Summer 2017

Algorithms for Constructing Vehicle Trajectories in Urban Networks Using Inertial Sensors Data from Mobile Devices

Umana Ahmed
Old Dominion University

ALGORITHMS FOR CONSTRUCTING VEHICLE TRAJECTORIES IN URBAN NETWORKS USING INERTIAL SENSORS DATA FROM MOBILE DEVICES

by

Umama Ahmed
B.S. December 2008, Bangladesh University of Engineering and Technology
M.S. August 2010, Marquette University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MODELING AND SIMULATION

OLD DOMINION UNIVERSITY
August 2017

Approved by:

Dr. Mecit Cetin (Director)

Dr. Rick McKenzie (Member)

Dr. Roland Mielke (Member)

Dr. Andrew Collins (Member)

Dr. Rajesh Paleti (Member)

ABSTRACT

ALGORITHMS FOR CONSTRUCTING VEHICLE TRAJECTORIES IN URBAN NETWORKS USING INERTIAL SENSORS DATA FROM MOBILE DEVICES

Umama Ahmed
Old Dominion University, 2017
Director: Dr. Mecit Cetin

Vehicle trajectories are an important source of information for estimating traffic flow characteristics. Lately, several studies have focused on identifying a vehicle's trajectory in traffic network using data from mobile devices. However, these studies predominantly employed GPS coordinate information for tracking a vehicle's speed and position in the transportation network. Considering the known limitations of GPS, such as, connectivity issues at urban canyons and underpasses, low precision of localization, high power consumption of device while GPS is in use, this research focuses on developing alternate methods for identifying a vehicle's trajectory at an intersection and at a urban grid network using sensor data other than GPS in order to minimize GPS dependency. In particular, accelerometer and gyroscope data collected using smartphone's inertial sensors, and speed data collected using an on-board diagnostics (OBD) device, are utilized to develop algorithms for maneuver (i.e., left/right turn and through), trip direction, and trajectory identification.

Different algorithms using threshold of gyroscope and magnetometer readings, and machine learning techniques such as k-medoids clustering and dynamic time warping are developed for maneuver identification and their accuracy is tested on collected field data. It is found that, clustering based on maximum and minimum value of gyroscope readings is effective for maneuver identification. For trip direction identification at an intersection, two different methods are developed and tested. The first method utilizes accelerometer, gyroscope and OBD speed data, and the 2nd method employs magnetometer and acceleration data. The results

demonstrate that the developed method using accelerometer, gyroscope and OBD speed data are effective in identifying a vehicle's direction. An effective algorithm is developed using OBD speed information, maneuver and trip direction identification algorithms to identify vehicle's trajectory at a grid network. Techniques for noise removal and orientation correction to transfer the raw data from phone's local coordinate to global coordinate system are also demonstrated.

Overall, this research eliminates the need for continuous GPS connectivity for trajectory identification. This research can be incorporated in methods developed by researchers to estimate traffic flow, delays, and queue lengths at intersections. This information can lead to better signal timings, travel recommendations, and traffic updates.

This dissertation is dedicated to my family for their endless support and encouragement.

ACKNOWLEDGMENTS

The successful completion of my dissertation would not be possible without the help and support from many people. First and foremost, I would like to thank my adviser Dr. Mecit Cetin for his guidance, support and consideration.

I extend many, many thanks to the committee members, Dr. Rick McKenzie, Dr. Roland Mielke, Dr. Andrew Collins, and Dr. Rajesh Paleti, who guided me with their valuable inputs.

I would like to thank my friends and colleagues who made this journey an enjoyable one.

The credit for my successful completion of PhD degree also goes to my father, Faruque Ahmed, and my mother, Nurjahan Begum. Their continuous support and prayers helped me to come this far.

My heartiest thanks to my husband, Dr. Mostafa Uddin. He always motivated and encouraged me to work harder and to aim for bigger achievements. Finally, I would like to mention my daughter, Aaima Mustafa, who is an integral part of this journey. Thank you for bringing pure joy and blessings to my life.

Lastly, thanks to the Almighty Allah for giving me strength to take challenges and for His countless blessings in my life.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

Page

LIST OF FIGURES

Figure Page

Page

Page

CHAPTER 1
INTRODUCTION

In recent years, traffic congestion has become a substantial problem for most of the big cities around the world. According to a study focusing on the economic impact of congestion on urban cities, in 2011, urban Americans had to travel 5.5 billion hours more and purchase an extra 2.9 billion gallons of fuel because of traffic congestion, resulting into a total cost of congestion of $121 billion [1]. The essential step to control congestion is traffic monitoring, through which the necessary information to take appropriate traffic control measures for relieving congestion can be obtained. Traditionally, traffic monitoring has been performed using dedicated sensing infrastructure such as loop detectors, radars, video cameras, etc. Because of the high cost of installation and maintenance of these sensing infrastructures, the traffic networks covered and monitored are limited to mostly highways and major arterial streets. For the last few years, a promising approach of inferring real time traffic information via intelligent transportation system (ITS) applications installed in mobile devices is becoming popular, because of lower cost and prospect of more global coverage of traffic network. Studies utilizing data from mobile devices to estimate traffic flow, lane detection, and trajectory identification have been growing as a result. However, studies focusing on trajectory identification using mobile data predominantly employed GPS coordinate information[1]. GPS has some significant limitations. Proper functioning of GPS requires clear signal reception from GPS satellites. In urban cities with dense high-rise buildings and under passes, the signals can get obstructed, causing GPS outage for an extended period of time. In addition, GPS is vulnerable to spoofing attacks [2]. In some situations, blockages of some satellites but not others occur, resulting into large error in

[1] IEEE Transactions and Journals style is used in this thesis for formatting figures, tables, and references.

positional coordinates. High power consumption of device while GPS is in use, is another limitation of using continuous GPS localization [3, 4].

This research is focused on developing methods to identify a vehicle's trajectory at an intersection and at an urban grid network utilizing inertial sensor data from mobile devices. Particularly, this research utilizes smartphone's inertial sensor data: gyroscope, accelerometer data, and speed data collected using an on-board diagnostics (OBD) device. The objective of this research is to find a vehicle's trajectory without relying on information collected from GPS, so that the trajectory information can be supplemented in case of GPS outage.

At a four-legged intersection, a vehicle can execute mainly 3 maneuvers from each leg of intersection: left turn, right turn and through maneuver, resulting into total 12 possible movements at the intersection (Fig. 1). Thus, this research focuses on developing methods for maneuver identification (left turn, right turn, and through) and trip direction identification at an intersection to construct vehicle's trajectory at the intersection. Next, as at an urban grid network, for same origin and same destination, there exist multiple routes (as shown in Fig. 2), this research focuses on developing methods to identify vehicle's trajectory at the grid network. Due to sensitivity of the inertial sensors, the raw data collected by the phone contains noises. Thus, applying appropriate noise removal technique is an important preprocessing step before any algorithm development. In addition, the phone can remain in any random orientation inside the vehicle. An appropriate orientation correction technique is thus necessary to be developed and applied on raw data. This will ensure that the develop trajectory identification algorithms are applicable to data from all phones regardless of their orientation inside the vehicle.

