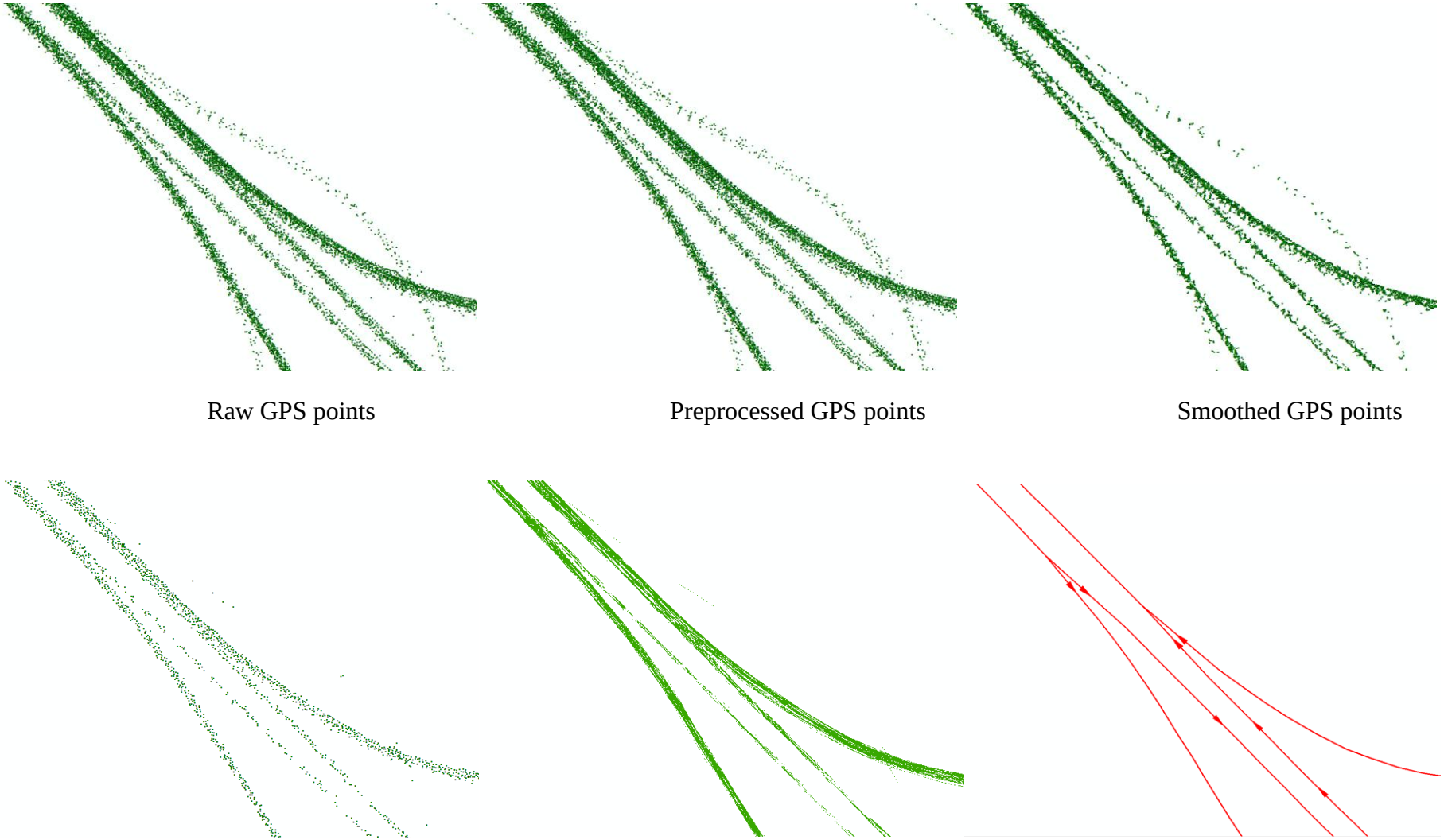


Reformed GPS trajectories



Merged road centerlines

(b) Straight Road Segment



Raw GPS points | Preprocessed GPS points | Smoothed GPS points

Extracted representative points | Reformed GPS trajectories | Merged road centerlines

(c) Y-split Road Section

Figure 5.2: A detailed view of results at each step of the proposed methodology at various road sections: (a) road intersection, (b) straight road segment, (c) Y-split road section.

It is impracticable to extract a road centerline by merging original GPS trajectories along the same road. As shown in Fig. 5.3, without the aforementioned processing, the noisy data in the point cloud occurred during the traffic congestion caused the bended GPS trajectories; and the noisy data (biased GPS measurement) caused a number of GPS trajectories offsetting from the road and overlapped with other trajectories of opposite directions. In order to generate road centerlines for roads of various complex geometric shapes, GPS trajectories along each road are reformed by linking representative points of similar movement on the same road.

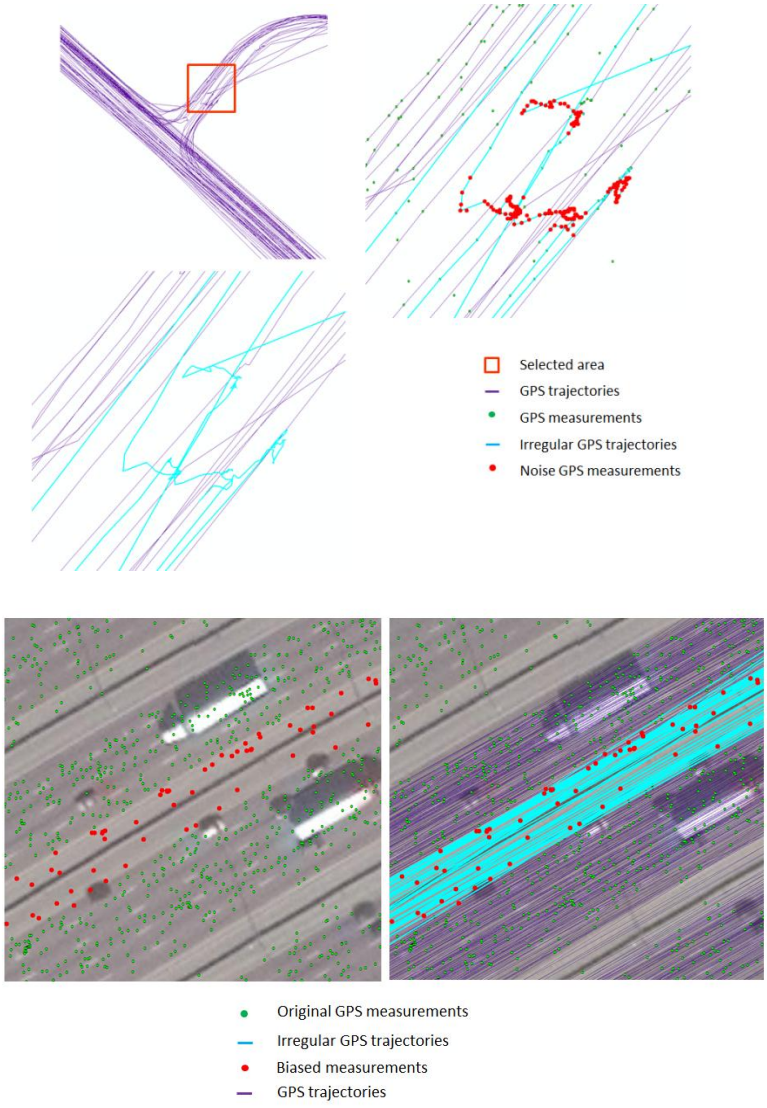


Figure 5.3: Deficiency of merging original GPS trajectories along the same road.

The searching radius (maximum distance amongst consecutive points) for linking representative points along the same road is important to the GPS-trajectory reforming. The optimal value is determined by testing different values of searching radius (as listed in Table 5.1) on the sample area with 12603 representative points. Table 5.1 contains the size of input table, the generated polyline segments, the number of reformed GPS trajectories consist of a set of polyline segments, and consumed time. The input data is an individual table that stores representative points with coordinates and directions and their neighbors within the search radius.

Table 5.1: Statistics of applying different values of search radius

Searching Radius (m)	Size of Input Data (# of rows)	# of Polyline Segments	Consumed time (sec)	# of GPS trajectories
10	61468	5776	349.67	2634
15	127518	8123	510.83	2048
20	206996	9150	568.57	1668
25	302136	9754	647.83	1460
30	414970	10192	870.81	1365
35	540898	10492	908.8	1288
40	677832	10701	979.7	1180
45	824432	10832	1212.01	1143
50	983466	10925	1315.99	1083
55	1157378	11029	1454.52	1063

Fig. 5.4 gives experimental results of linking representative points based on various values of searching radius. The longer the search radius, the more representative points are linked. The searching radius less than 20 m only provides a large number of discrete short connections. The number of long-distance links increases when larger value of searching radius is applied, but most of the connections are still in the form of sinuous polylines. The result is significantly improved once the value of searching radius is greater than or equal to 50 m, because more long-distance linear shaped links can be produced.

