OnlineAirTrajClus: An Online Aircraft Trajectory Clustering for Tarmac Situation Awareness

Wei Shao
School of Science
RMIT University
Melbourne, Australia
wei.shao@rmit.edu.au

Flora D. Salim
School of Science
RMIT University
Melbourne, Australia
flora.salim@rmit.edu.au

Jeffrey Chan
School of Science
RMIT University
Melbourne, Australia
jeffrey.chan@rmit.edu.au

Kai Qin
School of Science
RMIT University
Melbourne, Australia
kai.qin2@rmit.edu.au

Jiaman Ma
School of Science
RMIT University
Melbourne, Australia
jiaman.ma@rmit.edu.au

Bradley Feest
Aerospace Systems
Northrop Grumman Corporation
Redondo Beach, USA
Bradley.Feest@ngc.com

***Abstract*—On-ground aircraft trajectory information plays a key role in airport situations awareness prediction and management. Airport administration needs to arrange and schedule the time and order of aircraft landing and take-off events based on a precise and real-time information of on-ground aircraft. Recently, a large dataset of GPS-derived aircraft at airports, available from the Federal Aviation Administration (FAA), provides researchers with an opportunity to monitoring on-ground aircraft trajectory. In this paper, we present a framework to incrementally cluster airport aircraft trajectories based on the GPS data. The framework consists of two steps: 1) Classifying airport aircraft data according to spatial and temporal information. 2) Merging the similar aircraft trajectories incrementally. We evaluate our framework experimentally using a state-of-the-art test-bed technique, and find that it can effectively and efficiently construct and update on-ground aircraft trajectory map.**

***Index Terms*—On-ground aircraft, Spatio-temporal data, Trajectory clustering, situation awareness**

I. Introduction

Trajectory data are widely collected from various areas such as animal migration, transportation, travelling and air traffic. In air traffic domain, increasing amount of air traffic data attracted substantial attention of general public and researchers because it provides significant economical benefits for airline industry. Federal Aviation Administrations (FAA) recorded GPS trajectory information for each aircraft at United States airports, which offers an opportunity to study the situational awareness map of on-ground aircraft at the airport [1].

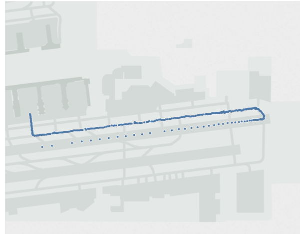
In the airport traffic and control system, situational awareness is of paramount importance. Situational awareness map at the airport indicates a location based map where the value of each element describe the property of the specific coordinate [2]. The map is constructed from radars and GPS system. From constructing such map, we can monitor ground transport and evaluate the conditions of on-ground transport infrastructure since situational awareness map provides a reliable and accurate analytic tool for airport traffic controllers by offering them features and indicators they require for on-ground air traffic management and scheduling [3].

Situational awareness map can also be used to describe the pattern of aircraft trajectory and airport complexity with the goal of increased utilisation of airport capacities. The significant boosting number of flights in the limited airport space and time schedule multiplies the complexity of planning, management, and monitoring missions, which raise the urgent needs of relevant data analytic. In recent years, many research work focus on air traffic flow monitoring, controlling and scheduling [4]. However, very few researchers pay attention to the on-ground moving vehicles and aircraft on the tarmac (or officially called *apron* by FAA). It can help airport traffic managers to recognise patterns of aircraft movement in order to reduce traffic congestion [5] [6], and predict the anomaly of air traffic [7]. Situational awareness map is widely used in aerospace management to monitoring and characterise the air traffic flow [8] [9].

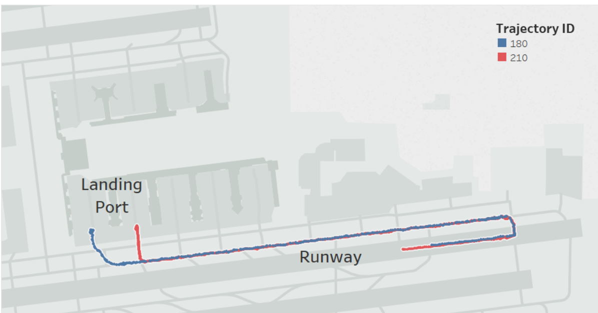
Also, situational awareness map construction is a required preprocessing step for many existing problems for traffic control at airport such as traffic congestion and anomaly detection. Our observations show that most aircraft departure and landing trajectories are not the shortest or optimised. By constructing the situational awareness map, we can extract abnormal trajectories. For example, Figure 1a shows a trajectory for an aircraft taxiing in parking area. It shows that this aircraft takes a long time taxiing in parking area. Using situational awareness map, we can detect such case and optimise the taxiing schedule. Secondly, in the parking area, there is no any fixed or regular tarmac for aircraft. An aircraft can go anywhere before landing and taking off, not

This research was supported by Northrop Grumman Corporations USA through the funding for the project "Spatio-temporal Analytics for Situation Awareness in Operations". Wei Shao is a PhD candidate supported by RMIT Sustainable Urban Precinct Project (SUPP) research grant "iCo2mmunity: Personal and Community Monitoring for University-wide Engagement towards Greener, Healthier, and more Productive Living".

(a) Aircraft trajectory in the parking space



(b) An regular aircraft trajectory with different speed



(c) Two aircraft trajectories on the same runway

Fig. 1. An example of aircraft trajectory on the road and off the road.

just taxiing on the tarmac or runway. Figure 1c depicts two landing aircraft trajectories with similar paths. In this scenario, the routes of these aircraft are all on the runway or tarmac. Therefore, constructing a aircraft situational awareness map is more complicated than building a regular vehicle trajectory map such as taxi.

In comparison to traditional methods, inferring an on-ground aircraft trajectory map using aircraft GPS trajectories is more challenging. Firstly, airport runways are different from other sources of GPS data such as taxi or cars. The trajectories of on-ground aircraft at the airport are more flexible and uncertain because the common tarmac are much narrower than airport runways. Second, the speeds and headings of aircraft are more uncertain than other vehicles, because all aircraft at airports need to follow the protocols and directions from air traffic management systems or the traffic controllers. Also, the span of aircraft is much larger than other vehicles. Figure 1b illustrates a regular aircraft take-off trajectory. The GPS locations are recorded at the same time interval, but the spatial distances between each GPS point are significantly different, because the speed of aircraft changes rapidly in the last stage of take-off. This leads to difficulty to estimate routes when speed changes rapidly across a few GPS locations. Thirdly, in comparison to common road network, airport aircraft route networks change more rapidly. It always changes based on the environmental factors such as wind or climate. Therefore, batch processed-based methods are not suitable for aircraft route network construction. An incremental learning approach is necessary to be used to update the route network with new aircraft trails. In summary, there is a large disparity in the requirements for inferring situational awareness map of on-ground aircraft from GPS data.