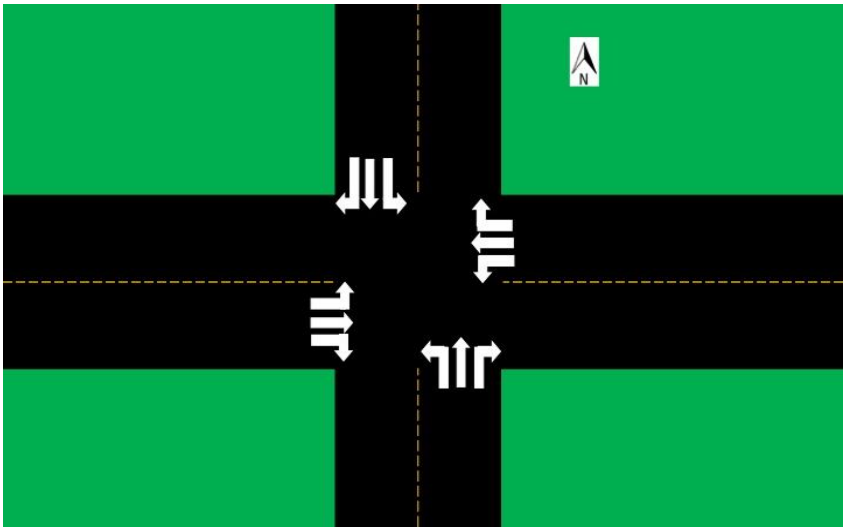

Fig. 1. Twelve possible maneuvers at a four-legged intersection.

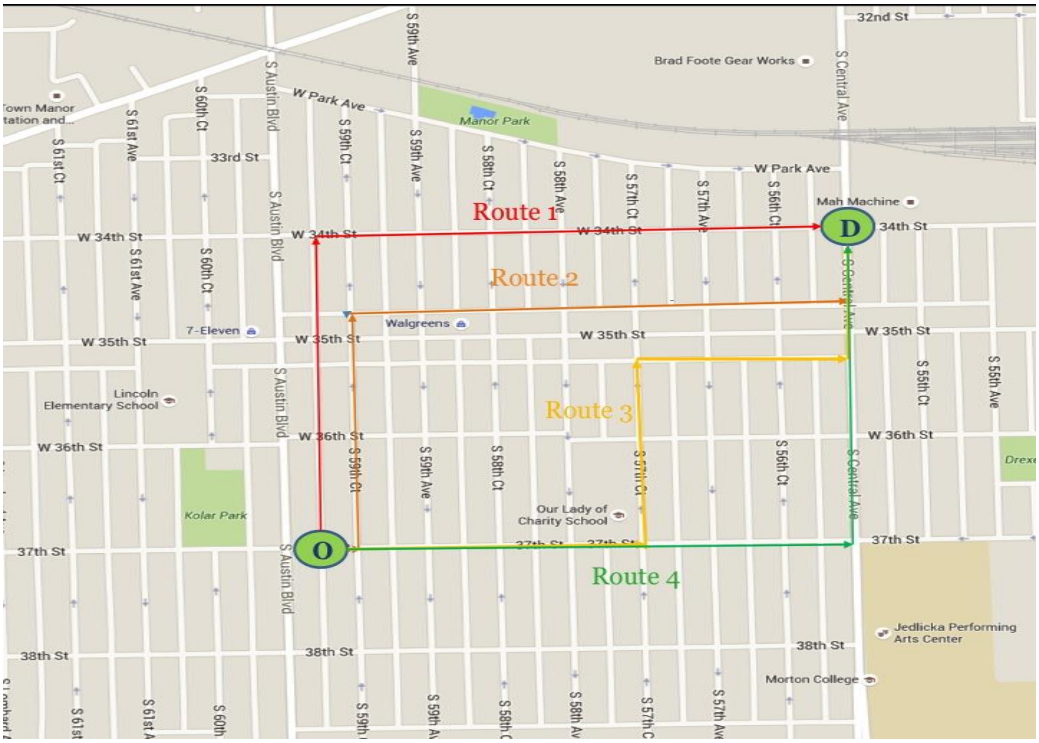

Fig. 2. An urban grid network with multiple routes between origin (O) to destination (D).

1.1 Objectives

The specific objectives of this research are as follows:

- To develop a method to identify a vehicle's trajectory at an intersection using a smartphone's inertial sensor data and OBD speed data.
- To investigate which feature(s) of the inertial sensor data provide the most useful information for identifying different maneuver: left turn, right turn, and through maneuver at an intersection and develop an algorithm for maneuver identification of a vehicle at an intersection.
- To develop a method to identify vehicle's direction through an intersection using a smartphone's inertial sensor data.
- To develop an algorithm that can identify vehicle' trajectory at an urban grid traffic network with known origin and destination information.
- To develop and employ appropriate noise removal technique and orientation correction technique.

1.2 Potential Application

The potential applications of this research are demonstrated by a use case scenario. Consider a scenario, where a vehicle is a part of a vehicular ad-hoc network [5]. A Roadside Unit (RSU) is installed in one of the traffic signal lights (Fig. 3) at a particular intersection. The RSU acts as a communication device and can communicate with the phone via Bluetooth. While within the RSU's range, the smartphone in the vehicle acts as an on-board unit (OBU) and transmits the collected data to the RSU intermittently [6, 7]. RSU sends the data packet to the server. This data packet includes inertial sensor data and OBD data collected by phone. This

research will identify the vehicle's trajectory at an intersection and thus can work as an input to estimate traffic flow and queue length at the intersection. This information will lead into better signal time optimization at the intersection. Similarly, at an urban traffic network, suppose two RSUs are installed at two intersections. A vehicle is traveling in the network and the phone is collecting the inertial sensor data and OBD data (Fig. 4). As soon as the vehicle comes near RSU 1, the phone sends a data packet to it. As the vehicle keeps moving, it goes out of the range of RSU 1. After a while, the vehicle reaches within the range of RSU 2 and the phone sends the collected data packets to it. The RSUs send the data packets to the server. Thus, RSU 1 can be considered the origin and RSU 2 can be considered the destination of the vehicle. However, there exist multiple routes to go from RSU 1 to RSU 2. This research will identify the vehicle's trajectory. This information can lead to quantifying traffic flow and travel time in each route, which in turn can be incorporated to provide better signal coordination between different intersections. Most importantly, this research eliminates the need for continuous GPS connectivity for trajectory identification.