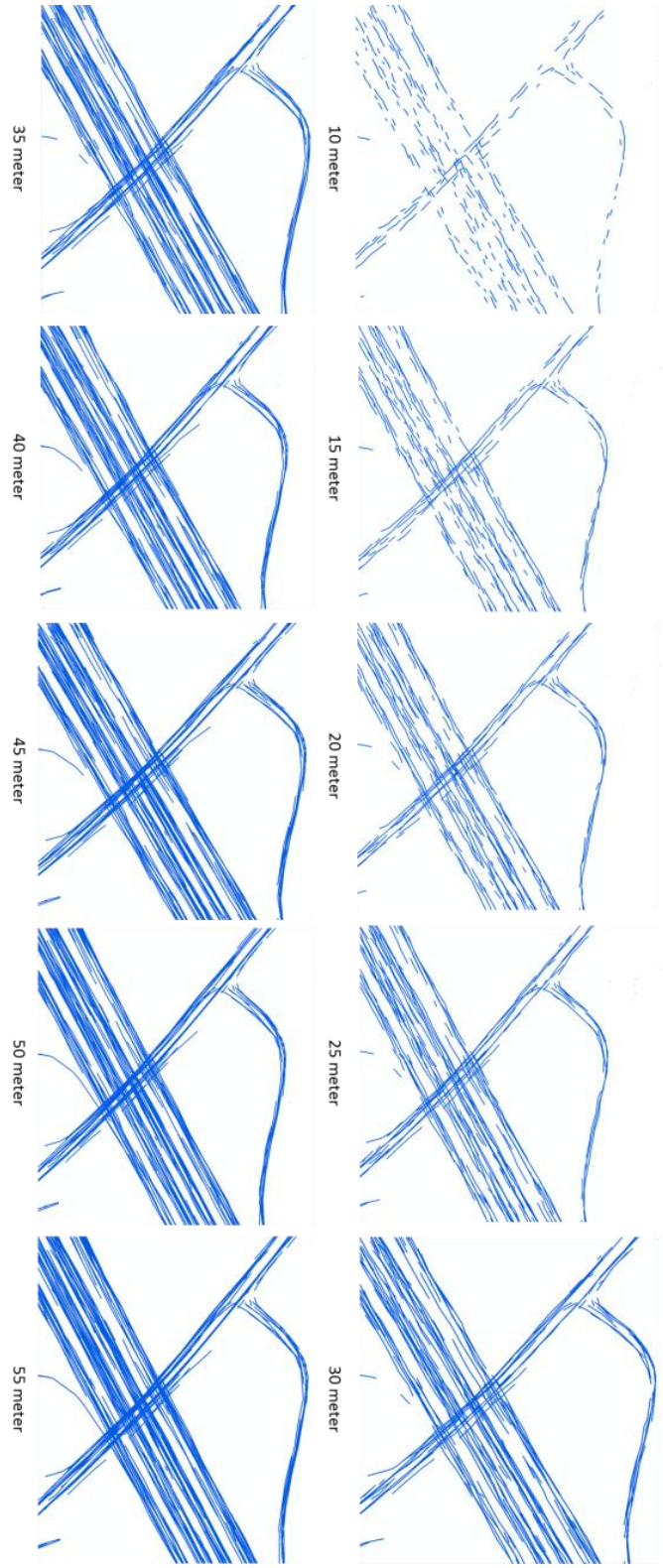


Figure 5.4: Reformed GPS trajectories based on different values of searching radius.

As shown in Fig. 5.5, the processing time is linearly dependent on the size of the input data. It took more than half an hour to generate new GPS trajectories from 12603 representative points when 55-meter searching radius was used. Recalling the definition of reforming GPS trajectories in Section 3.4, only the representative point with shortest distance to the current point can be connected if more than one representative point met another two requirements. Fig. 5.4 presents that performances of applying 50-m and 55-m searching radius have similar outputs showing dense converged trajectories on individual road. Making use of 50-m searching radius takes less time to implement the GPS trajectories reconstruction compared to using 55-m, and preserve the connectivity of reconstructed GPS trajectories at road Y-split sections (e.g. highway ramp merge to highway collector in Fig. 5.4). Therefore, 50-m searching radius deems to be a better trade off that conforms to the requirements of linking representative points, and provides an acceptable representation of road geometry for merging converged polyline segments on the same road.

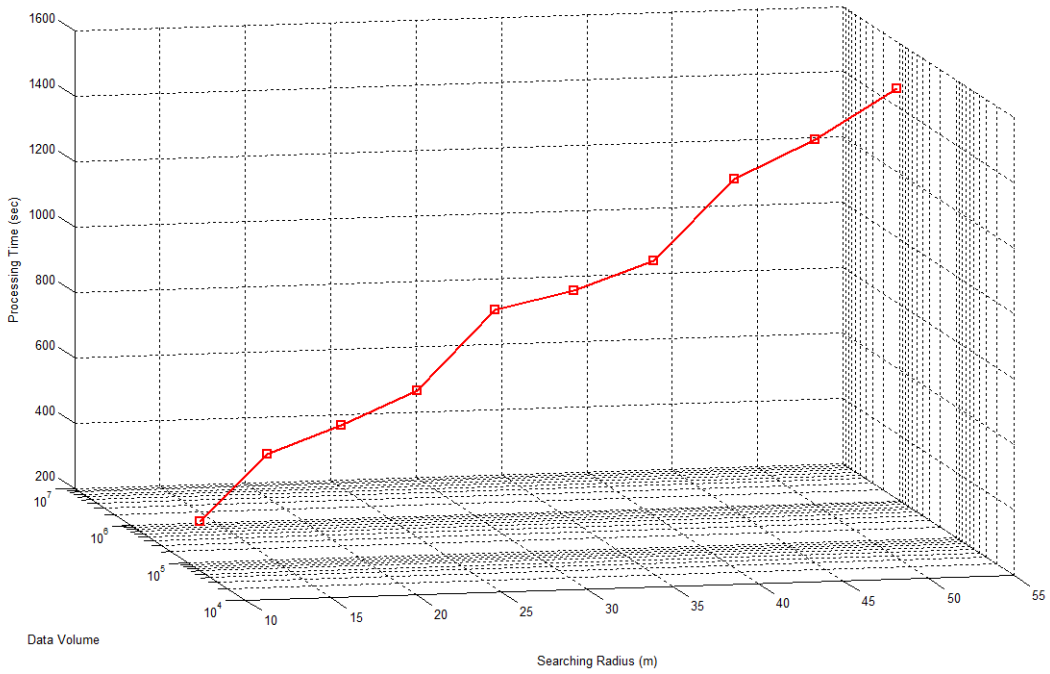


Figure 5.5: Testing the search radius for reforming GPS trajectories in terms of processing time and the number of input data

The search radius is only one of prerequisite for selecting candidate points to reconstruct GPS trajectory. The more concerns in GPS reconstruction algorithm are the geometric relationship between representative points and the topological relationship between representative points and the preprocessed trajectories.

5.2. Visual Inspection

Visual comparison of the extracted road network and aerial images shows that extracted road centerlines match well with road features in the aerial images, which are digital orthophoto image of 15-cm spatial resolution provided by MNR. Fig. 5.6 shows the overview of the correspondence between the extracted road network and road features in three classical regions. The majority of road centerlines can be obtained and matched with the geometry of road features except that several road centerlines are discrete at highway ramps or road intersections.

Figure 5.6: Visual inspection of extracted road network with high-resolution aerial orthophoto image.

Fig. 5.7 shows close-up views of extracted road centerlines at the road intersection, Y-split section, and highway interchanges. Bi-directional road centerlines, modeling each direction of travel as a separate alignment, can be extracted from massive GPS points for providing a "close-to-reality" presentation of the road network. However, the position of the extracted road centerline is slightly oscillated within the boundary of road feature in the image. For example, the extracted road centerlines are offset from the middle of the road curve in Fig. 5.7 (d) and (e), while most of them are close to their actual locations along the straight road segments in Fig. 5.7 (c). Another significant contribution of the proposed methodology is to generate the topological connectivity of extracted road centerlines at the road intersection and the complex Y-split section. As presented in Fig. 5.7 (a, b, and f), the ramp connections and road merging/splitting in support of turning information can be automatically modeled instead of manually modifying connectivity after the road network has been created.

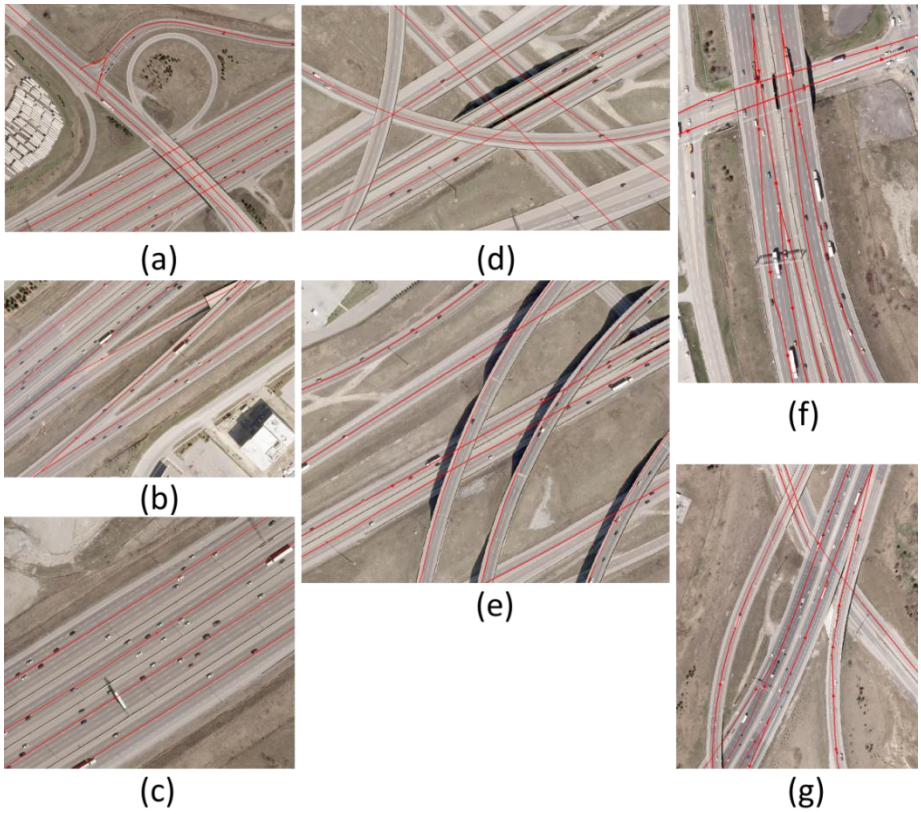


Figure 5.7: Close-up views of extracted road network overlaid with high-resolution aerial orthophoto image.

5.3. Quantitative Evaluation

The quality of road map in terms of positional accuracy must be assessed and justified, in order to produce a digital road map that meets with practical requirements such as navigation and mapping (Willrich, 2002). The effectiveness of the proposed methodology is investigated by assessing the horizontal accuracy of extracted road centerlines to the ground truth data. In order to achieve effective assessment of accuracy, extracted road centerlines in classic regions (in region 1 and region 2) shown in Fig. 5.1 are split into two sets of straight segments and curve segments. The highway horizontal alignment data (green-color line segments in Fig. 5.8) provided by the MTO are selected as the ground truth, because it provides information related to the geometric design of the highway, such as the length, direction, and position/layout of the centerline of the highway on the ground.

Figure 5.8: MTO highway horizontal alignment data (green) overlaid with 15-cm resolution aerial orthophoto.