In this paper, we propose an on-ground aircraft trajectory clustering framework which is incremental, low computational cost, analytic and data-driven. This framework can create an on-ground aircraft trajectory situational awareness map. Each take-off and landing trajectory can have a corresponding graph as the underlying representation of the route networks in the map. The framework can create the map from scratch, and update the map step-by-step with new on-ground aircraft trajectories. Additionally, our framework does not need any training process or much heuristic knowledge. Also, we introduce a pre-classification and de-noising method to represent the airport aircraft trajectory data, which is able to mine the useful information from a massive amount of trajectories data. We also propose a simple but effective approach to merge the similar airport aircraft trajectories. This intermediate merging method can find the central lines of trajectories incrementally. We also employ visualisation method and some state-of-art test-bed to evaluate our framework. Our contributions in this paper come in three parts:

- We analyse the characteristics of aircraft trajectory data and propose a classification approach to extract useful route information from massive amounts of GPS trajectory data.
- We define the on-ground aircraft situational awareness map construction problem and propose an incremental clustering approach for updating the map with new aircraft trajectories.
- We conduct an extensive experiment on a large real-world dataset to demonstrate the effectiveness and efficiency of our approach in three aspects.

II. Related Work

Inferring a situational awareness map using clustering methods has become a popular research topic in recent years. GPS trajectory has been utilised in smart mobility applications [10]–[13]. In this section, we briefly review the related work which can be separated into two categories: points clustering and trajectory clustering.

Points Clustering: In the early years, most researches focus on points clustering. Edelkamp er.al. [14], in 2003, used GPS points to refine a map by a sequence of steps, including map segmentation, road segment clustering, noisy data removal, road centreline generation and lane finding. This provided heuristic information to subsequent studies. Cao and Krumm [15] tried to cluster the GPS traces by simulating physical attraction between them. In 2010, Agamennoni et al. [16] and Fathi et al. [17] proposed approaches which can infer the road maps from GPS data. It samples the nodes along the centreline and then incrementally linked them together to yielding the final graph. It also uses a dominant set framework to cluster the points. Qiu and Wang [18] propose a road map inference framework based on segmentation and grouping in 2016. The authors use DBSCAN with orientation constrain to divide the GPS data into clusters. The main contributions of this work is that they develop a new cluster algorithm to generate the point clusters and propose a Hidden Markov Model-based map matching algorithm to build the topological relationship of the centrelines.

Trajectories Clustering: As opposed to GPS point clustering, which focuses on neighbouring points, trajectory and sub-trajectory clustering methods are based on route similarities. Some studies, such as [19] [20], focus on using GPS trajectories to construct street networks and discover popular routes. There are some other studies, such as [21], [22] and [23] mine the patterns of trajectories to understand different movement patterns. Wei et al. [24] proposed a framework called RICK to construct the popular routes from uncertain trajectories. RICK is able to construct the top-k routes which pass through the a specific location within a certain time period. It first constructs the regions on a graph and uses probability estimation to infer the edges, then they infers the most popular routes by a score function. Uduwaragoda et al. [25] extract the lanes and its boundary from GPS data using a kernel density estimation based method. But their method needs an existing road map with the road centreline. Karagirgou et al. [26] proposed a new layered map construction algorithm in 2017. The aim of the approach is to generate one road network layer and fuse it into one single network. The steps of this approach including segmenting the trajectory data based on the corresponding types of movement and constructing the topology of the road network hierarchically. The above researches are focus on the trajectories with certain route network. The trajectories with uncertainty patterns are more complex. Kuijpers et al. [27] analyse trajectories using uncertainty information. Pfoser et al. [28], Nanni et al. [29] and Giannotti et al. [30] are studying the trajectories of moving objects with uncertain patterns. Another interesting work [31] takes advantage of the social media to infer the road map. Authors use a data mining technique and natural language processing tool to extract the spatial information and a map of the road network.

Above works take advantages of clustering method and evaluate the result with many different criteria such as topology, path completeness and other clustering evaluation metrics [32]. Our work combine both point-based clustering and trajectory clustering solutions. It firstly clustering on-ground aircraft trajectory incrementally.

III. Overview

In this section, we model and define the on-ground aircraft trajectory map construction and updating problem, and outline the solution framework.

A. Trajectory Definition

A trajectory $\mathbf{t_i} \in \mathbf{T}$ is a vector of GPS point x_i, where x_i is the (GPS) point position of aircraft or vehicle at time τ_i. The measurement of x_i has seven features shown in Table I. Latitude and longitude information reveals the location of the GPS point. The timestamp is the time that the GPS point is recorded. ID is a global identity of GPS data point. Trajectory ID denotes the aircraft or vehicle trajectory that the GPS point x_i belongs to. The speed and heading of x_i can be estimated by the position of GPS point x_{i-1} and x_i. See Table I for summary of the symbols used.

TABLE I
ATTRIBUTE LIST OF GPS POINT x_i

Feature	Symbol	Description
Latitude	lat_i	Latitude
Longitude	lon_i	Longitude
Timestamp	τ_i	UTC time when recording the GPS point x_i
ID	id_{x_i}	The unique identity of GPS point x_i
Trajectory ID	$\mathbf{t}(x_i)$	identity of trajectory where $x_i \in t_i d$
Speed	$speed_i$	The velocity of vehicle or aircraft
Heading	$\mathbf{h_i}$	The clock-wise angle between the moving direction and the earth true north

B. Problem Definition

We define the on-ground aircraft trajectory map construction problem as below: Given n trajectories $\mathbf{T} = \{\mathbf{t_1}, \mathbf{t_2}, ...\mathbf{t_n}\}$, we aim to construct a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{G}$ is a directed graph and $\mathbf{V}$ denotes the geographical location in the graph. The edge $e_{ij} \in \mathbf{E}$ represents the possible direct paths between $\mathbf{v_i}$ and $\mathbf{v_j}$, where $\mathbf{v_i}, \mathbf{v_j} \in \mathbf{V}, i \neq j$.

C. System Framework

Figure 2 depicts an overview of our framework, which consists of two main components: on-ground aircraft trajectory data preprocessing and airport aircraft route network generation.