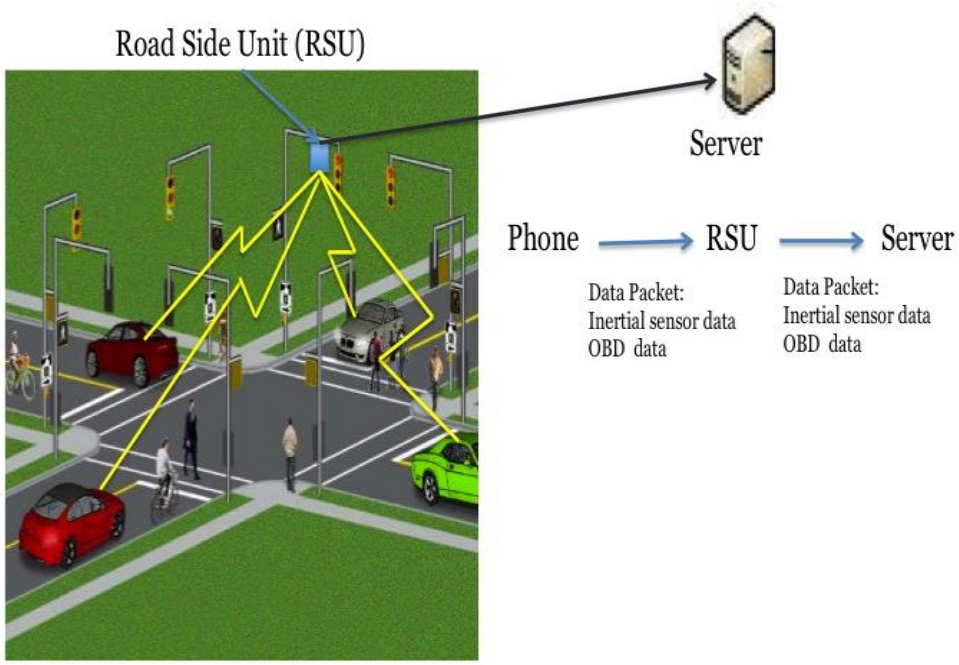

Fig. 3. Use case scenario 1.

Fig. 4. Use case scenario 2.

1.3 Organization of the Dissertation

Chapter 1 has introduction, objectives and potential applications of the work done in this dissertation. Chapter 2 contains the background of collected data from mobile devices and the data collection system for this research. Chapter 3 describes literature review on orientation correction techniques and trajectory identification methods. Chapter 4 includes overview of some of the machine learning methods that are used in the model development. Chapter 5 describes data prepossessing requirements and the noise removal techniques and orientation correction techniques. Chapter 6 includes the description of the methods explored and developed for maneuver identification. Chapter 7 describes the method for trip direction identification at an intersection. Chapter 8 includes the method for distance estimation and algorithm for trajectory identification at an urban grid network. Chapter 9 concludes the dissertation with the summary of findings, results, contributions, and possible applications of the developed models and future work.

CHAPTER 2
BACKGROUND ON DATA COLLECTION USING MOBILE DEVICES

Data collection using mobile devices has become an important technology for acquiring information regarding real-time speed, travel time, queue measurements, and volume estimation. Intelligent transportation system (ITS) applications that are specifically designed for collecting real-time traffic data are installed in the vehicles through various equipment, such as electronic tags, smart phones, Global Positioning System (GPS) receivers, etc. and transmit collected data using roadside receivers/transmitters, cellular phones, etc. [8].

This data collection system using mobile devices has several advantages over traditional traffic detectors. It has a low cost per unit of data, can continuously collect data as the probe vehicle travel through the network, the installation requires no traffic disruption, and data collection is automated. However, this system also has some disadvantages. For example, the initial development of the software that collects the data requires advanced software designing skills, may have high initial cost of purchasing and installing the devices, and raises privacy concerns of the drivers as their movement can be continuously monitored by tracking the probe data collected from their vehicles [8].

For this research, data have been collected using an application developed for smartphone and transmitted the collected data using cellular transmission. The detailed description of the application is provided in data description and preprocessing section.

2.1 Data Collection Using Smartphones

Modern smartphones are equipped with advanced sensing capabilities. They have in-built GPS receivers, proximity sensors, light sensors, accelerometer, gyroscope, and magnetometer. Some high-end smartphones even contain barometers, thermometers, and air humidity sensors.

The internal view of a smart-phone is shown in Fig. 5. The sensing data collected using accelerometers, gyroscopes, and magnetometers, and data from GPS receivers are within the scope of this study.

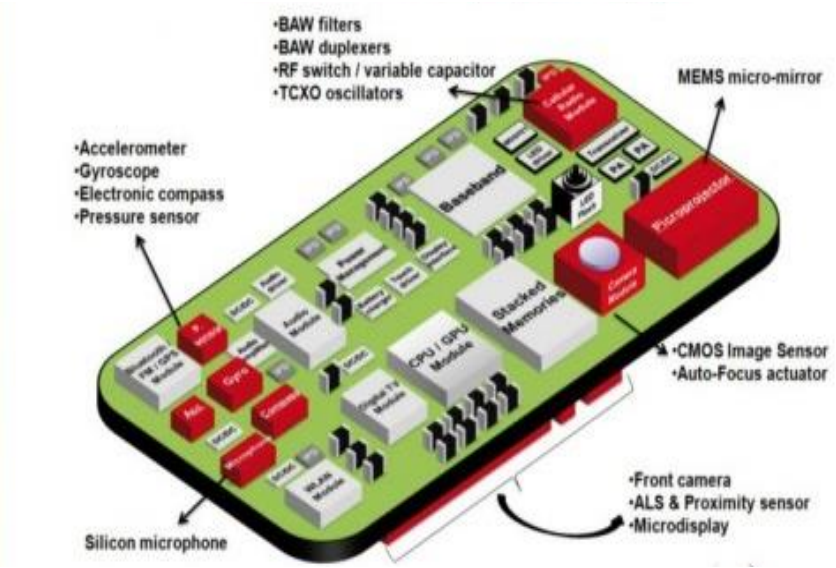

Fig. 5. Internal view of Smartphone board [11].

The built-in accelerometer can measure the acceleration and deceleration experienced by the phone. The acceleration values are provided in XYZ coordinate along phone's local axis in meter/second^2. The description and figure of phone's local axis is presented at the end of this section. The device rested in Earth also captures the gravitational acceleration ($g = 9.8\ ms^{-2}$). Thus the acceleration measured by a mobile device represents the total acceleration due to the motion of phone and the earth's gravity [9].

The gyro sensor detects the change in angular speed of the phone. The swinging, rotating, tilting motion is thus detected by the gyro sensor. The gyroscope is calibrated such that when

laid in horizontal plane, the gyroscope reading is zero. Any change in the orientation of the phone is detected by the gyroscope and change of angular speed corresponding to phone's local axis is provided in XYZ coordinate in radian/second. The in-built accelerometer/gyroscope sensor in a smartphone is shown in Fig. 6.