The horizontal accuracy of extracted road centerlines is assessed by measuring difference from the ground truth data. Every extracted road centerline in two sets is split into a set of equidistant points. The spacing between consecutive points on the extracted road centerline is defined to be 10 m, in order to match with the geometric shape of the road. The difference is defined as the perpendicular distance from the point to the reference alignment (from MTO), while both of them belong to the same road. Fig. 5.9 demonstrates three samples of the reference alignments and the corresponding set of equidistant points in straight segment set and curve segment set. Equidistant points close to the road intersection or Y-split section (in dash-line circle of red color) are located farther from the reference alignment than the others along the straight road segment.

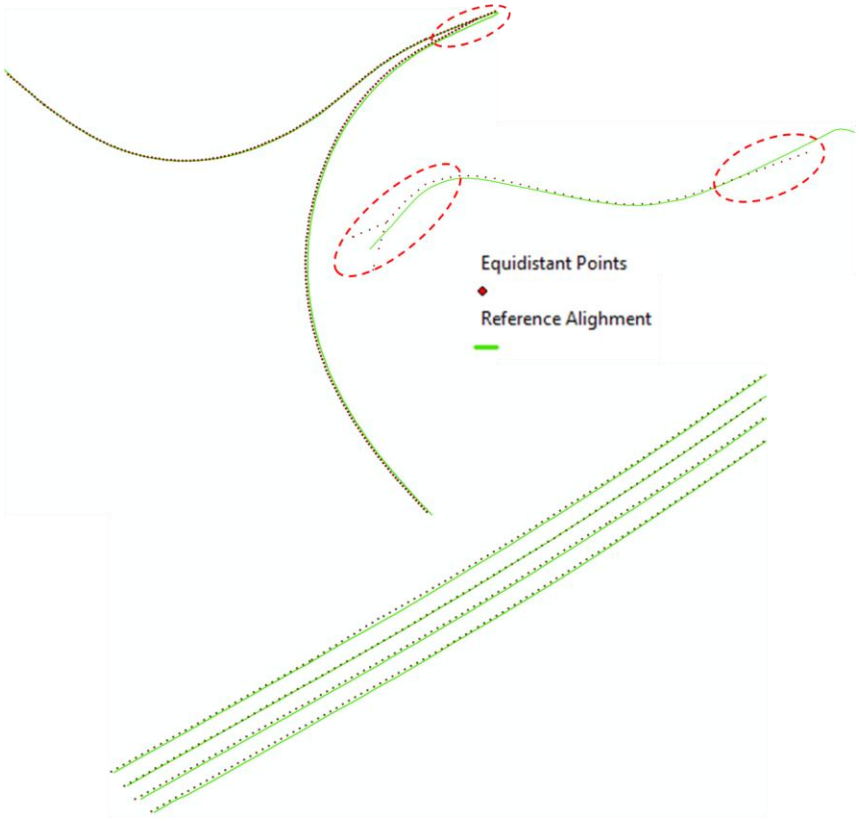


Figure 5.9: Illustrating the approach of evaluating horizontal accuracy of extracted road centerlines.

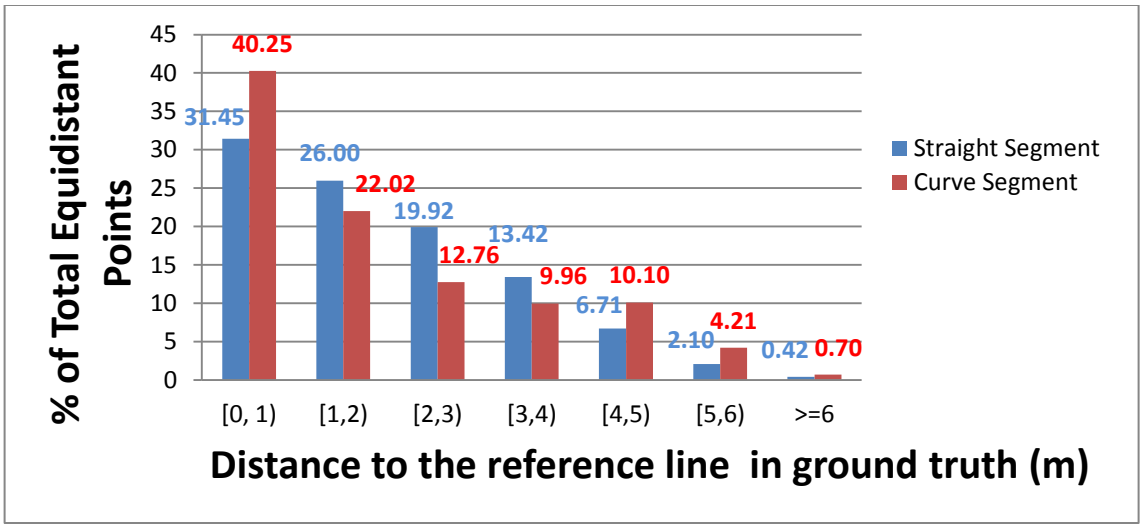


Figure 5.10: Difference between extracted road centerlines and ground truth.

The statistical summary of the quantitative evaluation is presented by using Fig. 5.10 and Table 5.2. Fig. 5.10 shows the differences between extracted road centerlines and the ground truth data in two sets: road curve segment and straight segment. In general, 90.78% of 951 equidistant points in the straight-segment set and 84.99% of 313 equidistant points in the curve-segment set are within four-meter range of the ground truth. The extracted road centerlines in the straight-segment set (59.37%) are closer to the ground truth than those (44.74%) from the curve-segment set, within the range from 1 to 4 m. However, there are less equidistant points from the straight-segment set than those of the curve-segment set in the one-meter buffer of the ground truth. It is concluded that accuracy of the extracted road centerline on road curve segment is higher than that of the straight road segment within one-meter range of the ground truth.

Table 5.2 summaries the quantitative evaluation in terms of distance measurement from equidistant points to reference alignment. There are only 0.7% and 0.42% of equidistant points from both sets are far from the ground truth (over 6 m), while the maximum errors from both sets are 6.123 m and 6.849 m for the curve segment set and straight segment set, respectively. The root-mean square error (RMSE) is used to measure the horizontal accuracy of the result because it is frequently used to measure the difference between predicted value and actual value. The overall horizontal accuracy of

extracted centerlines measured by RMSE for 95% of the result is 1.424 m in road curve set and 1.252 m in straight road set, respectively.

Table 5.2: Horizontal accuracy of extracted road centerlines based on the ground truth data

	Curve Segment (meter)	Straight Segment (meter)
Minimum Error	0.002	0.007
Maximum Error	6.123	6.849
Mean	1.870 (1.87)	1.953 (1.95)
Standard Deviation	1.593 (1.59)	1.390 (1.39)
RMSE (95% of the results)	1.424	1.252

5.4. Effects of GPS Point Density

The experimental results cover majority of major roads and highways within three selected regions in the study area which has the largest volume of collected smartphone GPS data. The extracted road centerline is sometimes discrete on the highway ramp and found missing on the local road, because the Traffic Alert users do not normally use the application when driving along the local roads. This section analyzes the effects of GPS data point density on the extraction of road centerlines.

Fig. 5.11 illustrates the calculated magnitude per square meter from raw GPS data that fall within a 3x3 rectangular neighborhood around each square-meter area. The color of dark green denotes the high-density areas though where vehicles are frequently travelling. The lighter color areas mean fewer opportunities for vehicles travelling on roads. Therefore, it is inferred that the smaller the magnitude of point density per square-meter area, less chance of extracting the road centerline.

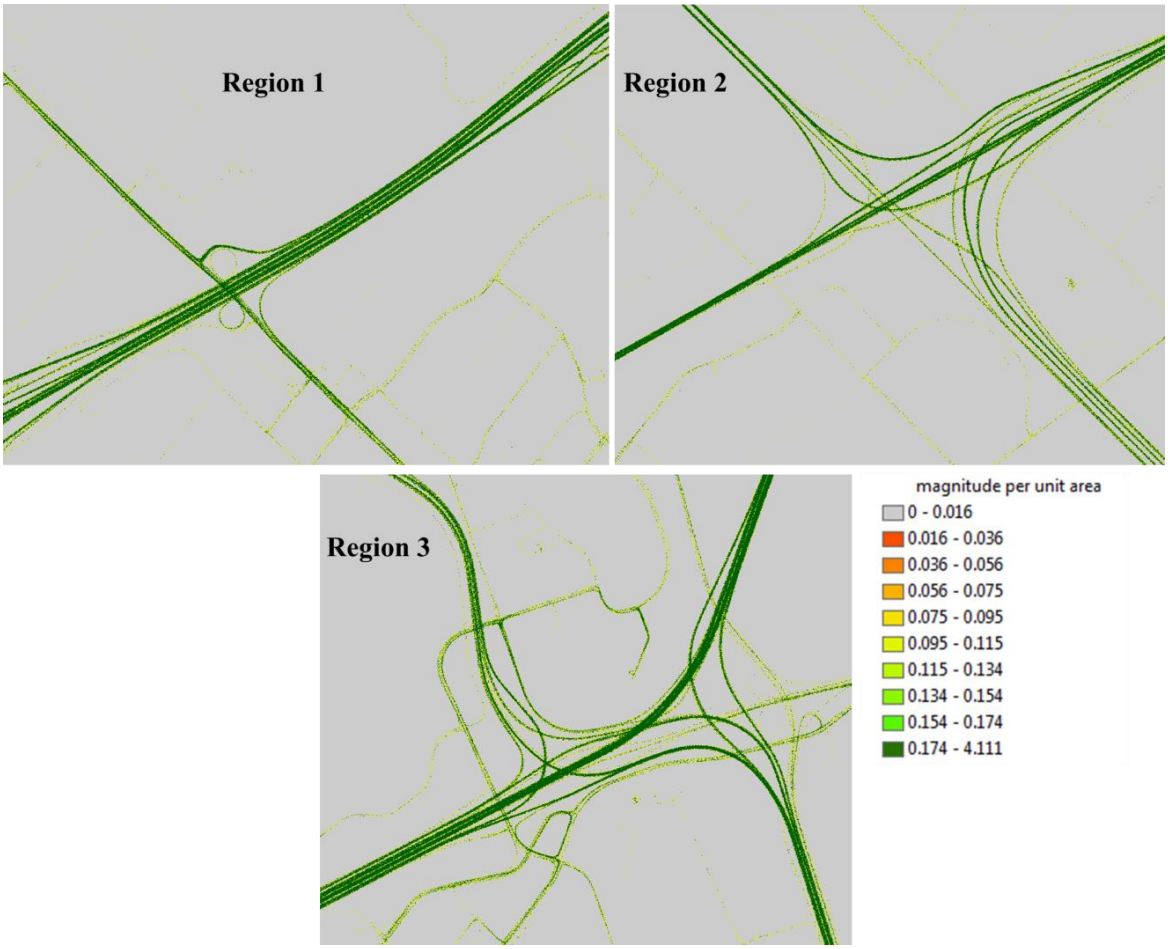


Figure 5.11: Point density of GPS data within three different regions.