1) Preprocessing: This component takes the aircraft trajectories and performs two main tasks: 1) Trajectory data cleaning, which aims to remove noise and, false data, and interpolate the trajectories; 2) Trajectory data classification, which classifies the aircraft data into four categories: parking, patrolling, landing and take-off (details in Section IV).

2) Situational Awareness Map Generation: This component mainly generates a situational awareness map of the on-ground aircraft trajectory based on pre-processed GPS trajectory points. It incrementally updates the map by adding new trajectories to the trajectory pool one-by-one. For each new trajectory, the part which shares the same tarmac or runway with the existing trajectory pool will be merged into the existing route network. The other part is then directly added to the current trajectory pool (details in Section V).

IV. Trajectory Data Pre-processing

In this section, we describe two components: data cleaning and trajectory classification.

A. Trajectory Data Cleaning

Raw aircraft trajectory data at airport is noisy and lacks important features such as headings and speed. Examples of noise include aircraft GPS data that are either outside the boundaries of the airport or reflect the trajectories data of aircraft in the

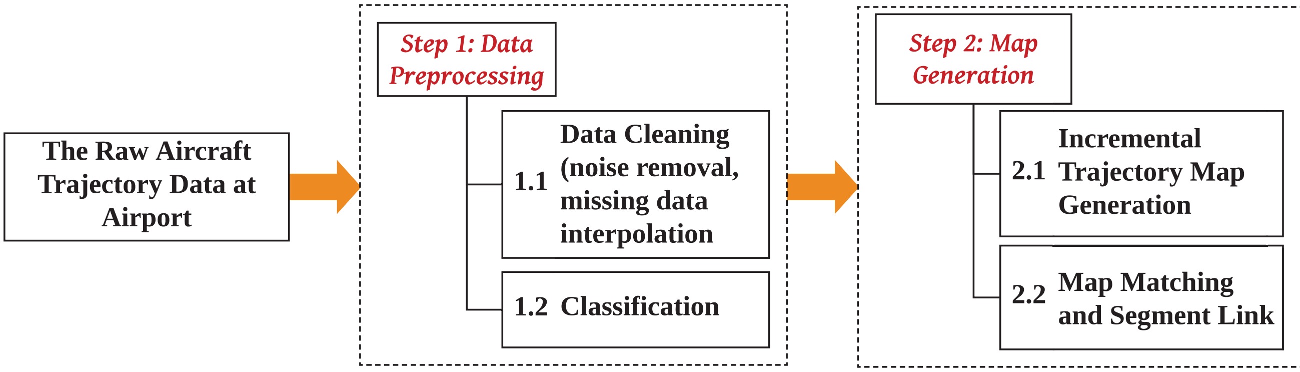


Fig. 2. Overview of the situational awareness map construction framework.

air. These are irrelevant and are filtered. We use a shape file which is a popular geographical spatial vector data format for geographical information software to extract airport aircraft GPS data from raw data and calculate the speed and headings of GPS points with the positions of x_i and x_{i-1} [33]. Additionally, the aircraft GPS points data do not contain the trajectory ID information. We group aircraft trajectories according the aircraft call signs and the time span to identify each GPS point x_i with a trajectory ID $\mathbf{t}(x_i)$. Aircraft trajectory data are complicated and nosily. Beside the simple data cleaning method, we also adopt the method from [34]. For example, we unify the spatial resolution and temporal resolution. We discard many noncontinuous trajectories and ground vehicles trajectories. We also consider about the positioning accuracy based on radar and GPS measurement error rate to remove the nosily points. We removed all singular points which is far away from other GPS points. Nevertheless, GPS noise cannot be fully filtered in our proposed method. Fortunately, it does not significantly influence our result because the size of aircraft is much larger than range of GPS error.

B. Trajectory Classification

Not all of airport aircraft trajectories data are useful to construct route networks. Some trajectory data are even have negative effect on extracting real route network from data. For instance, parking aircraft often stay in the parking zone and taxiing around one central location. This kind of trajectory does not reveal any route information. It not only wastes processing time but also mislead the airport traffic controller if we use such trajectory to generate the map. In this paper, we only use landing and take-off trajectories to generate the map. The aircraft traces can be classified into four categories: parking, patrolling, landing and take-off. We define each class of airport aircraft trajectory using spatio-temporal information as below:

Parking trajectory Parking trajectory $\mathbf{t_{parking}}$ is a trajectory $\mathbf{t}$ which only moves in a small area with the range R. Formally speaking, $\forall x_i, x_j \in \mathbf{t_{parking}}$, the distance between the x_i and x_j is less than the value R. The other condition is that the altitude of any $x_i \in \mathbf{t_{parking}}$ is the zero, which suggests that the trajectory is always on the ground (Here the altitude equals to zero is not the sea level but the airport ground level).

Patrolling trajectory Patrolling trajectory $\mathbf{t_{patrolling}}$is a trajectory with movable range of motion is larger than R and the altitude of any $x_i \in \mathbf{t_{patrolling}}$ equal to zero.

Landing trajectory Landing trajectory $\mathbf{t_{landing}}$ starts from the landing point of the aircraft and ends up at the parking space.

Take-off trajectory Take-off trajectory $\mathbf{t_{takeoff}}$ is recorded from the parking position to the take-off position.

We distinguish the landing and take-off trajectories by following simple criteria: $\exists x_i \in \mathbf{t}$ and the altitude of x_i is greater than zero. Meanwhile, $\sum_{i<\frac{n}{2}} speed_i > \sum_{i>\frac{n}{2}} speed_i$. Then the $\mathbf{t}$ is the landing trajectory. If the $\sum_{i<\frac{n}{2}} speed_i < \sum_{i>\frac{n}{2}} speed_i$, the $\mathbf{t}$ is the take-off trajectory. This method is simple but useful for telling the difference between the landing and take-off trajectories in reality because the speed of a landing aircraft decreases with the time, while and that of an take-off aircraft taking off speed increases with the time.

The functional areas we defined above are not as same as their real function at the airport. We classify those areas into four categories because we aim to filter useless trajectories and use the most important trajectories to construct situation awareness map. Therefore, we simply the functionality of these trajectories.

V. Airport Situational Awareness Map Generation Approach

In this section, we propose an approach to constructing the airport situational awareness map from multiple aircraft trajectories which are generated from the pre-processing step. We first outline the framework of the proposed incrementally-updated approach and then describe each step in details.

A. Incremental Updating Airport Framework

1) Approach: The intuition of the incremental framework is to expand the road network with new trajectories through merging and sampling. This is inspired by some known patterns of airports, as described below.