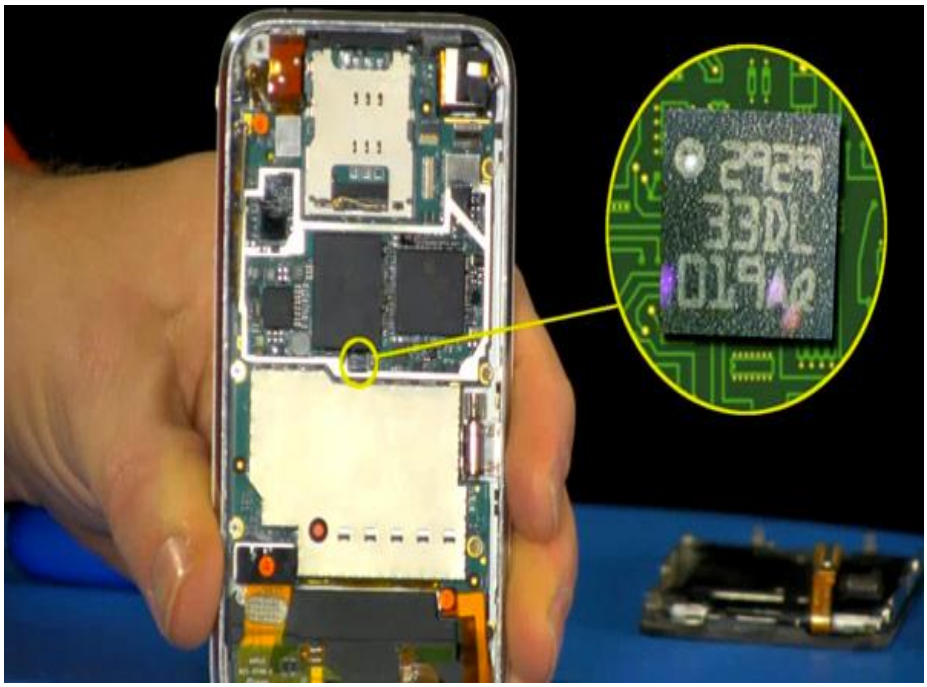

Fig. 6. Accelerometer/gyroscope sensor in Smartphone [12].

The GPS sensor inside the smartphone connects to 3 satellites and provides latitude and longitude information of the location of phone. Typical precision of GPS provided data is 20-50m, while maximum precision is 10m.

Magnetometer sensor measures the strength of the Earth's magnetic field in the phone's location and provides the values in XYZ coordinate in Micro Tesla (μT). The direction of the phone compared to the global North can be calculated using the values of Earth's magnetic field.

All the inertial sensor data are collected based on smart phone's local coordinate system. Phone's local coordinate frame is fixed in the center of it. The X axis of the local coordinate system is along the width of the phone and considered positive along the right side and negative toward the left side. The Y axis is along the length of the phone and positive towards the top and negative towards the bottom of the phone. The Z axis is perpendicular to the screen of the phone and positive towards outward direction [10]. Fig. 7 demonstrates the phone coordinate system.

When a user carries his smartphone while traveling in a vehicle, the smartphone can capture the acceleration and deceleration of the vehicle. The change in angular speed of the vehicle due to turn or bends along the path is captured by the smartphone using gyro sensors. The magnetometer data can be utilized to get the direction of magnetic north at the location of the vehicle.

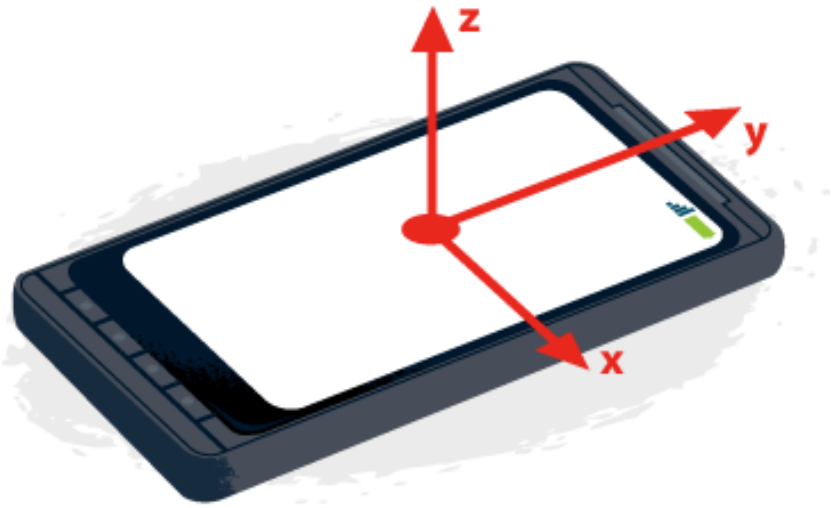

Fig. 7. Smartphones' local coordinate system [10].

CHAPTER 3
LITERATURE REVIEW

This chapter describes previous research works on coordinate conversion, turn identification, route matching, and activity identification.

3.1 Literature Review on Coordinate Conversion

As mentioned in the previous chapter, conversion of forces from phone's local coordinate to world coordinate is required to find signature pattern in movements of phone regardless of their individual orientation. Many previous studies have applied different coordinate conversion technique to transform the forces from local to the desired coordinate system.

Mohssen et al. utilized smartphone sensor data to estimate the user orientation relative to Earth's coordinate system [13]. They converted the smartphone data from a phone's local coordinate to a global coordinate system. They defined a 3 coordinate system: world coordinate system defined by North (N), East (E) and Gravity (-G); the smartphone's local coordinate system is defined as (X, Y, Z), and the user coordinate system defined by the user's forward direction (F), right side related to F (S) and (-G) (as shown in Fig. 8).

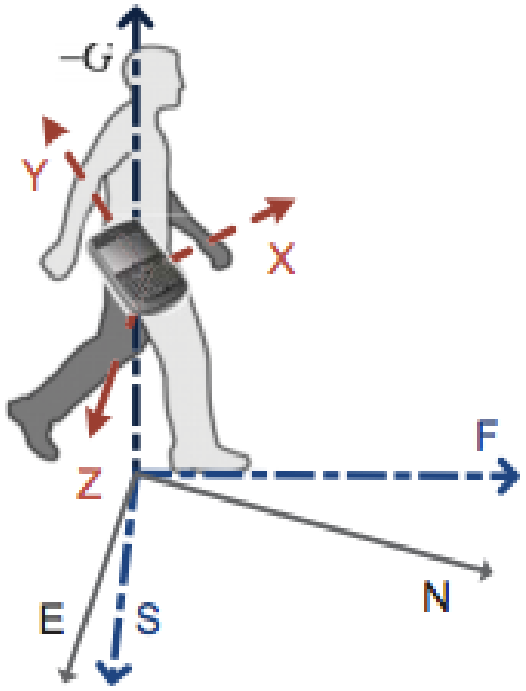

Fig. 8. Earth's, smartphone's and user's coordinate system [13].

Applying the concept of rotation matrixes Mohssen et al. determined the phone orientation angles pitch (α), roll (β) and yaw (γ). Yaw (γ) is the phone orientation angle relative to North. Fig. 9 demonstrates the angles. In order to do the determination, the authors utilized the raw magnetometer data and the gravity component of the phone's acceleration at its arbitrary location. These orientation angles are used for coordinate conversion of the phone's estimated forces from its local coordinate system to Earth's coordinate system. Finally, the raw acceleration values obtained at phone's local coordinate is converted to global coordinate by using quaternion conversion as follows by using $q.\ l..q*$. Where l is the linear acceleration in local coordinate, q is the quaternion vector and q^* is the conjugate of q. This converted acceleration value is used to find the direction of user relative to earth's plane [13].