To better understand this inference, Fig. 5.12 shows the effect of point density on the coverage of extracted road centerlines by using a partial cloverleaf interchange in the first region as an example. In the close-up view, there are four direct ramps and two loop ramps. The direct ramp connecting the highway and the major local road has more chance of having vehicles travelling on than another direct ramp (merging to the highway from the major local road), even though their point densities are within the same range (0.174 – 4.111). Therefore, a continuous road centerline can be extracted for the existing direct ramp while a discrete road centerline was generated for the emerging direct ramp. In contrast, there is less chance to construct road centerline for other direct ramps and loop ramps due to low point density of GPS data collected. Nevertheless, such problem can be resolved if the GPS data used were collected from a longer period. Taking the loop ramp

and the direct ramp (in black dash-line circles) as examples, they are not qualified for generating road centerlines because the number of the reconstructed GPS trajectories does not meet with the requirement of polyline-segment clustering algorithm in Section 3.5.

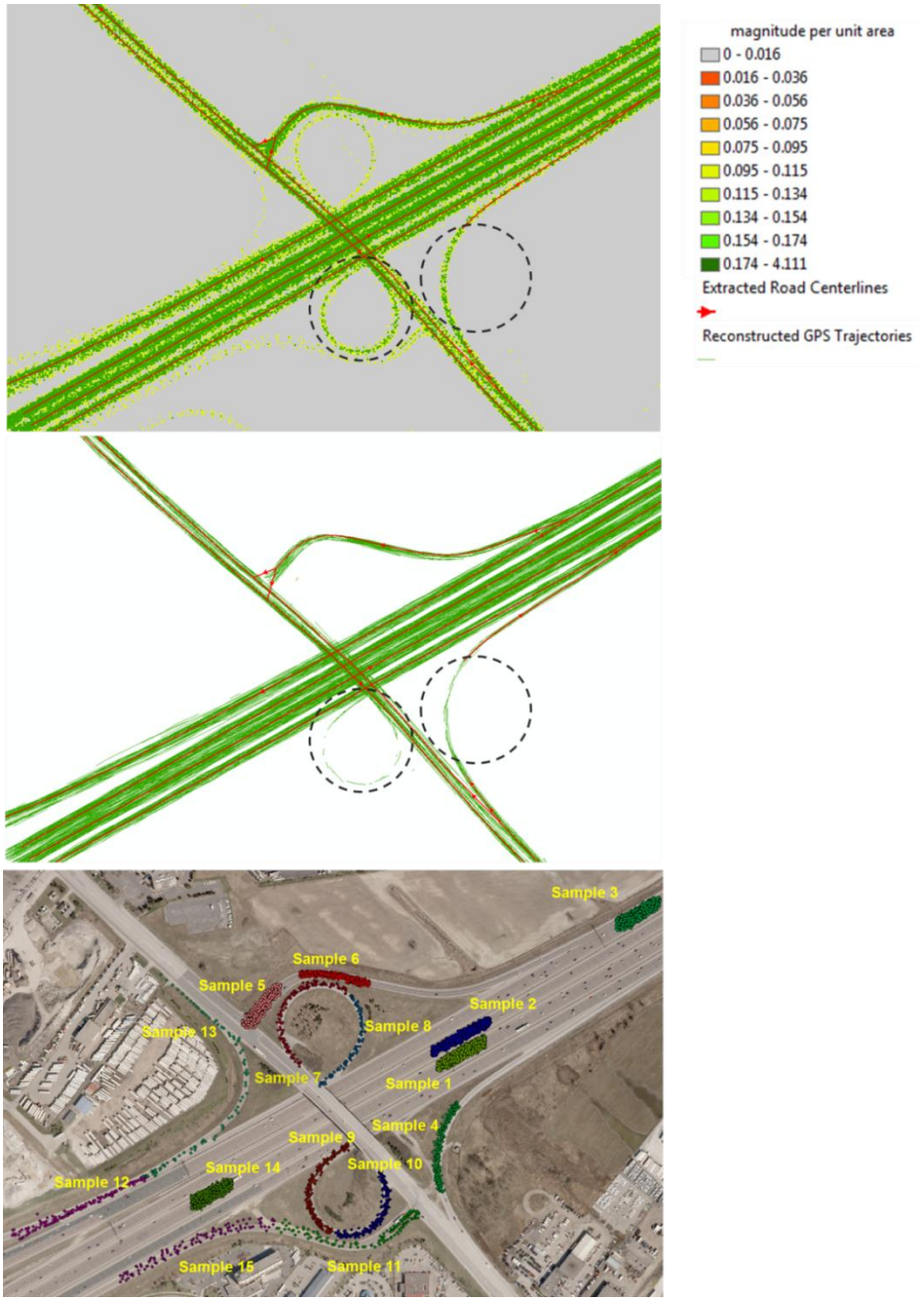


Figure 5.12: Close-up view of the effect of point density on extracted road centerlines.

It is necessary to determine the minimum point density of raw GPS data (number of points per square meter) which ensures the successful extraction of road centerlines. As shown in Fig. 5.12, 15 sample GPS data were selected from these three road types, direct ramp, loop ramp, and straight road segment. Each sample contains similar value of square-meter area (1600 m^2) except that sample 8 on the loop ramp has relative small value of area (944 m^2). Table 5.3 summaries the number of points and point density for each sample. The minimum value of point density required for this proposed methodology is 0.3 points per square meter which is within the calculated range (0.174 – 4.111) as shown in Fig. 5.12. In addition, the discrete road centerline is generated for direct ramp (black dash-line circle) due to that the point density of the sample 4 is smaller than the threshold value. Therefore, the minimum point density (0.30 points per square meter) must be utilized in the data preprocessing in order to generate the completed coverage of extracted road centerlines, which captures the most important connectivity and geometry properties of the actual road network.

Table 5.3: Summary of point density of sample GPS data on different types of roads

Sample	Points	Polygons (1 sq. m)	Road Type	Density (pts/sq m)
2	958	1600	Straight	0.60
1	901	1600	Straight	0.56
14	741	1632	Straight	0.45
3	627	1600	Straight	0.39
5	615	1600	Direct	0.38
6	474	1600	Direct	0.30
4	259	1600	Direct	0.16
10	213	1600	Loop	0.13
9	197	1600	Loop	0.12
12	140	1600	Straight	0.09
8	81	944	Loop	0.09
11	134	1600	Direct	0.08
7	124	1600	Loop	0.08
13	86	1216	Direct	0.07
15	108	1600	Direct	0.07

Min. point density for extracting road centerline

Chapter 6. Conclusions and Future Work

This chapter summarizes the work described in previous chapters and recapitulates the research objectives mentioned in Chapter 1 and how they were achieved, followed by a list of contributions to the subject of road network extraction based on GPS trajectories. The future work which could improve the research outcomes is then discussed.

6.1. Conclusions

As depicted in Chapter 1, the GPS trajectories collected from smartphones have been recently used to extract road geometric data for road network database updating and road maps refinement. This new approach entails a fast and inexpensive way of updating existing road maps and refining road maps with near real-time changes (e.g., new roads not showing in the existing road network database). Nevertheless, the presence of inevitable GPS noisy point clouds and uncertain distribution of smartphone GPS trajectories cause major obstacles for automatically extracting road network without using a reference road map.

As discussed in Chapter 2, the main challenge of existing methods is how to automatically generate road centerlines that can effectively capture both accurate geometry and connectivity of the actual road network. Some of the factors, such as data reduction without affecting underlying road network extraction, spatial clustering algorithms that distinguish nearby parallel roads and road splits are subject to various parameters, topological and geometrical relationships for generating accurate road junctions, and constructing bi-directional road centerlines, have not yet been fully investigated.

In this regard, this thesis demonstrated an attempt to fill the current gap by developing the new automatic methodology and prototype for extracting road network data from smartphone GPS data. This research focuses on integrating the modified conventional point- and polyline-based approaches to automate the entire process of accurately extracting road network without using any reference map. The proposed method includes five stages:

1) The preprocessing filters and weighted-mean smoothing algorithm were applied to remove extraneous, duplicated or inaccurate GPS data points from raw smartphone GPS trajectories.
2) The modified density-based point clustering method was applied to extract representative points along each lane on the road.
3) The representative points belonging to the same lane were connected based on their topological relationships and directions, in order to remain faithful to the underlying road network geometry.
4) The road centerlines were derived by using customized density-based polyline segment clustering method to merge those reformed GPS trajectories, which share the same geometric attributes on the road.
5) The road centerlines were topologically connected together to generate a completed road network.

The contributions of this thesis research are to overcome the main challenges found in similar studies as discussed in Chapter 2.

1) Overall 96.65% GPS data in point clouds are deemed to be noise and can be removed by utilizing the speed filter derived from the raw GPS data analysis.
2) GPS trajectories on parallel roads of similar direction or the same road of opposite directions can be clarified by applying the modified 4-meter circular window smoothing and the customized density-based point clustering algorithms that take the accuracy of GPS data point as the weight.
3) The bi-directional road centerlines can be extracted from a set of sequential polyline segment clusters on the same road of opposite directions, if collected GPS data meet with the minimum point density requirement. The extracted road network data can be added into the digital road map which meets with practical requirements (e.g. navigation and linear referencing system), because each extracted road centerline contains geographic location (e.g. starting and ending positions) and corresponding attributes (e.g. moving direction and turning direction).