Road segment homogeneity Different aircraft trajectories on the same road segment are usually similar spatially and temporally. For example, the speeds and headings of GPS points of aircraft trajectories on the same road segment are similar within the same trajectory type, such as landing or take-off. Hence, GPS points with similar features and closer positions are more likely to be on the same road segment.

Road junction heterogeneity The GPS points located in junction areas are likely to be heterogeneous in both the spatial and temporal domains. A junction area can be viewed as a connecting joint of several road segments. Each segment can contain many different trajectories with different attributes. The GPS points of the trajectories of different road segments also differ in speed, heading and other spatial and temporal features.

Centreline with high density Many studies claim that road centreline have higher point densities than road edges [25], [35], which suggests that most GPS points in the same road segment are likely to locate around the centreline. Inspired by this idea, we also found that the average trajectory on a given road segment is also around its centreline.

Taking these observations into consideration, our algorithm has three phases:

Stage 1: Classifying The algorithm starts by classifying the GPS points of new trajectories into two categories: existing GPS points belonging to the trajectory pool and new GPS points that are to be added to the trajectory pool.

Stage 2: Merging and Adding In this stage, the algorithm separately handles the two classes of GPS points generated from Stage 1. The approach merges the similar GPS points and adds the new points, which can explore the new route in the map to the pool.

Stage 3: Segment Linking In the last stage, the GPS points need to be linked as segments. We use segment lists to represent the trajectories.

2) Algorithm Design: Algorithm 1 gives the pseudo-code of our incremental airport map expansion algorithm. In the initialisation stage, one of the trajectories is set as the initial airport map or put into the trajectory pool. Then, for each iteration of the airport map expansion (Lines 1 to 14), the algorithm will update the airport map with new coming trajectories. The GPS data of the new trajectories are classified into two categories in Line 3. If the GPS point x_i is similar to points in the trajectory pool, x_i is merged with its similar points in the neighbourhood (Line 5). Otherwise, we add the new points to the trajectory pool (Line 9). The new edges also are generated to connect new points in the pool (Lines 7, 13). Finally, when all new trajectories have been used, the algorithm terminates, and the graph $\mathbf{G}$ is returned as the map.

We define some terms in Algorithm 1 in more details as follows:

Classify() The aim of the classifying function is to classify the GPS points x_i into $Class_A$ and $Class_B$. The GPS point

Algorithm 1 Framework of Incremental Airport Map Expansion

Input: Trajectory Dataset $\mathbf{T}$
Output: Road network $\mathbf{G} = (\mathbf{V}, \mathbf{E})$
Initialisation : Select first trajectory $\mathbf{t_1} \in \mathbf{T}$ and initialise the trajectory pool as $\mathbf{G} \leftarrow \mathbf{t_1}$
1: **for** $i = 2$ to n **do**
2: **for** $\forall x_j \in \mathbf{t_i}$ **do**
3: *Stage 1:* Classify(x_j)
4: **if** ($x_j \in Class_A$) **then**
5: $x_j^\star = \text{Merge}(x_j, x_k \in \Phi(x_j))$
6: $\mathbf{V} \leftarrow \mathbf{V} - x_j \cup x_k \in \Phi(x_j)$
7: $\mathbf{E} \leftarrow \mathbf{E} - E_r$, $E_r = (x_p, x_q)$ and x_p or $x_q \in x_j \cup \Phi(x_j)$
8: **else if** ($x_j \in Class_B$) **then**
9: $\mathbf{V} \leftarrow x_j$
10: **end if**
11: **end for**
12: generate $\mathbf{E}^\star$ from $x_j \in class_B$ with τ_i
13: $E \leftarrow E + E^\star$
14: **end for**
15: **return** $\mathbf{G}$

$x_i \in Class_A$ lies on the existing map $\mathbf{G}$, and the other points that belong to $Class_B$ belong to new road segments.

Merge() The Merge operation aims to merge the GPS point x_i in new trajectory with its similar neighbours $\Phi(x_i)$. $x_j^\star$ denotes the new merged GPS point and $\mathbf{E}^\star$ denote the new generated edges between points in $Class_B$ and map $\mathbf{G}$. E_r is the new edge that connects newly merged points and other points in map $\mathbf{G}$.

B. New Trajectory GPS Points Classification

Since the trajectories of aircraft are noisy and redundant, it is important to distinguish redundant data from new trajectory data. Therefore, we propose a new GPS point classification method that employs both spatial and temporal information. The motivation and details are discussed in this section.

1) Approach: Any road segment has multiple aircraft trajectories on it. Aircraft routes on a given road segment should be similar, as the boundary and direction of each trail is fixed. Aircraft trails are different from those of other common vehicles, as mentioned before. In Section IV, all aircraft trajectories were classified into four classes. For each class, aircraft actions and trajectories are similar. For example, landing aircraft always follow the same actions: landing on a specific runway, taxiing along the runway and going to the parking area according to the air traffic controllers' instructions. For most cases, the patterns of aircraft landing areas are similar. That is, aircraft speeds and trails are similar when they are located in the same road segment, which suggests that the features of GPS points located in the same road segment will be similar.

2) Methodology: The classification approach is to check if GPS point x_i is located on existing map $\mathbf{G}$. Aircraft trajectory data is typical spatio-temporal data. It is necessary to analyse the similarity of GPS points both spatially and temporally. In the spatial perspective, similar GPS points should be located in close proximity and their directions should be similar. In the temporal perspective, speed similarities should be taken into consideration.

We need to find the K nearest neighbours to merge in the spatial dimension. The amount of aircraft trajectory GPS data is huge. Therefore, a kd-tree was used in our algorithm to search the k nearest neighbours in the spatial domain [36], [37]. The kd-tree runs in $O(\mathbf{M}\log\mathbf{M})$ time, which can rapidly find K nearest neighbours within a short period. Additionally, quan tree and other tree structures can be applied to the neighbour search problem [38]. For each neighbour $x_j \in \Phi(x_i)$, we use speed and heading information to check whether they are similar in both the spatial and temporal domains. We use a score function (Eq.1) to measure the similarity between new GPS points x_i and their neighbours.

$$score(x_i, x_j) = \frac{\sum_{||x_i-x_j||<R} e^{-\Delta^2(h_i,h_j)} \cdot e^{-||x_i-x_j||^2}}{\sum_{||x_i-x_j||<R} e^{-||x_i-x_j||^2}} \quad (1)$$

where $||\cdot||^2$ measures the Euclidean distance between two GPS points, Δ gives the angle between two headings, and R denotes the radius of the neighbour areas.