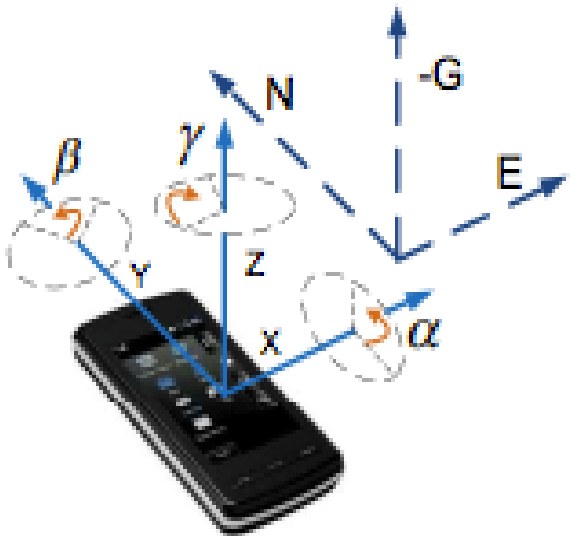

Fig. 9. Phone orientation angles [13].

Wang et al. used the concept of rotation matrix in order to align a phone's coordinate system with the vehicle's [14]. Fig. 10 shows how the authors defined the phone's coordinate system (Xp, Yp, Zp) and the vehicle's coordinate system (Xc, Yc, Zc). In order to find the rotation matrix that rotate phone's local coordinate to vehicle's coordinate, the authors defined three unit coordinate vectors under the vehicle's coordinate system as $\hat{\boldsymbol{\imath}}$, $\hat{\boldsymbol{\jmath}}$, and $\hat{\boldsymbol{k}}$.

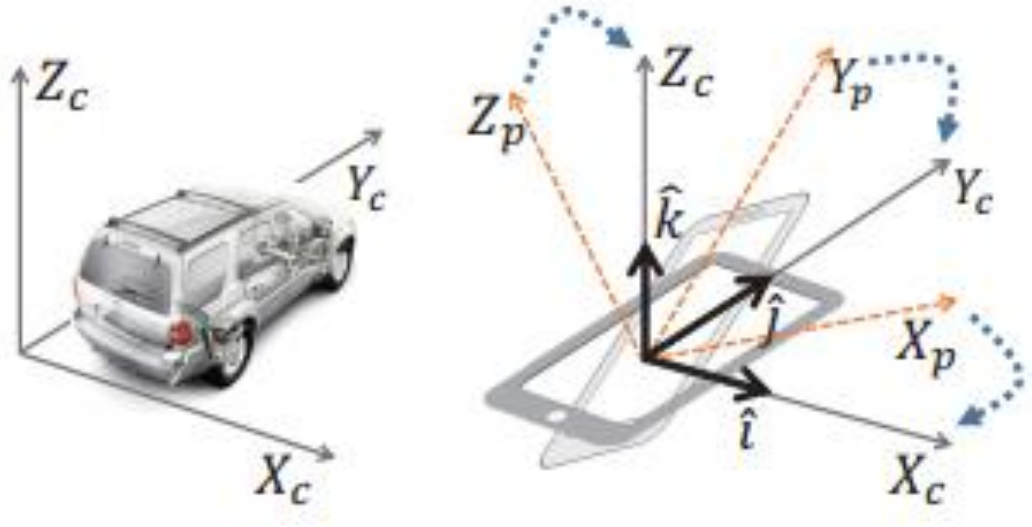

Fig. 10. Illustration of a phone's and vehicle's coordinate system [14].

Wang et al. utilized the acceleration and gyroscope data collected using the smartphone to obtain the rotation matrix consisting of three unit vectors $\hat{\boldsymbol{i}}$, $\hat{\boldsymbol{j}}$, and $\widehat{\boldsymbol{k}}$ along vehicle's local X, Y and Z axis [14]. The unit vector $\widehat{\boldsymbol{k}}$ is obtained by estimating the gravity component from the raw acceleration data. The authors applied a low pass filter on the three axis raw accelerometer readings to separate the constant components from those. These constant components are the gravity part of the acceleration and normalized to generate the unit vector $\widehat{\boldsymbol{k}}$. The unit vector $\hat{\boldsymbol{j}}$ is derived by extracting the linear component of acceleration while the vehicle decelerates. The unit vector $\hat{\boldsymbol{i}}$ is derived by vector multiplication of $\hat{\boldsymbol{j}}$ and $\widehat{\boldsymbol{k}}$. One drawback of this method is that, it relies on correct identification of a vehicle stops or turns as it is assumed that vehicle decelerates before making stops or turns. However, modern smartphones are also very sensitive and thus can capture slightest disturbance on the phone, as well as contain noisy data. Also, any user interaction with the phone during the journey is also captured by the sensors. Thus, the accuracy of coordinate alignment by this method relies on accurate estimation of a vehicle's deceleration.

Some researchers tried to simplify the alignment method by aligning just one axis to vehicle's coordinate. Nawaz et al. only aligned the Z- axis of the phone to vehicle's Z axis [15]. They defined the vehicle's coordinate system by three orthogonal axis by X_v, Y_v, and Z_v and phone's coordinate system as X_p, Y_p, and Z_p, as shown in Fig. 11. The authors showed that the turn of vehicle can be identified by the gyroscopes value along Z_p axis as gyroscope senses the angular rotation the vehicle experiences by taking turn. Plotting the gyroscope data along X_p and Y_p axis while the vehicle takes turn, the authors demonstrated that there is no change of the gyroscope values because of turns along these two axes. Thus, the authors ignored these two axis and only aligned Z_p to Z_v. From the raw acceleration data collected by phone, the authors ran a low pass filter on the output to extract the gravity components of the acceleration along the phone's local coordinate system. This gravity acceleration on phone's local coordinate is used to determine the inclination angle θ between Z_p and Z_v. θ is used to calculate the quaternion that rotates Z_p to Z_v.

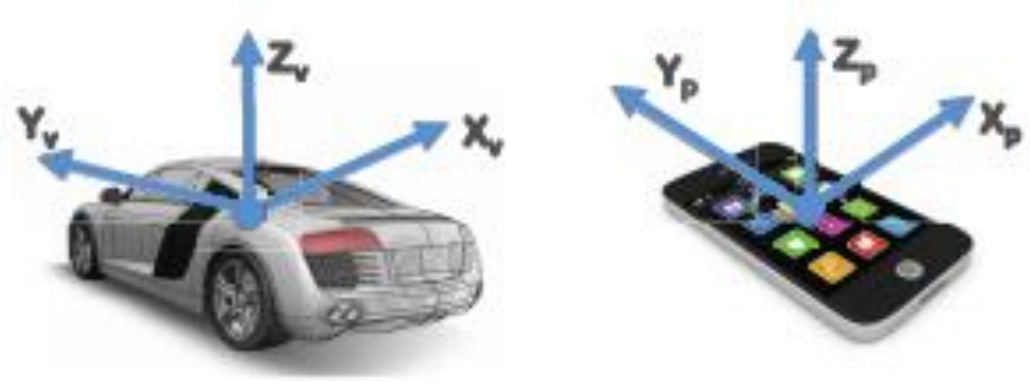

Fig. 11. Vehicle coordinate and smartphone coordinate system [15].