4) The polyline segment cluster is formed by using polyline-based clustering algorithm among reconstructed GPS trajectories. Unlike those similar studies as reviewed in Section 2.2.2, the polyline-based clustering algorithm is an innovative approach to further identify GPS trajectories on different roads and group the same-road GPS trajectories together without using any reference map.
5) The connectivity of extracted road centerlines can be automatically derived based on the semantic road-knowledge based rules of their endpoints as well as the topological relationship among endpoints of extracted road centerlines and the underlying reformed GPS trajectories. By means of the proposed method, there is no need for manually linking extracted road centerlines at the road junction or Y-split section.
6) The estimated road width is the byproduct of the polyline segment clustering algorithm. The position of extracted road centerline can be incrementally updated if recent collected GPS data are within its boundary.
7) A GIS-based software tool can be developed for generating the road network data based on smartphone GPS data, because each procedure presented in Chapter 3 is developed as a standalone python script tool for ESRI ArcMap. There is no manual operation required, since these tools are sequentially strung together where feeding the output of one tool as the input for another tool.

As of this writing, none of the aforementioned studies has performed quantitative evaluation by comparing the results with the actual ground truth maps (MTO highway alignment data). The visual inspection presented in Chapter 5 demonstrates that majority of road centerlines can be obtained and matched well with the geometry and connectivity of road features. Moreover, the horizontal accuracy of extracted road centerline measured by RMSE is 1.424 m and 1.252 m for curved road segments and straight road segments, respectively. The maximum offsets from the ground truth data are 6.123 m and 6.849 m for the curve road segments and straight road segments, respectively. The experimental results are more accurate than ESRI North American detailed street dataset (from sub to

12 m)[24] and Ontario Road Network (up to 10 m) [25] published by Land Information Ontario (LIO), even though they are derived by using high-quality data from the road authorities such as TomTom North American, Inc., municipalities, and MNR. Therefore, it is proven that the proposed methodology can output the high-quality road network data if sufficient amount of smartphone GPS data possibly of low quality is provided and the minimum point density of source data is satisfied. To conclude, this thesis addressed an effective and automatic way to build detailed road network based on large-scale and coarse-grained smartphone GPS data without using any reference map data.

6.2. Future Work

This thesis research provides a foundation for developing an automatic self-learning GIS application for updating existing road maps and refining road maps with real-time changes. Some possible extensions of the proposed methodology are listed as follows:

Data cleaning: This thesis proposed to incorporate the derived threshold values of speed and the change of moving direction in the data preprocessing algorithm. The accelerated speed and time gap between any two consecutive positioning points could be other optimal parameters that should be taken into account. Since GPS data were collected with the sampling rate of one second, point clouds could be formed if the acceleration of one vehicle is relatively slow on the road during a certain time period.

Road information: Additional road associated information can be addressed based on the estimated road width and movement patterns, such as the number of lanes, speed limit, and road restriction, etc. For example, the number of lanes on the road is able to be estimated based on the known road width on every road segment and the approximate land width defined in the Geometric Design Guide for Canadian Roads. Mining driving directions, road width, and GPS data point density based on historical GPS trajectories cam provide drivers with the timely detour recommendation by predicting the traffic alert.

[24] http://library.duke.edu/data/files/esri/esridm/2010/streetmap_na/streets.html
[25] http://publicdocs.mnr.gov.on.ca/View.asp?Document_ID=17566&Attachment_ID=37853

Road coverage: As described in Section 5.4 the point density of collected GPS data is a crucial factor to provide good coverage over all the roads, including highways and local roads. Building a detailed and fine-grained road network database is possible if sufficient data is in urban area is provided. Furthermore, the connectivity of extracted road centerlines at road junctions could be presented in a better way on the vector road map if an optimal line fitting algorithm is applied in future.

Appendix

Algorithm 1: Representative Point Extraction Algorithm

```
INPUT: Smoothed positioning points (pt), SPT
temPt=Ø; --- stores points found in current cluster, but already stored in another cluster
clsdPt = Ø; --- stores points already be clustered
tempC= Ø; --- stores preliminary centroids of clusters
resultC= Ø; --- stores final representative points
// Step 1: Generating preliminary representative points
FOR pt IN SPT:
    IF pt NOT IN clsdPt
        FOR nearby point (npt) IN 4-m buffer of pt:
            IF npt IN clsdPt
                ADD npt into temPt;
            ELSE IF npt NOT IN clsdPt
                ADD npt into clsdPt;
            END IF
        END FOR
        ADD pt into clsdPt;
        IF pt has neighbours within its 4-m buffer
            CALCULATE centroid of current cluster using Eq.3.2 and Eq. 3.3;
            FOR nearby point (npt) IN 4-m buffer of pt:
                IF npt is closer to centroid of current cluster
                    REMOVE npt from its old cluster in tempC;
                    ADD npt into current cluster in tempC;
                ELSE IF npt is outside of current cluster
                    REMOVE npt from clsdPt and current cluster;
                END IF
            END FOR
            FOR npt in current cluster:
                IF npt's surrounding points (nnpt) NOT IN current cluster but within 4-m buffer of the centroid
                    ADD nnpt into current cluster;
                    IF nnpt NOT IN clsdPt then
                        ADD nnpt into clsdPt;
                    END IF
                END IF
            END FOR
        ELSE IF temPt NOT empty
            FOR tpt IN temPt:
                IF tpt is closer to centroid of current cluster than its old one
                    REMOVE tpt from tempC and add it into current cluster
```

```
                        END IF
                    END FOR
                ELSE
                    centroid = pt
                    ADD all points of current cluster into tempC;
                END IF
            END IF
        END FOR
// Step 2: Split a cluster to two or more clusters if inside point has different moving direction comparing to its centroid
FOR temclsPt IN tempC:
    ddirect  --- calculate cosine value of direction difference between temclsPt and its centroid
    IF ddirect = (0, radian(11˚)]
        ADD temclsPt into Cluster1; --- road non-similar direction to centroid, but maybe on the same road
    ELSE IF ddirect =(radian (11˚),1]
        ADD temclsPt into Cluster2; --- similar direction to centroid, on the same
    ELSE IF ddirect < 0
        ADD temclsPt into Cluster3; --- opposite direction to centroid, on different road
    ELSE
        ADD temclsPt into Cluster4; --- perpendicular direction to centroid, at intersection
    END IF
    CALCULATE new centroid of each cluster and assign it to be representative point;
    ADD representative point into resultC;
END FOR
```

Algorithm 2: Trajectory Reconstruction Algorithm

```
1:  curr : Current point
    Subroutine1: determine initial single optimal preceding or succeeding point;
    Subroutine2: classify point IN candidatePointSet into 2 groups regard to curr:
    ahead[] or behind[];
    Subroutine3A: determine optimal succeeding point from ahead[];
    Subroutine3B: determine optimal preceding point from behind[];
    Subroutine 4: continuously search preceding point;
    Subroutine 5: continuously search succeeding points;
2:  // Main Body
3:  DO
4:      curr = New Rep.point;
5:      candidatePointSet = [ points with 50-m of curr AND sharing common traces ] ;
6:      candidatePointCount = count of candidatePointSet;
7:      IF candidatePointCount = 1
8:          Subroutine1;
9:          IF single optimal preceding point
10:             Subroutine 4;
11:             reformed GPS trajectories;
12:         ELSE
13:             Subroutine 5;
14:             reformed GPS trajectories;
15:         END IF
16:     ELSE
17:         IF candidatePointCount > 1
18:             Subroutine2;
19:             Store classified points into two group: ahead[] and behind[]
20:             IF ahead[ ] IS NOT empty AND behind[ ] IS NOT empty
21:                 Subroutine3A and 3B;
22:                 Subrountine 4 AND Subrountine 5;
23:                 reformed GPS trajectories;
24:             ELSE
25:                 IF ahead[ ] IS NOT empty AND behind[ ] IS empty
26:                     Subroutine3A
27:                     Subroutine 5;
28:                     reformed GPS trajectories;
29:                 ELSE
30:                     IF ahead[ ] IS empty AND behind[ ] IS NOT empty
31:                         Subroutine3B
32:                         Subroutine 4;
33:                         reformed GPS trajectories;
34:                     END IF
35:                 END IF
36:             END IF
37:         END IF
38:     END IF
```

```
39: WHILE ((candidatePointCount != 0 ) OR (candidatePointCount > 1 AND (ahead[]
    IS NOT empty OR behind[] IS empty)))
40:
41: // Subroutine 1: determine single optimal preceding or succeeding point
42: Az1 = azimuths from current to candidate;
43: Az2 = azimuths from candidate to current;
44: Avg = average of current and candidate directions;
45: IF |Az1 - Avg| LESS THAN EQUAL 11°
46:     create connection from curr to succeeding point;
47: ELSE
48:     IF |Az2- Avg| LESS THAN EQUAL 11°
49:         create connection from preceding point to curr;
50:     ELSE
51:         start from next available Rep.point;
52:     END IF
53: END IF
54: IF connection IS valid
55:     start from next available Rep.point;
56: ELSE
57:     mark succeeding/preceding point as connected AND store the connection;
58:     store (curr AND succeeding point) OR (preceding point AND curr)
59: END IF
60:
61: // Subroutine 2: classify point IN candidatePointSet into 2 groups regard to
    curr: ahead[] or behind[]
62: T = different between azimuth (from curr to candidate point) and direct of curr;
63: IF T GREAT THAN 180°
64:     T = T – 360°;
65: END IF
66: IF T LESS THAN -180°
67:     T = T + 360°;
68: END IF
69: IF(T GREAT THAN -90°) AND (T LESS THAN 90°)
70:     candidate point is ahead curr;
71: ELSE
72:     candidate point is behind curr;
73: END IF
74:
75: // Subroutine 3A/3B: determine optimal preceding/succeeding point from
    ahead[]/behind[]
76: minAz1: Az from curr to candidate point;
    minAz2: Az from candidate point to curr;
    point: preceding point/ succeeding point;
    Dist: distance b/n candidate point and curr;
    Avg: average direction of curr and candidate point;
77: minAz1 = 0;
```

```
78:  minAz2 =0;
79:  Dist =0;
80:  Avg=0;
81:  point = NULL;
82:  FOR i IN ahead[](behind[])
83:          IF Az of i !=0 and minAz1(minAz2) ==0
84:                  minAz1(minAz2)= Az of i;
85:                  Dist = distance of i to curr;
86:                  Avg = avg of i and curr;
87:                  point = ID of i;
88:                  Continue;
89:          END IF
90:          IF Az of i !=0 AND minAz1(minAz2) !=0
91:                  IF | Az of i – avg of i and curr| < |minAz1(minAz2) – Avg|
92:                          minAz1(minAz2)= Az of i;
93:                          Dist = distance of i to curr;
94:                          Avg = avg of i and curr;
95:                          point = ID of i;
96:                  ELSE IF | Az of i – avg of i and curr| == |minAz1(minAz2) – Avg|
97:                          IF distance of i to curr < Dist
98:                                  minAz1(minAz2)= Az of i;
99:                                  Dist = distance of i to curr;
100:                                 Avg = avg of i and curr;
101:                                 point = ID of i;
102:                         END IF
103:                 END IF
104:         ELSE IF Az of i !=0 AND minAz1(minAz2) !=0 AND Az of i ==
             minAz1(minAz2)
105:                 IF distance of i to curr < Dist
106:                         minAz1(minAz2)= Az of i;
107:                         Dist = distance of i to curr;
108:                         Avg = avg of i and curr;
109:                         point = ID of i;
110:                 END IF
111:         END IF
112: END FOR
113:
114: // Subroutine 4: continuously search and connect the optimal succeeding
     points.
     // Subroutine 5: continuously search and connect the optimal preceding points.
115: INITIALIZE
116: // Subroutine 4: current & succeeding points; (curr= succeeding; prev = current)
     // Subroutine 5: current & preceding points; (curr=preceding; next= current)
117: SELECT unconnected nearby points in similar direction to curr;
118: candidatePointSet = [ points with 50-m of curr AND sharing common traces with
     curr and prev / next] ;
```

```
119: candidatePointCount = count of candidatePointSet;
120: IF candidatePointCount = 1
121:       Subroutine 2
122:       IF candidatePoint IS succeeding of curr
123:             reformed connection from curr to succeeding point;
124:             prev = curr
125:             curr =succeeding
126:       ELSE IF candidatePoint IS preceding of curr
127:             reformed connection from preceding point to curr;
128:             next=curr
129:             curr=preceding
130:       ELSE
131:             curr = New Rep.point;
132:       END IF
133: ELSE IF candidatePointCount >1
134:       Subroutine3A or 3B
135:       reformed connection from preceding point to curr OR from curr to succeeding;
```