In the temporal domain, we assume that speed of each GPS point x_i is associated with a Gaussian distribution $N \sim (\mu, \sigma)$, where μ denotes the average speed of $x_j \in \Phi(x_i)$ and σ is the standard deviation of the GPS point speeds in the neighbour area. The probability of x_i belonging to the same segment as other neighbouring GPS points can be calculated as follows:

$$P_{speed} = |\int_{speed_i}^{\mu} N(\mu, \sigma) d(speed)| \quad (2)$$

Considering both spatial and temporal criteria, the final probability of GPS point x_i belong to the $Class_A$ equals:

$$w_j = score(x_i, x_j) \times P_{speed} \quad (3)$$

C. Merge approach

In the GPS point merging step, we use a simple midpoint method to estimate the new GPS point location. The new GPS point is calculated by finding the centre point for the similar neighbour GPS points. The new GPS point $x_j^\star = (lat, lon)$ is calculated by the following equation

$$\begin{aligned} lat_j^\star &= \frac{1}{m} \sum_{x_j \in Class_A} lat_j \\ lon_j^\star &= \frac{1}{m} \sum_{x_j \in Class_A} lon_j \end{aligned} \quad (4)$$

where the m is the number of similar points in the $\Phi(x_i)$

D. Inferring Edge Link

The previous work sampled a large amount of GPS data to generate new GPS points representing trajectories. The vertices of map G were extracted from the aircraft GPS point cloud. In this part, we aim to determine the edges $\mathbf{E}$ from trajectory information. Since the temporal information τ of each GPS point is recorded, the order information of GPS points is confirmed. The segmentation information is only lost after the merge-adding steps for those merged points. As a result, it is necessary to add edges after the merging and addtion of new points step. After the merging step, the simplest solution is to connect new merged point with points connected with $x_j \in \mathbf{\Phi}$. After the adding step, the new merge point replace the original points in the new trajectory. Therefore, the edge information remains. Nevertheless, the temporal information should is lost after merging and adding. However, the order information of GPS points is preserved.

VI. Results and Evaluation

In this section, we conduct extensive experiments to evaluate the effectiveness and efficiency of our proposed framework. The aircraft trajectory datasets are described firstly. Then we evaluate our function classification methods using ground truth extracted from aviation official website. We also evaluate the clustering approach using both subjective criteria such as visual analytic and object criteria such as completeness and precision metric.

A. Los Angeles International Airport Datasets

The Los Angeles International Airport (LAX) is one of the busiest airport in the United States. It has more than 10,000 aircraft trajectories per day. Each trajectory consists of dozens to thousands of GPS points. The volume of trajectory data is also extremely large, and can reach 20GB monthly in CSV format. Fortunately, it is not necessary to employ all existing trajectories to infer the situation awareness map at the airport. In this case study, a small subset of trajectories was extracted from Federal Aviation Administration (FAA) database. We selected data from 1 August to 31 August 2016 and pre-process all data using method we motioned in the pre-processing section.

A summary of this subset of data is given in Table II. All data was recorded by radars and sensors at the airport and in the aircraft.

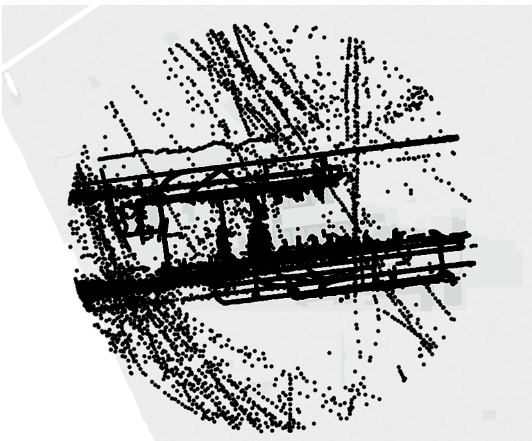
TABLE II
Overview of Los Angeles International Airport

Description	Value
Volume of dataset	138MB
Number of records	1080059
Number of features	10
Time span	2016-07-31 14:00:01 to 2016-08-01 13:59:59
Number of aircraft type	124
Number of callsigns	2091
Number of trajectories	2165

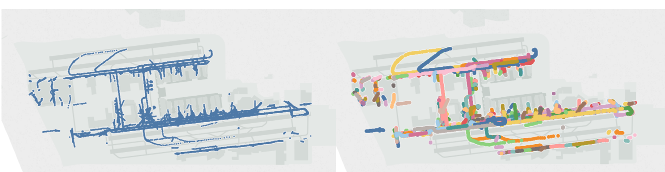
The number of records in Table II denotes the number of GPS points in the dataset. The features for each record include both aircraft GPS points and trajectory information such as longitude, latitude, trajectory ID, time stamp and etc. It also contains the some specific information for aircraft such as the type of the aircraft (example: Boeing 747), the call sign of aircraft (example, AAL2043).

B. Data Pre-processing Result

The raw dataset obtained from FAA was chaotic and noisy. Figure 3a shows a GPS points map of raw GPS points located in LAX area. We used the method given in Section IV to clean and pre-process the raw data.



(a) Raw aircraft GPS points map



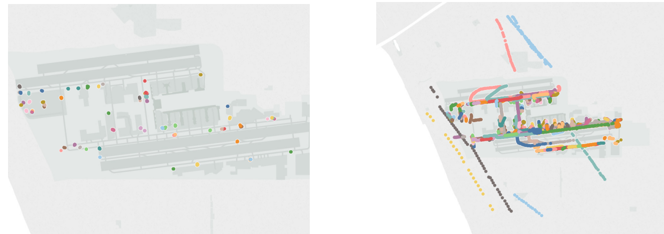
(b) Cleaned GPS points map (c) Cleaned trajectory map

Fig. 3. The raw data and the output trajectory data from step 1.1 in data preprocess.

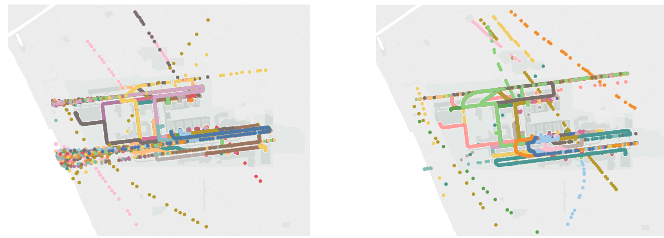
Figure 3 shows the raw and cleaned map. It contains all GPS points on the ground. Figure 3b shows all GPS points at the airport and Figure 3c denotes trajectories with colours. Compared with Figure 3a, the cleaned map removes all airborne GPS data and trajectories, which are irrelevant to the airport route network.