3.2 Literature Review on Route Matching

In recent years, several studies have utilized smartphone collected probe data to identify users' driving routes. Two studies have been found to have similarity to the research work of this dissertation. Nawaz et al. utilized smartphone collected accelerometer and gyroscope data to identify users' significant driving routes [15]. When a vehicle makes a turn, a change of angular speed is observed by the phone's gyroscope. Thus, each route has a unique signature in the form of variation in angular speed. The authors applied this concept to differentiate routes based on the unique signature of angular speed for each route. They first drove through different routes to collect each route's unique signature sensed by the gyroscope. Then, when the smartphone data from a journey that needed route identification are being collected, the authors applied Dynamic Time Warping (DTW) to calculate dissimilarity or distance between the ordered sequences of the new journey with previously collected data from known routes. A dynamic time warping algorithm calculates dissimilarity or distance between two ordered sequences (as shown in Fig. 12). It allows required compression or expansion of the ordered axis for best aligning the two sequences. The route's gyroscope data that have the least distance with the new journey's collected data are identified as the journey's route. Using dynamic time warping, the authors performed clustering of routes of collected angular speeds of a vehicle from *n* journeys made by a user. Journeys included journeys between the same origin and destination and vice-versa, the same origin different destination and vice-versa, and completely different origins-destinations. The authors computed normalized DTW distances between each pair of journeys to construct an $n \times n$ dissimilarity matrix. This dissimilarity matrix D is used with any k-medoids clustering algorithms to group similar journeys or routes together. The collected dataset included 43 journeys on 15 routes, with the shortest route being 4 km and the longest being 90 km. It was

observed that using k-medoids with k=8, the routes were organized in well-separated clusters. Similar routes were being organized within the same cluster or a cluster nearby. Fig. 13 shows the results of route clustering [15].

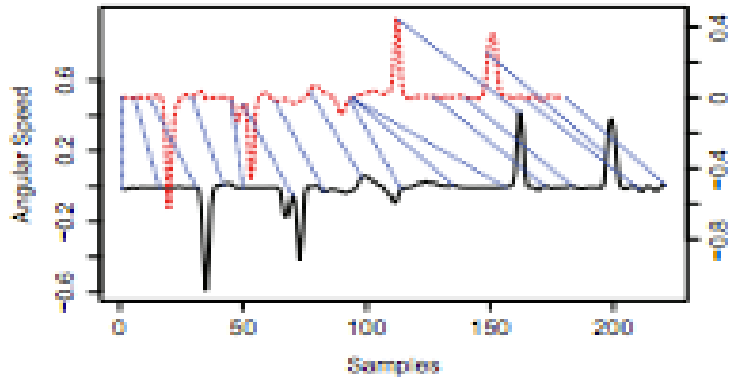

Fig. 12. Dynamic time warping between two trips [15].

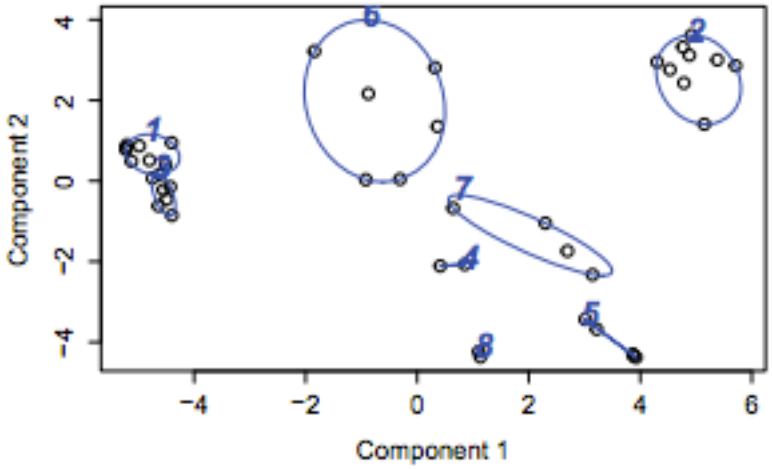

Fig. 13. Clustering of routes based on angular speed [15].

Nawaz et al. also argued that this approach would also be applicable for route identification in Manhattan style grid based network where turns at intersections haves similar signatures. To demonstrate this, the authors collected 1,000 taxi trips in New York with an average trip length of 9 miles. The dataset contained travelled distance, pickup, and drop off locations but did not contain actual routes taken by taxis. The authors used a routing algorithm on Open Street Map road network of New York to find the shortest route between pickup and drop off locations for each trip. For each shortest route road segment for each trip, they computed turning angles for all turns along the route and encoded these turns into alphabets according to some defined rules. Thus, each route is represented with a concatenated string in a particular order with each alphabet is associated with turns along the route. The common sequence of characters between two trips represents similarity between two trips based only on turns. One drawback of this approach is that, it can only identify reasonably long trips. The authors did not demonstrate whether this approach works for trips of smaller length. Another drawback is that, in grid network, multiple routes from same origin to same destination may contain a similar sequence of turns. How such routes are differentiated from each other is not discussed in the paper. Also, the authors did not show how accurately the identification of such sequences associated with turns in each trip based on smartphone probe data can be done.

Another study that bears similarity with this dissertation's research approach was done by Thiagarajan et al. [16] who matched the route of a vehicle to a map using cellular (GSM) fingerprints. A GSM fingerprint refers to a set of observed IDs of cell towers and their associated received signal strength (RSSI) value. The authors applied a two-pass algorithm to match a route to a map. In the first pass, they applied a Hidden Markov Model (HMM) to divide space into grid cells and determine the most likely sequence of traversed grid cells. In the second pass,

another HMM was built to match the traversed grid cell sequence to road segments. First, the authors divided a segment of a map into small grids. When a vehicle is within the segment, the users' smartphone provides the observed IDs of cell towers in that location and their associated RSSI values for up to 6 neighboring towers along with the associated cell tower. A training database was built with the known location from the GPS and the RSSI with the observed cell towers for the particular grids. Using the training database, the emission and transition probability was calculated and the HMM of grid sequencing was developed. Once the sequence of grids the vehicle traversed is known from the first HMM model, the authors then converted this grid sequence into a sequence of (latitude, longitude) coordinates. In the next stage, Segment Matching takes as input the sequence of points from the Smoothing and Interpolation phase, along with turns and movement hints from the phone, to determine the most likely sequence of segments traversed using a different HMM. The authors utilized the time series signals from accelerometer, compass, and gyroscope from the phone to infer information about whether the phone being tracked was moving or not, and processed the orientation sensor readings from a compass or a gyroscope to heuristically infer vehicular turns. The algorithm step is shown in Fig. 14. One drawback of this approach is that it still requires GPS information for building the training database and the HMM model.