Algorithm 3: Sub-trajectories Clustering Algorithm (Polyline Spatial Clustering)

ReformC: stores reformed GPS trajectories in descending order regarding number of components (polyline segments);
Subroutine6: Sweep line algorithm to partition nearby polyline segments and stored them as candidate divisional line segments for clustering;
Subroutine7: Recursive line clustering algorithm to incrementally cluster divisional line segments on the same road with current polyline segment (lineid);
Subroutine8: Calculate shortest distance between a point and a line segment;
candidateNLS: dictionary store candidate divisional line segment (direct change <= 11°) of a lineid;
actualNLS: dictionary store actual divisional line segments on the same road with a lineid;
tracepassingLS: dictionary store reformed GPS trajectory's ID passing a lineid;
FOR connection **IN** ReformC
    SELECT the first polyline segment FROM all segments of the connection
    // the first polyline segment must have no previous segment along the moving direction of entire connection
    INITIALIZE lineid = first polyline segment.ID
    WHILE lineid != 0
        IF lineid is never be clustered
            STORE IDs of all nearby (50-m)polyline segments around lineid INTO nearFIDlst
            ADD lineid and its coordinates of endpoints, direct, and length INTO clustlst
            MARK lineid as used line segment
            IF nearFIDlst NOT EMPTY // lineid has nearby polylinesegments
                Subroutine6
                Subroutine7
            END IF
        lineid = nextline.ID
        END IF
    MARK all searched polyline segments as clustered
END FOR
// **Subroutine6: Sweep line algorithm**
CALCULATE slope and y-intercept of perpendicular line at starting point of lineid
CALCULATE slope and y- intercept of perpendicular line at ending point of lineid
INPUT: nearby polyline segment (coordinates of endpoints, direct) & lineid (coordinates of endpoints)
CALCULATE slope and y- intercept of nearby polyline segment
CALCULATE intersection points of nearby polyline segment with the perpendicular line at starting point of lineid and the perpendicular line at ending point of lineid
PARTITION nearby polyline segment into sub segments
IF sub segment is between perpendicular lines at starting and ending points of lineid
    STORE sub segment as candidate divisional line segments in candidateNLS for clustering
END IF

```
// Subroutine7: Recursive line clustering algorithm
FUNCTION RecursiveSearch(lineid)
    FOR one nearby polyline segment (nlineid) of lineid
        IF nlineid IN candidateNLS AND nlineid not being clustered
            Subroutine8 // Calculate shortest distances (dist1 & dist2) from
            endpoints of one divisional polyline segment to another divisional
            polyline segment
            IF dist1 <= allowable distance  AND dist2 <= allowable distance
                ADD  nlineid INTO actualNLS
                IF nlineid.TrajectoryID NOT IN tracepassingLS
                    ADD nlineid.TrajectoryID INTO tracepassingLS
                END IF
                RecursiveSearch(nlineid)
            END IF
        END IF
    END FOR
// Subroutine8: Perpendicular distance from one point to a line
INPUT: coordinates of starting point and ending point of a line & coordinates of a
point
CALCULATE vector of a line
CALCULATE length of a line
CALCULATE unit vector of a line
CALCULATE normal unit vector to a line
CALCULATE vector line from starting point of a line to a point
CALCULATE projection of vector line (from starting point of a line to a point) to
normal unit vector to a line (shortest distance)
```

Algorithm 4: Road Centerline Extraction Algorithm

```
INPUT: tracepassingLS;
       actualNLS;
FOR connection IN ReformC
    SELECT the first polyline segment FROM all segments of the connection
    INITIALIZE lineid = first polyline segment.ID
    WHILE lineid != 0
        IF lineid is never be clustered
            INITIALIZE prevline
            INITIALIZE curntline
            INITIALIZE nextline
            IF lineid has no previous polyline segment
                prevline = curntline = lineid
                nextline = lineid.NEXTLineID
            ELSE IF lineid has previous and next polyline segments
                prevline = lineid.PREVIOUSLineID
                curntline = lineid
                nextline = lineid.NEXTLineID
            ELSE // lineid has no next polyline segments
                prevline = lineid.PREVIOUSLineID
                nextline = curntline= lineid
            END IF
            FIND common reformed GPS trajectories passing through all
            three sequential polyline segments FROM tracepassingLS
            SELECT clustered nearby divisional polyline segments FROM
            actualNLS[curntline] WHERE actualNLS[curntline] has more
            than three components
            FOR each selected polyline segment
                APPLY Subroutine6 to get intersection starting and ending
                points
                ADD them into starting group and ending group,
                respectively
                CALCULATE the length of divisional polyline segment
                IF selected polyline segment.TraceID in common reformed
                GPS trajectories AND selected polyline segment never be
                clustered
                    ADD its new coordinates, direct, and length INTO
                    final cluster for calculating centerline
                ELSE
                    IF selected polyline segment never be clustered
                        ADD its new coordinates, direct, and length
                        INTO temporary cluster for refining centerline
                    END IF
                END IF
            END FOR
            APPLY Eq. 3.6 to calculate direction and coordinates of starting,
```

middle, and ending points of the road centerline segment from the final cluster

Subroutine9

IF temporary cluster NOT EMPTY

APPLY Eq. 3.11 plumb-line algorithm to detect if divisional polyline segments in the temporary cluster is inside of the polygon

IF inside of polygon

REFINE the coordinates and direction of extracted centerline segment

END IF

END IF

lineid = nextline.ID

Subroutine9 // constructing a polygon around the extracted centerline segment to cover the road width

Identify points in starting group as left-side or right-side of lineid and points in ending group as left-side or right-side of lineid;

IN each group // left-side of starting; right-side of starting; left-side of ending; right-side of ending

SELECT one point which has furthest distance to lineid as a vertex of the polygon coving the road width.

CONSTRUCT the polygon in direction of clockwise (left-side of starting, left-side of ending, right-side of ending, right-side of starting)

Algorithm5: Road Centerline Topological Connection Algorithm

priorityoptIndex: stores Road ID by descending order of # of components;
MptNbyLnDict**:** stores relationship between an endpoint and its 15-m nearby polyline segments;
LNDict: stores attributes of a polyline segment: connectionID and direction;
CLNbyLnDict: stores relationship between an endpoint and its 150-m nearby endpoints;
RoadDict: stores a road (RoadID) and endpoints of its polyline segments;