Figure 3c shows that the patterns of different trajectories are significantly different. As mentioned in Section IV, not all trajectories are useful in constructing an airport route network. We have discussed it in the Framework Section. Figure 4 illustrates all the classification result from the dataset.

In order to validate the effectiveness of our proposed solution, we evaluate our estimated result with the ground truth we extracted from aviation official website. Since the website only lists the take-off call sign and landing take-off, we can only validate the accuracy of landing trajectory estimation and take-off trajectory estimation. The confusion matrix of classification result is shown in Fig. 5. We can find that our trajectory classification performance is good. The



(a) Parking trajectory (b) Patrolling trajectory



(c) Landing trajectory (d) Take-off trajectory

Fig. 4. Four classes of aircraft trajectories at airport from step 1.2 in data preprocessing.

accuracy of the landing trajectory is 86% and take-off is 98%, which indicates our simple classification method performance good enough for the following clustering solution. Especially we only focus on landing and take-off trajectory map, the error which classifying landing to take-off is only 1% and classifying the take-off to landing is only 0.8%.

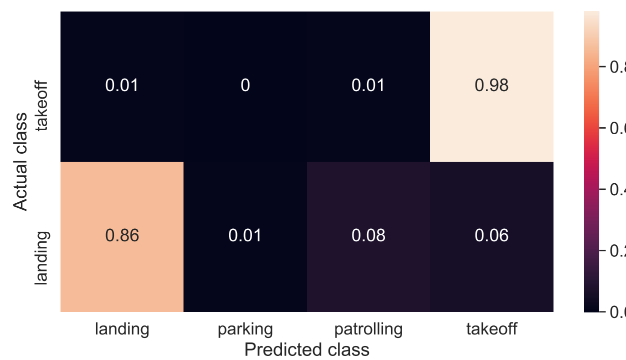


Fig. 5. Confusion matrix of trajectory classification.

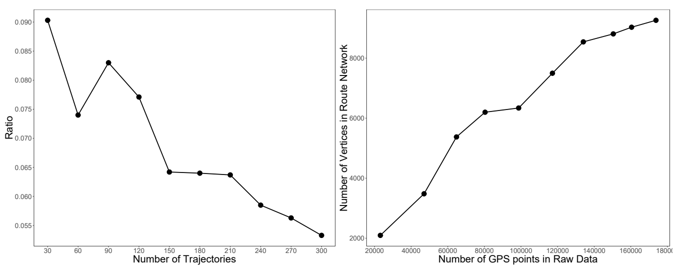
C. Evaluation of the Output Airport Map

In this section, we evaluate the quality of the map generated by our approach. We use both the state-of-art evaluation matrix proposed by Mahdi Hashemi [39] and some criteria particular to the characteristics of aircraft trajectory data.

1) Compression Ratio: Our approach aims to extract the on-ground aircraft trajectory map from a massive number of GPS points. Most existing GPS points are irrelevant for constructing the route network. We only retain key points and meaningful GPS points in the graph. The goal is to use a small number of vertices and trajectories to construct the airport route network of aircraft. Therefore, a ratio of the number of

GPS points in the raw dataset to constructed graph vertices is necessary to calculate. Since our algorithm are incremental, we would like to evaluate the ratio with increasing number of trajectories and GPS points. We evaluated the compression ratio in two aspects: 1) the ratio and number of trajectories we use and 2) the vertices and number of raw data points.

Figure 6 shows the compression ratio of vertices used to the raw GPS points in our approach. Figure 6a indicates that the compression ratio decreases with increasing number of trajectories. This is because most new trajectories are merged with previous ones as the route network graph expands. It also shows that with increasing numbers of trajectories used for updating the route network, our method needs fewer vertices to represent a new trajectory. Figure 6b shows the relationship between the number of GPS records used in our approach and the vertices in the constructed route network. The gradient of the line is the compression ratio. It shows that the gradient slight decrease with more GPS point used.

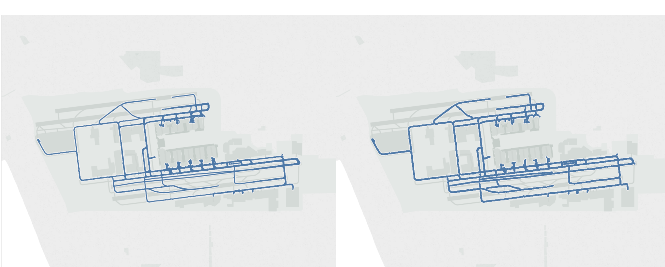


(a) Compression ratio and the number of trajectories

(b) The number of GPS records and vertices in route network

Fig. 6. Compression ratio.

2) Visual Comparison of Results: One direct and simple approach to assessing the result and estimating the performance of the algorithm is to visually compare the constructed map and a ground truth map. For this purpose, we present the construction result for an airport tarmac and runway map in Figure 7. Figure 7a is the map drawn by our hand. Figure 7b shows the generated map. It is difficult to to distinguish two figures from vision view, which indicates our result can represent the raw route network well.



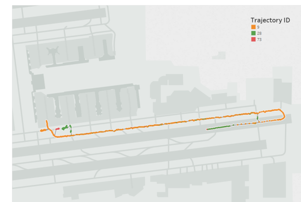
(a) Ground truth map

(b) Generated route network

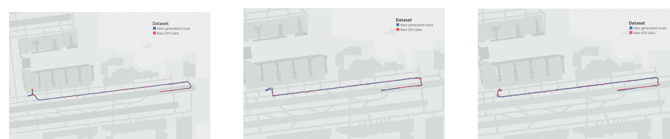
Fig. 7. Comparison of ground truth map and incrementally generated route network.

Figure 8 illustrates an example of comparing trajectories with raw data to newly generated graph with map matching. There are three trajectories in Figures 8a and 8b with IDs 9, 28 and 73. We use the different colour to denote the different trajectories. Figure 8c, 8d, and 8e shows the difference between original trajectories and map matching result in the newly generated graph. Figure 8g and Figure 8f show the local details of original trajectories and the map matching result.