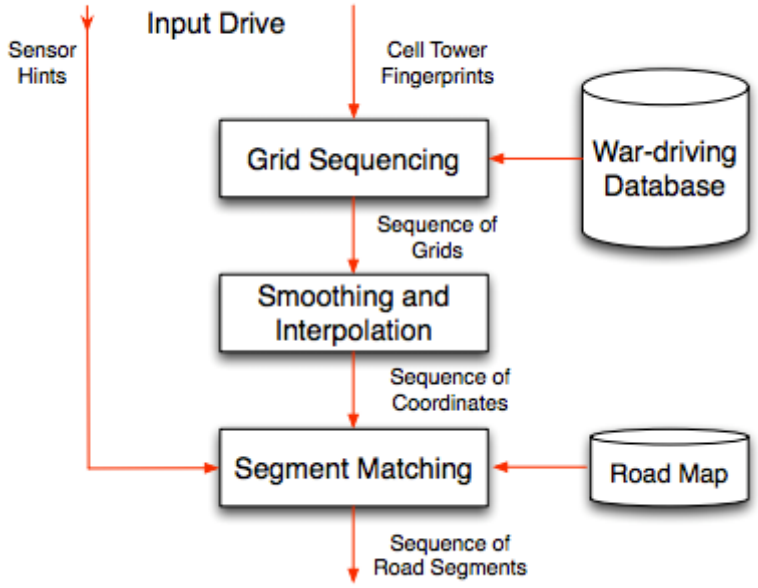

Fig. 14. Steps of Trajectory Mapping Algorithm by Thiagarajan et al. [16].

Another study related to this dissertation's work was done by Aly et al. [17]. The authors employed accelerometer and gyroscope sensors to detect a car's ambient road semantics, such as speed bumps, tunnels, bridges, stop signs, etc. The sequence of road semantics found from the car's trip is then applied in a map matching process and infer the car's current road location. Map matching is done using Hidden Markov Model (HMM) algorithm that takes the car's road semantics information, surrounding information, and the noise of the input data into account to enhance the accuracy of the estimated road segments and evaluate the candidate road segments. Finally, a Viterbi Algorithm is employed to dynamically and efficiently identify the most probable road segment. One drawback of that study is that it relies on variation in road semantics for map matching. For some routes with small length, there may not be very much variation in

semantics. For example, at an urban grid network, multiple routes may have similar sequence of semantics. In such cases, this approach may not work.

As GPS has accuracy issues at places that have disrupting GPS satellite connectivity due to high-rise structures, several studies have been devoted towards matching error-prone GPS readings of smartphones from moving vehicles to actual road segments. For example, Thiagarajan et al. explored the GPS and sensor readings from users in Chicago downtown and its vicinity riding on buses to match the bus trajectory with scheduled bus routes [18]. The data set contained both users riding buses and users stationary on the street. Using smartphone inertial sensor data collected from the users, the authors first developed an activity classification system based on acceleration values that could identify whether a user was walking or riding a bus. Once the users riding the bus were identified, the authors explored the GPS reading from those users to identify the trajectory of the bus. The authors collected bus shapes from publicly available data from the Chicago Transit Authority (CTA). Each bus shape consisted of a sequence of bus stops specified as latitude-longitude coordinates. Once the GPS reading of a user riding on bus was found, using the geospatial index lookup using the user GPS reading, the set of candidate bus routes were narrowed down to match. Then, they applied a least square minimization algorithm to each of the candidate routes that finds the bus shape that is the closest fit to the user's GPS sequences.

Goh et al. developed an online map-matching algorithm based on a Hidden Markov Model (HMM) to accurately map the GPS trajectories to the road network in real time [19]. A trajectory is a sequence of data points collected by a vehicle. Each trajectory point is specified by its longitude, latitude, speed, and a time stamp. For every trajectory point, the authors first identified a set of candidate road segments from which the data were most likely collected. Each

of these candidates is then represented as a hidden state in the Markov chain and an emission probability for each candidate is calculated based on the likelihood of observing the GPS point conditional on the candidate segment being the true match. The transition probability for every pair of adjacent hidden state in the chain was calculated such that the probability of the latter is dependent only on the former. The maximum likelihood path over the Markov chain that has the highest joint emission and transmission probabilities is identified as the vehicle trajectory.

Several authors developed mathematical methods for map matching using GPS data collected from a user's smart phone. For example, Bierlaire et al. developed a probabilistic map matching algorithm that employs both geographical and temporal information in GPS data to measure the likelihood that the data have been generated along a given path [20]. A probabilistic measurement model is employed to calculate the probability that a GPS recording device would have generated a sequence of measurements while following a given path. Basing on the work of Bierlaire et al., Chen et al. developed a probabilistic measurement model that employs GPS location data only [21]. The model calculates the likelihood of observing the smartphone data in a multimodal transport network. The probabilistic method results the output as a set of candidate true paths and gives the probability of each path being the actual one.

Using time stamped GPS data, Brakatsoulas et al. presented three map-matching algorithms that mapped trajectory on a road network by matching geometries [22]. The first algorithm employs a portion-by-portion map matching by projecting the trajectory on a map by matching distance and angles between the curves. The second and third approach aims to match the trajectory with candidate road network by comparing similarity measures: the Fréchet distance and the weak Fréchet distance, respectively, which provides the path that is the optimal distance to the trajectory.

Krumm et al. developed a map matching algorithm in which GPS coordinates are taken as input and roads are matched based not only on the location measurements, but also on the time stamps of the measurements [23]. The time constraint ensures that the sequence of matched roads can be reached from each other in the time intervals computed from the measurements' time stamps. Using the GPS input and candidate paths, a path that matches the location measurements and is reachable within the provided time constraints is chosen as the vehicle trajectory. The error in GPS measurements and the error in the estimates of path traversal times are optimized with a hidden Markov model (HMM). The HMM is based on a probability distributions representing the two errors and the distributions are based on a large database of trip information with GPS coordinates of drivers. The results indicate that the map-matching algorithm is accurate in matching measured locations to roads.

Several studies also focused on finding walking trajectory of pedestrians using a periodic GPS signal and a smartphone's inertial sensors. Ionut et al. developed an escort system where mobile phone sensors and an audio beacon system between different individuals are used to provide route guidance to an individual to reach a destination [24]. The system measures the accelerometer and compass readings from user-carried mobile phones to measure a user's approximate displacement from the initial position and to develop a walking trail. An audio beacon system is developed where a user's phone periodically sends audio signals to another individual's phone and also listens for signals from those devices. This audio signal between phones is uploaded to an audio server. The strength of the signal between phones is used as a guide to further correct the user's walking trail that was developed using the inertial sensors.

Constandache et al. developed a localization system that develops a unique signature of compass and accelerometer data for walking trails [25]. The mobile phone's accelerometer and

electronic compass is used to measure the walking speed and orientation of the mobile user. These readings are used to produce a directional trail that can be matched against walkable path segments within a local area map. Infrequent GPS readings are employed to periodically recalibrate the user's location. The results show that this is a promising approach to identify user's walking trail with low error.

Uddin et al. employed acceleration readings from a user's smartphone to identify walking direction. Using principle component analysis, they identified the major axis along which the acceleration has much variation [26].