1. // Constructing topological connectivity from the ending point of a road to the starting point of another road
2. **INITIALIZE** tempSplit: a dictionary temporarily stores next nodes of an endpoint of an extracted road centerline, where,
   tempSplit[key]=[(value pair)]; key =RoadID of nextnode, (value pair) = (nextnode, endpoint)
3. **FOR** road.ID **IN** priorityoptIndex:
4. SELECT last two ending points (endid1 & endid2) on the road;
5. CALCULATE Azimuth from endid2 to endid1;
6. SELECT all endpoints within 150-m of endid1 FROM CLNbyLnDict WHERE sharing common connections AND in similar direction to endid1
7. **FOR** each endpoint **IN** Selected endpoints:
8. **IF** Abs(cos (endpoint.Direct – endid1.Direct)) = [0.8, 1] // Y-Split road
9. CLASSIFY the endpoint into: left-sideofendid1[] or right-sideofendid1[ ]
10. SELECT the left-side endpoint has min. Azimuth from endid1 FROM left-sideofendid1[ ]
11. SELECT the right-side endpoint has min. Azimuth from endid1 FROM right-sideofendid1[ ]
12. **ELSE IF** Abs(cos (endpoint.Direct – endid1.Direct)) =[0, 0.8) // Intersection road
13. CLASSIFY the endpoint into: left-sideofendid1[] or right-sideofendid1[]
14. SELECT the left-side endpoint has shortest distance from endid1 FROM left-sideofendid1[ ]
15. SELECT the right-side endpoint has shortest distance from endid1 FROM right-sideofendid1[ ]
16. **END IF**
17. **END FOR**
18. INITIALIZE nextnode1 to zero
19. INITIALIZE nextnode2 to zero
20. **IF** the left-side endpoint **AND** the right-side endpoint **EMPTY**
21. nextnode1=0
22. nextnode2=0
23. **ELSE IF** the left-side endpoint **EMPTY**
24. nextnode1=0
25. nextnode2= the right-side endpoint
26. **ELSE IF** the right-side endpoint **EMPTY**

```
            nextnode1= the left-side endpoint
            nextnode2= 0
        ELSE
            IF the left-side endpoint.RoadID == the right-side endpoint.RoadID
                IF the left-side endpoint.ID > the right-side endpoint.ID
                    nextnode1= the left-side endpoint
                ELSE // the left-side endpoint.ID < the right-side endpoint.ID
                    nextnode2= the right-side endpoint
                END IF
            ELSE
                nextnode1= the left-side endpoint
                nextnode2= the right-side endpoint
            END IF
        END IF
        IF nextnode1 <> 0 AND nextnode2<>0
            ADD nextnode1 and nextnode2 INTO corresponding tempSplit;
            // E.g. tempSplit[RoadID of nextnode1]=[(nextnode1_1, endid1),
            (nextnode1_2, endid1),..( nextnode1_n, endid1)]
            tempSplit[RoadID of nextnode2]=[(nextnode2_1, endid1), (nextnode2_2,
            endid1),..( nextnode2_n, endid1)]
        ELSE IF nextnode1 <> 0
            ADD nextnode1 INTO tempSplit;
        ELSE IF nextnode2<>0
            ADD nextnode2 INTO tempSplit;
        ELSE
            CONTINUE
        END IF
    END FOR
    // Comparison of nextnodes stored in tempSplit, in order to select the optimal
    nextnode
    INITIALIZE newSplit: a dictionary stores refined nextnodes in value pairs
    FOR each road of nextnode IN tempSplit
        IF tempSplit has two value pairs
            IF two nextnodes close to each other
                SELECT nextnode further from endid1 as the optimal nextnode as
                the optimal nextnode;
            END IF
            ADD optimal nextnode INTO each value pair in newSplit;
        ELSE IF tempSplit has more than two value pairs
            IF all nextnodes close to the starting point of a extracted road centerline
                SELECT nextnode further from endid1 as the optimal nextnode as
                the optimal nextnode;
                ADD optimal nextnode INTO each value pair in newSplit;
            ELSE {all nextnodes locate at some polyline segments of a extracted
            road centerline
                ADD all  nextnodes INTO corresponding value pairs in newSplit;
```

66. // E.g. newSplit[RoadID of nextnodes]= [(nextnode1_1, endid1_1), (nextnode1_2, endid1_2),..(nextnode1_n, endid1_m)]
67. **ELSE** // tempSplit only has one value pair
68. ADD this unique value pair INTO newSplit;
69. **END FOR**
70. // Updating the starting point of each extracted road centerline, since some of them have new starting points (nextnode related to an endpoint of another extracted road centerline
71. **FOR** each road (RoadID)**IN** RoadDict
72. **IF** newSplit has RoadID
73. **IF** newSplit[RoadID] only has one value pair **AND** nextnode of the value pair is close to the original starting point of the current road
74. Update starting point IN RoadDict;
75. **ELSE IF** newSplit[RoadID] has more than one value pair
76. SELECT minimum nextnode.ID that is close to the original starting point of the current road;
77. Update starting point IN RoadDict;
78. **ELSE**
79. REMAIN original starting point IN RoadDict;
80. **END IF**
81. **ELSE**
82. REMAIN original starting point IN RoadDict;
83. **END IF**
84. **END FOR**
85. // Refining the topological connectivity amongst the starting point of one road and the ending point of another road
86. **FOR** road.ID **IN** priorityoptIndex
87. **IF** road has more than one value pair **IN** RoadDict
88. SELECT first two starting points (startid1 & startid2) on the road;
89. CALCULATE Azimuth from startid1 to startid2;
90. SELECT all endpoints within 150-m of startid1 FROM CLNbyLnDict WHERE sharing common connections AND in similar direction to startid1
91. **FOR** each endpoints **IN** Selected endpoints
92. CLASSIFY the endpoint into: left-side of startid1 or right-side of startid1
93. SELECT the left-side endpoint has min. Azimuth to startid1 **FROM** left-sideofendid1[ ]
94. SELECT the right-side endpoint has min. Azimuth to startid1 FROM right-sideofendid1[ ]
95. **END FOR**
96. INITIALIZE previousnode1 to zero
97. INITIALIZE previousnode2 to zero
98. **IF** the left-side endpoint **AND** the right-side endpoint **EMPTY**
99. previousnode1 =0
100. previousnode2 =0

```
        ELSE IF the left-side endpoint EMPTY
            previousnode1 =0
            previousnode2 = the right-side endpoint
        ELSE IF the right-side endpoint EMPTY
            previousnode1 = the left-side endpoint
            previousnode2 = 0
        ELSE
            IF the left-side endpoint.RoadID == the right-side endpoint.RoadID
                IF the left-side endpoint.ID > the right-side endpoint.ID
                    previousnode1= the left-side endpoint
                ELSE // the left-side endpoint.ID < the right-side endpoint.ID
                    previousnode2 = the right-side endpoint
                END IF
            ELSE
                previousnode1= the left-side endpoint
                previousnode2 =the right-side endpoint
            END IF
        END IF
        IF road. ID  NOT IN newSplit
            IF previousnode1<>0 AND previousnode1<>0
                ADD both INTO newSplit[road.ID ] AS new value pairs // [(startid1,
                previousnode2),(startid1, previousenode1)]
            ELSE IF previousnode2== 0
                ADD previousnode1 INTO newSplit[road.ID ]
            ELSE IF previousnode1== 0
                ADD previousnode2 INTO newSplit[road.ID]
            END IF
        END IF
    END FOR
    WRITE RoadDict AND newSplit INTO Feature Class
```

References

Achtert, E., Bohm, C., Kriegel, H., Kroger, P., & Zimek, A. (2006). Deriving quantitative models for correlation clusters. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 4-13.

Ahmed, M., & Wenk, C. (2012). Constructing street networks from GPS trajectories. *Algorithms--ESA 2012, ,* 60-71.

Baumgartner, A., Steger, C., Wiedemann, C., Mayer, H., Eckstein, W., & Ebner, H. (1996). Update of roads in gis from aerial imagery: Verification and multi-resolution extraction. *International Archives of Photogrammetry and Remote Sensing, 31,* 53-58.

Biagioni, J., & Eriksson, J. (2012a). Inferring road maps from GPS traces: Survey and comparative evaluation. *Transportation Research Board 91st Annual Meeting,* (12-3438)

Biagioni, J., & Eriksson, J. (2012b). Map inference in the face of noise and disparity. *Proceedings of the 20th International Conference on Advances in Geographic Information Systems,* Redondo Beach, California. 79-88. doi: 10.1145/2424321.2424333

Biagioni, J., Gerlich, T., Merrifield, T., & Eriksson, J. (2011). Easytracker: Automatic transit tracking, mapping, and arrival time prediction using smartphones. *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems,* 68-81.

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

Cao, L., & Krumm, J. (2009). From GPS traces to a routable road map. *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems,* 3-12.

Charles D. Ghilani, Paul R.Wolf (Ed.). (2002). *Elementary surveying: An introduction to geomatics* (thirteenth ed.). Upper Saddle River, New Jersey: Pearson Education, Inc.

Chen, C., & Cheng, Y. (2008). Roads digital map generation with multi-track gps data. *Education Technology and Training, 2008. and 2008 International Workshop on Geoscience and Remote Sensing. ETT and GRS 2008. International Workshop On, , 1* 508-511.

Chen, J., Leung, M. K., & Gao, Y. (2003). Noisy logo recognition using line segment hausdorff distance. *Pattern Recognition, 36*(4), 943-955.

Chen, T. Q., & Lu, Y. (2002). Color image segmentation—an innovative approach. *Pattern Recognition, 35*(2), 395-405.

Chen, Y., & Krumm, J. (2010). Probabilistic modeling of traffic lanes from GPS traces. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems,* 81-88.

Dal Poz, A. P., Zanin, R. B., & Do Vale, G. M. (2006). Automated extraction of road network from medium-and high-resolution images. *Pattern Recognition and Image Analysis, 16*(2), 239-248.

Dal Poz, A., & Do Vale, G. (2003). Dynamic programming approach for semi-automated road extraction from medium-and high-resolution images. *ISPrS Archives, 34*(Part 3), W8.

Davies, J. J. (2009). *Programming networks of vehicles.* (Technical Report No. UCAM-CL-TR-761). Cambridge, United Kingdom: University of Cambridge, Computer Laboratory.

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

Easa, S. M. (2002). Geometric design. *Civil engineering handbook* (2nd ed.,) CRC Press.

Edelkamp, S., & Schrodl, S. (2003). Route planning and map inference with global positioning traces. *Computer Science in Perspective,* , 128-151.

Ekpenyong, F., Palmer-Brown, D., & Brimicombe, A. (2009). Extracting road information from recorded GPS data using snap-drift neural network. *Neurocomputing, 73*(1), 24-36.

Fortier, A., Ziou, D., Armenakis, C., & Wang, S. (1999). Survey of work on road extraction in aerial and satellite images. *Center for Topographic Information Geomatics, Ontario, Canada.Technical Report, 241*

Gao, J., & Wu, L. (2004). Automatic extraction of road networks in urban areas from IKONOS imagery based on spatial reasoning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science, 35*

Gonzalez, R. C., & Woods, R. E. (Eds.). (2008). *Digital image processing* (3rd ed.) Prentice Hall.

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

Grossman, Robert and Sabala, Michal and Aanand, Anushka and Eick, Steve and Wilkinson, Leland and Zhang, Pei and Chaves, John and Vejcik, Steve and Dillenburg, John and Nelson, Peter and others. (2005). Real time change detection and alerts from highway traffic data. *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, 69-69.

Gruen, A., & Li, H. (1997). Semi-automatic linear feature extraction by dynamic programming and LSB-snakes. *Photogrammetric Engineering and Remote Sensing, 63*(8), 985-994.

Guo, R., Li, D., Kartalopoulos, S., Buikis, A., Mastorakis, N., & Vladareanu, L. (2008). Road detection method for land consolidation using mathematical morphology from high resolution image. *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, (13)

Guo, D., Liu, S., & Jin, H. (2010). A graph-based approach to vehicle trajectory analysis. *Journal of Location Based Services, 4*(3-4), 183-199.