(a) Example of trajectories in raw GPS data



(b) Example of trajectories in generated route network



(c) Trajectory 9 in the raw GPS data and newly generate route network

(d) Trajectory 28 in the raw GPS data and newly generate route network

(e) Trajectory 73 in the raw GPS data and newly generated route network

(f) Local difference between distinct trajectories in the raw GPS data

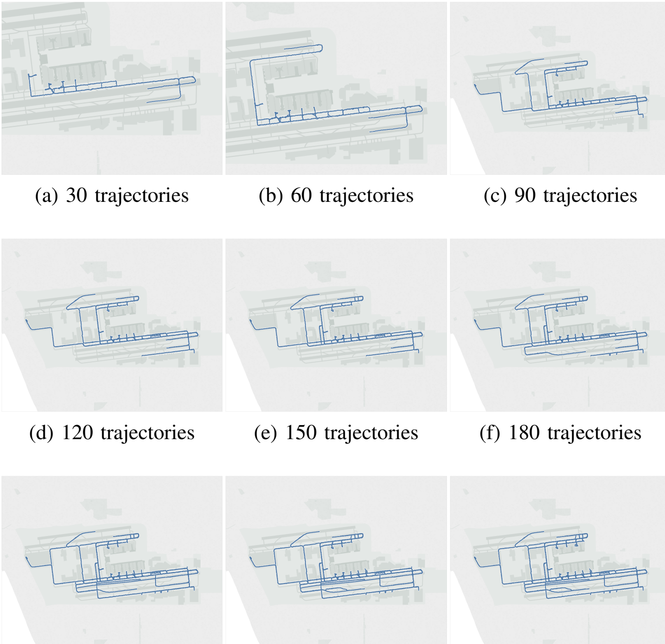
(g) Difference between distinct trajectories in generated route network

Fig. 8. An example of trajectories in raw GPS data and in a newly generated route network.

We present the construction process step-by-step in Figure 9. It shows the construction process with the addition of 30 to 270 trajectories. We can see that new routes are discovered as more trajectories are added in. As more trajectories are used in updating process, fewer routes are constructed with the same number of new trajectories, which also matches our compression ratio result: the compression ratio improves with more trajectories. This is because fewer vertices are needed to represent the same number of trajectories in a larger route network graph.

3) Evaluation metric: Hashemi et.al. proposed two quantitative metrics including completeness and precision to evaluate the quality of constructed route network [39]. We apply these two metrics to evaluate our results in this section.

The completeness metric shows how well the new route network covers the ground truth. The metric uses distances to match the segments in new constructed route network and



(a) 30 trajectories (b) 60 trajectories (c) 90 trajectories

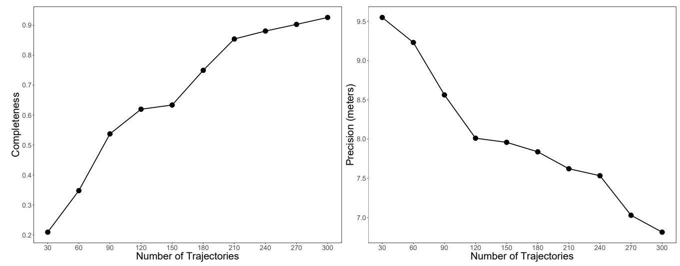
(d) 120 trajectories (e) 150 trajectories (f) 180 trajectories

(g) 210 trajectories (h) 240 trajectories (i) 270 trajectories

Fig. 9. An example of trajectories in raw GPS data and in a new generated route network.

segments in ground truth. The distance calculation method is shown in their paper [39]. For each segment in constructed route network, they are matched with the closed segment in ground truth. The length of matched segments in ground truth is denoted as l and the total length of segments in ground truth is L. The completeness is calculated as $completeness = l/L$. The precision metric indicates how close the constructed segments are to the ground truth counterparts. We use the average distance between matched segments to measure the precision.

Figure 10a shows the completeness with increasing number of trajectories and Figure 10b shows the precision metric. Here, we calculate the completeness metric based on the same ground truth route network. The completeness increases with more trajectories used because the more routes has been constructed by more trajectories. We can also find that the increase rate decreases slightly because the route information that new trajectories can provide is less than the trajectories used for route network construction. Many trajectories offer redundant route information if the trajectory can be map matching in the graph. The ground truth map is constructed by hand with observation of around 400 trajectories. Using 300 trajectories to construct the route network, the completeness achieved around 93%. The precision metric calculates the average distance between the constructed route network and matching ground truth graph. The range of distance is between 7 to 9 meters. That is, for each segment in constructed graph, we calculate the distance between each vertices in this segment and corresponding segment in the ground truth map. The average distance between those segments is the precision. We find that the precision become better with more trajectories, which suggests that more trajectories can boost the precision of location of segments in the constructed route graph.



(a) Completeness metric (b) Precision metric

Fig. 10. Evaluation metric with incremental updating approach from LAX dataset for the different number of trajectories used.

4) Running Time: We tested our framework on a PC with a 3.3GHz Intel Core i7 processor and 32GB RAM. We implemented the program in R [40]. Table III shows the running time of each part of our framework for the Los Angeles International Airport. Except for data preparation time, the total updating time was only around 45 seconds. If we use high performance languages such as C++ or Java, calculation could be made in real-time

TABLE III
RUNNING TIME OF LOS ANGELES INTERNATIONAL AIRPORT DATASET

Description	**Value**
#GPS points	1,080,059
#Trajectories	2,165
Data cleaning	45.38s
Trajectory classification	0.28s
Incremental updating	44.72s
Total running time	approx. 90s

VII. CONCLUSIONS

In this paper, we propose an incremental approach to construct on-ground aircraft map at Los Angeles International Airport from massive amount of GPS data. We formulate the on-ground aircraft trajectory map creation and updating problem, provide the approach to clean, and process huge volumes of on-ground aircraft GPS data. We classify aircraft trajectory into four categories: parking, patrolling, landing and take-off using a simple solution. We also propose a new clustering algorithm to merge existing routes and add new routes with new coming trajectories, which is able to update the map in time. We conducted an extensive experiment on a large amount of aircraft trajectory data from Los Angeles International Airport to demonstrate the effectiveness in three aspects: compression ratio, visualisation comparison and evaluation metric. Our work still has some limitations. In the future, we will explore more incremental clustering methods and extend this framework to construct other types of trajectory map. We also plan to test our proposed solution to other airports and similar scenarios.

References

[1] C. Chen, X. Liu, T. Qiu, and A. K. Sangaiah, "A short-term traffic prediction model in the vehicular cyber–physical systems," *Future Generation Computer Systems*, 2017.

[2] C. Lundquist, T. B. Schön, and F. Gustafsson, *Situational awareness and road prediction for trajectory control applications*. London: Springer London, 2012, vol. 1-2, pp. 365–399.

[3] F. Seraj, N. Meratnia, and P. J. Havinga, "Rovi: Continuous transport infrastructure monitoring framework for preventive maintenance," in *Pervasive Computing and Communications (PerCom), 2017 IEEE International Conference on*. IEEE, 2017, pp. 217–226.

[4] G. Andrienko, N. Andrienko, G. Fuchs, and J. M. C. Garcia, "Clustering Trajectories by Relevant Parts for Air Traffic Analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 34–44, 2018.

[5] D. Bertsimas and S. S. Patterson, "The air traffic flow management problem with enroute capacities," *Operations research*, vol. 46, no. 3, pp. 406–422, 1998.

[6] X. Kong, Z. Xu, G. Shen, J. Wang, Q. Yang, and B. Zhang, "Urban traffic congestion estimation and prediction based on floating car trajectory data," *Future Generation Computer Systems*, vol. 61, pp. 97–107, 2016.

[7] M. Pusadan, J. Buliali, and R. Ginardi, "Anomaly detection of flight routes through optimal waypoint," in *Journal of Physics: Conference Series*, vol. 801. IOP Publishing, 2017, p. 012041.

[8] L. Li, M. Gariel, R. J. Hansman, and R. Palacios, "Anomaly detection in onboard-recorded flight data using cluster analysis," in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, 2011, pp. 4A4–1–4A4–11.

[9] M. Conde Rocha Murca, R. DeLaura, R. J. Hansman, R. Jordan, T. Reynolds, and H. Balakrishnan, "Trajectory Clustering and Classification for Characterization of Air Traffic Flows," in *16th AIAA Aviation Technology, Integration, and Operations Conference*. American Institute of Aeronautics and Astronautics, 2016. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2016-3760

[10] M. S. Rahaman, M. Hamilton, and F. D. Salim, "Predicting Imbalanced Taxi and Passenger Queue Contexts in Airport," in *Proceedings of the Pacific Asia Conference on Information Systems (PACIS)*, 2017.

[11] M. S. Rahaman, Y. Mei, M. Hamilton, and F. D. Salim, "Capra: A contour-based accessible path routing algorithm," *Information Sciences*, vol. 385-386, pp. 157 – 173, 2017.

[12] M. S. Rahaman, M. Hamilton, and F. D. Salim, "Queue context prediction using taxi driver knowledge," *Proc. of the Knowledge Capture Conference (K-CAP)*, 2017.

[13] A. Sadri, F. D. Salim, Y. Ren, W. Shao, J. C. Krumm, and C. Mascolo, "What will you do for the rest of the day?: An approach to continuous trajectory prediction," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 4, pp. 186:1–186:26, Dec. 2018.

[14] S. Edelkamp and S. Schrödl, "Route planning and map inference with global positioning traces," *Computer Science in Perspective*, pp. 128–151, 2003.

[15] L. Cao and J. Krumm, "From gps traces to a routable road map," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. ACM, 2009, pp. 3–12.

[16] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Robust and accurate road map inference," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3946–3953.

[17] A. Fathi and J. Krumm, "Detecting road intersections from gps traces," *Geographic Information Science*, pp. 56–69, 2010.

[18] J. Qiu and R. Wang, "Road map inference: A segmentation and grouping framework," *ISPRS International Journal of Geo-Information*, vol. 5, no. 8, p. 130, 2016.

[19] Z. Chen, H. T. Shen, and X. Zhou, "Discovering popular routes from trajectories," in *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*. IEEE, 2011, pp. 900–911.

[20] M. Ahmed and C. Wenk, "Constructing street networks from gps trajectories." in *ESA*. Springer, 2012, pp. 60–71.

[21] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 330–339.

[22] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, "Wherenext: a location predictor on trajectory pattern mining," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 637–646.

[23] Z. Yin, L. Cao, J. Han, J. Luo, and T. Huang, "Diversified trajectory pattern ranking in geo-tagged social media," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 980–991.

[24] L.-Y. Wei, Y. Zheng, and W.-C. Peng, "Constructing popular routes from uncertain trajectories," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 195–203.

[25] E. Uduwaragoda, A. Perera, and S. Dias, "Generating lane level road data from vehicle trajectories using kernel density estimation," in *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013, pp. 384–391.

[26] S. Karagiorgou, D. Pfoser, and D. Skoutas, "A layered approach for more robust generation of road network maps from vehicle tracking data," *ACM Trans. Spatial Algorithms Syst.*, vol. 3, no. 1, pp. 1–21, 2017.

[27] B. Kuijpers, B. Moelans, W. Othman, and A. A. Vaisman, "Analyzing trajectories using uncertainty and background information." in *SSTD*. Springer, 2009, pp. 135–152.

[28] D. Pfoser and C. Jensen, "Capturing the uncertainty of moving-object representations," in *Advances in Spatial Databases*. Springer, 1999, pp. 111–131.

[29] M. Nanni and D. Pedreschi, "Time-focused clustering of trajectories of moving objects," *Journal of Intelligent Information Systems*, vol. 27, no. 3, pp. 267–289, 2006.

[30] F. Giannotti, M. Nanni, D. Pedreschi, F. Pinelli, C. Renso, S. Rinzivillo, and R. Trasarti, "Unveiling the complexity of human mobility by querying and mining massive trajectory data," *The VLDB JournalThe International Journal on Very Large Data Bases*, vol. 20, no. 5, pp. 695–719, 2011.

[31] J. Li, Q. Qin, J. Han, L.-A. Tang, and K. H. Lei, "Mining trajectory data and geotagged data in social media for road map inference," *Transactions in GIS*, vol. 19, no. 1, pp. 1–18, 2015.

[32] W. Shao, F. D. Salim, A. Song, and A. Bouguettaya, "Clustering big spatiotemporal-interval data." *IEEE Trans. Big Data*, vol. 2, no. 3, pp. 190–203, 2016.

[33] E. ESRI, "Shapefile technical description, jul. 1998," 1998.

[34] G. Andrienko, N. Andrienko, and G. Fuchs, "Understanding movement data quality," *Journal of location Based services*, vol. 10, no. 1, pp. 31–46, 2016.

[35] C. Chen, C. Lu, Q. Huang, Q. Yang, D. Gunopulos, and L. Guibas, "City-scale map creation and updating using gps collections," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1465–1474.

[36] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[37] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 891–923, 1998.

[38] H. Samet, "The quadtree and related hierarchical data structures," *ACM Computing Surveys (CSUR)*, vol. 16, no. 2, pp. 187–260, 1984.

[39] M. Hashemi, "A testbed for evaluating network construction algorithms from gps traces," *Computers, Environment and Urban Systems*, vol. 66, pp. 96–109, 2017.

[40] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: http://www.R-project.org/