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

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

Herman, E. A. (2002). Benefits of a bi-directional route system. *Proceedings of the Annual ESRI International User Conference, California, FL*, 1-12.

Herrera, J. C., Work, D. B., Herring, R., Ban, X. J., Jacobson, Q., & Bayen, A. M. (2010). Evaluation of traffic data obtained via GPS-enabled mobile phones: The mobile century field experiment. *Transportation Research Part C: Emerging Technologies, 18*(4), 568-583.

Hu, X., & Tao, C. (2005). A reliable and fast ribbon road detector using profile analysis and model-based verification. *International Journal of Remote Sensing, 26*(5), 887-902.

Jang, S., Kim, T., & Lee, E. (2010). Map generation system with lightweight GPS trace data. *Advanced Communication Technology (ICACT), 2010 the 12th International Conference On, , 2* 1489-1493.

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

Kharrat, A., Popa, I. S., Zeitouni, K., & Faiz, S. (2008). Clustering algorithm for network constraint trajectories. *Headway in Spatial Data Handling,* , 631-647.

Lee, J., Han, J., & Whang, K. (2007). Trajectory clustering: A partition-and-group framework. *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, 593-604.

Li, J., Qin, Q., Xie, C., & Zhao, Y. (2012). Integrated use of spatial and semantic relationships for extracting road networks from floating car data. *International Journal of Applied Earth Observation and Geoinformation, 19*, 238-247.

Li, Z., Lee, J., Li, X., & Han, J. (2010). Incremental clustering for trajectories. *Database Systems for Advanced Applications,* 32-46.

Limaa, F., & Ferreiraa, M. (2009). Mining spatial data from gps traces for automatic Road Network extraction. *In: 6th International Symposium on Mobile Mapping Technology*, Presidente Prudente, Sao Paulo, Brazil.

Lin, X., Zhang, J., Liu, Z., & Shen, J. (2008). Semi-automatic extraction of ribbon roads from high resolution remotely sensed imagery by T-shaped template matching. *Proceedings of SPIE, the International Society for Optical Engineering,* 71470J. 1-71470J. 8.

Lin, X., Zhang, J., Liu, Z., & Shen, J. (2009). Semi-automatic road tracking by template matching and distance transform. *Urban Remote Sensing Event, 2009 Joint*, 1-7.

Liu, X., Zhu, Y., Wang, Y., Forman, G., Ni, L. M., Fang, Y., & Li, M. (2012). Road recognition using coarse-grained vehicular traces.

Lumelsky, V. J. (1985). On fast computation of distance between line segments. *Information Processing Letters, 21*(2), 55-61.

Matheron, G., & Serra, J. (2002). The birth of mathematical morphology. *Proc. 6th Intl. Symp. Mathematical Morphology*, 1-16.

Mayer, H., Laptev, I., Baumgartner, A., & Steger, C. (1997). Automatic road extraction based on multi-scale modeling, context, and snakes. *International Archives of Photogrammetry and Remote Sensing, 32*(Part 3), 106-113.

Mena, J. (2003). State of the art on automatic road extraction for GIS update: A novel classification. *Pattern Recognition Letters, 24*(16), 3037-3058.

Mena, J., & Malpica, J. (2005). An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery. *Pattern Recognition Letters, 26*(9), 1201-1220.

Mena, J. (2006). Automatic vectorization of segmented road networks by geometrical and topological analysis of high resolution binary images. *Knowledge-Based Systems, 19*(8), 704-718.

Menard, T., & Miller, J. (2011). Comparing the GPS capabilities of the iPhone 4 and iPhone 3G for vehicle tracking using FreeSim_Mobile. *Intelligent Vehicles Symposium (IV), 2011 IEEE,* 278-283.

Mohammadzadeh, A., Tavakoli, A., & Zoej, M. V. (2004). Automatic linear feature extraction of iranian roads from high resolution multi-spectral satellite imagery. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 35*, 764.

Mohammadzadeh, A., Tavakoli, A., & Valadan Zoej, M. J. (2006). Road extraction based on fuzzy logic and mathematical morphology from pan-sharpened ikonos images. *The Photogrammetric Record, 21*(113), 44-60.

Niehoefer, B., Burda, R., Wietfeld, C., Bauer, F., & Lueert, O. (2009). Gps community map generation for enhanced routing methods based on trace-collection by mobile phones. *Advances in Satellite and Space Communications, 2009. SPACOMM 2009. First International Conference On,* 156-161.

Park, S., & Kim, T. (2001). Semi-automatic road extraction algorithm from IKONOS images using template matching. *Paper Presented at the 22nd Asian Conference on Remote Sensing, , 5* 9.

Peteri, R., & Ranchin, T. (2002). Extraction and update of street networks in urban areas from high resolution satellite images. *In YLC Armenakis (Ed.), ISPRS Commission IV Symposium" Spatial Data Handling", Volume Vol. XXIV,*

Peteri, R., & Ranchin, T. (2003). Multiresolution snakes for urban road extraction from ikonos and quickbird images. *23nd EARSeL Annual Symposium" Remote Sensing in Transition*, 141-147.

Quackenbush, L. J. (2004). A review of techniques for extracting linear features from imagery. *Photogrammetric Engineering & Remote Sensing, 70*(12), 1383-1392.

Rajeswari, M., Gurumurthy, K., Omkar, S., Senthilnath, J., & Reddy, L. P. (2011). Automatic road extraction using high resolution satellite images based on level set and mean shift methods. *Electronics Computer Technology (ICECT), 2011 3rd International Conference On, , 2* 424-428.

Schroedl, S., Wagstaff, K., Rogers, S., Langley, P., & Wilson, C. (2004). Mining GPS traces for map refinement. *Data Mining and Knowledge Discovery, 9*(1), 59-87.

Shi, W., Shen, S., & Liu, Y. (2009a). Automatic generation of road network map from massive gps, vehicle trajectories. *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference On,* 1-6.

Shi, W., Shen, S., & Liu, Y. (2009b). A method for updating road map via GPS. *Proceedings of The 2009 International Workshop on Information Security and Application, , 16* 21-22.

Shaker, A., Yan, W.Y., & Easa, S.M., (2010). Using stereo satellite imagery for topographic and transportation applications: an accuracy assessment. *GIScience and Remote Sensing,* 47(3), 321-337.

Shaker, A., Yan, W.Y., & Easa, S.M., (2011). Construction of digital 3D highway model using stereo IKONOS satellite imagery. *GeoCarto International,* 26(1), 49-67.

Shukla, V., Chandrakanth, R., & Ramachandran, R. (2002). Semi-automatic road extraction algorithm for high resolution images using path following approach. *Indian Conference on Computer Vision, Graphics and Image Processing, , 6* 201-207.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. (Vol. 26). CRC press.

Steger, Carsten and Mayer, Helmut and Radig, Bernd and others. (1997). The role of grouping for road extraction. *Automatic Extraction of Man-made Objects from Aerial and Space Images (II), 245,* 256.

Steiner, A., & Leonhardt, A. (2011). A map generation algorithm using low frequency vehicle position data. *Transportation Research Board, 90th Annual.*

Transportation Association of Canada (TAC). (1999). Geometric Design Guide for Canadian Roads, *TAC,* Ottawa, Ontario.

Tavares, J. M. R. S., & Padilha, A. J. M. N. (1995). A new approach for merging edge line segments. *Proceedings of RecPad'95 - 7th Portuguese Conference on Pattern Recognition,* Aveiro, Portugal.

Valero, S., Chanussot, J., Benediktsson, J. A., Talbot, H., & Waske, B. (2010). Advanced directional mathematical morphology for the detection of the road network in very high resolution remote sensing images. *Pattern Recognition Letters, 31*(10), 1120-1127.

Wang, J., Rui, X., Song, X., Wang, C., Tang, L., Li, C., & Raghvan, V. (2011). A weighted clustering algorithm for clarifying vehicle GPS traces. *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International,* 2949-2952.

Wang, X., & Hamilton, H. (2010). Using clustering methods in geospatial information systems. *Geomatica, 64*(3), 347-361.

Wiehe, S. E., Carroll, A. E., Liu, G. C., Haberkorn, K. L., Hoch, S. C., Wilson, J. S., & Fortenberry, J. D. (2008). Using GPS-enabled cell phones to track the travel patterns of adolescents. *International Journal of Health Geographics, 7*(1), 22.

Willrich, F. (2002). Quality control and updating of road data by GIS-driven road extraction from imagery. *INTERNATIONAL ARCHIVES OF PHOTOGRAMMETRY REMOTE SENSING AND SPATIAL INFORMATION SCIENCES, 34*(4), 761-767.

Worrall, S., & Nebot, E. (2007). Automated process for generating digitised maps through gps data compression. *Australasian Conference on Robotics and Automation,*

Yoon, J., Noble, B., & Liu, M. (2007). Surface street traffic estimation. *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services,* 220-232.

Young, S. (2007). Real-time traffic operations data using vehicle probe technology. *Proceedings of the 2007 Mid-Continent Transportation Research Symposium,* 16-17.

Yun, L., & Uchimura, K. (2007). Using self-organizing map for road network extraction from ikonos imagery. *International Journal of Innovative Computing, Information and Control, 3*(3), 641-656.

Zandbergen, P. A. (2009). Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning. *Transactions in GIS, 13*(s1), 5-25.

Zandbergen, P. A., & Barbeau, S. J. (2011). Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones. *Journal of Navigation, 64*(03), 381-399.

Zhang, L., Thiemann, F., & Sester, M. (2010). Integration of GPS traces with road map. *Proceedings of the Second International Workshop on Computational Transportation Science,* 17-22.

Zhang, Q., & Couloigner, I. (2004). A wavelet approach to road extraction from high spatial resolution remotely-sensed imagery. *Geomatica, 58*(1), 33-39.

Zhao, H., Kumagai, J., Nakagawa, M., & Shibasaki, R. (2002). Semi-automatic road extraction from high-resolution satellite image. *INTERNATIONAL ARCHIVES OF PHOTOGRAMMETRY REMOTE SENSING AND SPATIAL INFORMATION SCIENCES, 34*(3/A), 406-411.

Zhao, Y., Liu, J., Chen, R., Li, J., Xie, C., Niu, W., . . . Qin, Q. (2011). A new method of road network updating based on floating car data. *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International,* 1878-1881.