Several other studies have focused on identifying a user's location using a phone's inertial sensor data to reduce dependence on continuous GPS connectivity. For example, Paek et al. employed accelerometer readings from phones to identify a user's movement from an initial position [27]. A Bluetooth connection is then used between nearby devices to get a GPS location update rather than turning on its own GPS device. This study just focused on periodic localization of a user's device not the trajectory identification. Liu et al. employed the signal strength of Wi-Fi signals in smartphones and accelerometer and compass data to identify a user's location inside a building [28]. The user's direction of movement and distance walked is estimated using compass and accelerometer data. Using Wi-Fi signal strength, a training database of the building's room layout is created and a map matching is done based on observed signal strength and the determined movement direction.

3.3 Literature Review on Activity Detection

Several studies have focused on activity detection of a user by exploring smartphone inertial sensor data. For example, Thiagarajan et al. developed a system that detects a user's

state, such as walking, in a stationary vehicle, or in a moving vehicle, based on accelerometer input [16]. First, the authors detected a user's motion by sampling the accelerometer at 1 Hz, and continuously computing an exponentially weighted mean and standard deviation of the X,Y, and Z accelerometer data. If a phone is in a stationary position, an incoming sample lies inside of three standard deviations on any axis. If the phone is in motion the incoming accelerometer lies outside of the three standard deviation range. In order to detect walking motion, the authors computed a discrete Fourier transform (DFT) for a sliding window of the raw accelerometer values. If the magnitude of DFT coefficients are at (1-3 Hz) range, which is common to walking, then the motion is detected as walking motion. In order to find vehicular motion, first, the walking motion data are filtered out using the walking motion detection technique. Then, using the training data, two distributions of acceleration samples, the moving vehicle and stationary vehicle, is modeled by Laplace distribution and probability density function is estimated. Then, Bayes theorem is applied to compute the probability of a sample coming from a moving vehicle distribution.

Eren et al. applied Bayes classification on accelerometer and gyroscope data collected from drivers to identify the driving behavior of the drivers [29]. They classified driving behavior as safe driving and risky driving. Based on a training data set, the authors calculated the prior probability of safe and risky driving. Then, using accelerometer and gyroscope observations from driving events, the probability of each observation coming from safe driving or risky driving is calculated using a Bayesian interference. Their approach was able to correctly detect the safe driving and risky deriving instances 93.3% times. Similar work was done by Johnson et al. who explored accelerometer, gyroscope, and device's Euler rotation angle values to differentiate aggressive versus non-aggressive driving [30]. The authors first focused on

developing template of eight maneuvers or events: left turn, right turn, hard left, hard right, swerve left, swerve right, U-turn, and hard U-turn. Using a training data set of multiple trips of each maneuvers, the authors first employed the K-Nearest Neighbors (k-NN) classification method with k = 3 to determine the type of event. In order to determine whether or not a driving event is non-aggressive or aggressive, they applied a DTW algorithm with the sequence of smartphone sensor data from unknown events to each of the events template. The DTW algorithm finds the optimal path between each template with the unknown event's sequence. The template that has the lowest cost of the aligned path is identified as the maneuver type of the unknown event.

Hemminki et al. explored smartphone accelerometer data to detect the transportation mode of a user [31]. The authors extracted a series of features from a horizontal accelerometer that captures characteristics of acceleration and breaking patterns for different transportation modes. They extracted statistical features (e.g., mean, variance, and kurtosis), time-domain metrics (e.g., double integral, auto-correlation, and zero crossings), frequency-domain metrics (e.g., energy, six first FFT components, entropy, and the sum of FFT coefficients) and peak-based features such as volume, intensity, length, kurtosis, skewness, etc. Frame-based features were found to be effective to capture characteristics of high-frequency motion caused by a user's physical movement during pedestrian activity, or during motorized periods, from a vehicle's engine and contact between its wheels and surface, etc. Peak-based features characterized acceleration and deceleration periods. After the features were extracted, the authors employed a discrete Hidden Markov Model (HMM) and a variant of AdaBoost for transportation mode detection.

Chu et al. designed a Driver Detection System (DDS) that relies on smartphone sensors to recognize micro-activities in humans, that is different for the driver and the passenger [32]. Using the accelerometer and gyroscope data from the driver's and passenger's smartphone, the authors developed a set of sensor signatures for movements that differ for a driver and for a passenger. The considered movements included the vehicle entry swing in which the user's lower body movement is different based on whether the driver is entering to the left or to the right of the vehicle; pedal press: that determines if the user is putting pressure on the brake or gas pedal given that the phone is in kept in the user's inner leg pocket; seat-belt, in which the user's turning direction toward left or right will depend based on the seating arrangement inside the car. They also employ microphone audio data from the phone in order to differentiate whether the user is sitting in front or rear of the vehicle. The authors employed support vector machine (SVM) to classify the input data for signature for each different activity. The evaluation for their techniques revealed that the accuracy of their developed techniques for each of the signature movement detection varied from 88% to 95%.

CHAPTER 4
MACHINE LEARNING METHODS

In this chapter, the detailed machine learning methods that will be used later in the dissertation, are described.

4.1 Application of Dynamic Time Warping (DTW) Technique

Dynamic time warping (DTW) is a well-known technique to find an optimal alignment between two given time series sequences while allowing warping of the sequences in a nonlinear fashion to match each other [33]. This algorithm matches one point of a sequence to one or more point of another sequence [34, 35]. DTW has been widely used for speech recognition to compare different speech patterns. However, recently it has been employed in various other purposes, such as shape recognition, gesture recognition, map matching, etc.[15, 36].

It is intended in this research to apply a DTW algorithm to find the distance between the time series sensor data of an unknown maneuver with a known maneuver. A signature dataset consisting of sensor data from a known maneuver of a vehicle is considered as the template. A DTW technique will be applied to match the sequence of the dataset from unknown maneuvers with a time series sensor from the template maneuver.

Suppose, A= $\{a_1, a_2, a_3 \dots \dots a_n\}$ is a time series of an inertial sensor data along a particular axis which recorded a specific force that is experienced by the smartphone when the vehicle completed a known maneuver at an intersection. B= $\{b_1, b_2, b_3 \dots \dots b_m\}$ is a time series of the same force along the same axis experienced by another smartphone in an unknown maneuver at an intersection. For eliminating the complexity of coordinate conversion, let's suppose that both the phones during both the maneuvers were at the same orientation and thus

the major axis that they experienced the maximum variation of force along, was same. Let's say that axis was Y-Axis of phone's local coordinate. If i =1, 2,…, s,….,n and j = 1, 2,….,s,…,m are the indexes of the two time series, A and B, respectively, then the optimal path that minimizes the total distance between the two series is

$$P = p_1, p_2, \dots, p_s, \dots, p_n , \quad (1)$$

where

$p_s=(i_s, j_s)$ is the warping function.

The dissimilarity or distance between the two sequences is as follows:

$$D_{DTW}(A,B) = \sum_{s=1}^{k} d(p_s)Z_s \Big/ \sum_{s=1}^{k} Z_s , \quad (2)$$

where

$d(p_s)$ is the distance between i_s and j_s,

$Z_s > 0$ is the weighting coefficient.

P is found by solving the following problem with certain constraints:

$$\min(D(A,B)). \quad (3)$$

The constraints are as follows:

$$i_{s-1} \leq i_s , \quad (4)$$

$$j_{s-1} \leq j_s , \quad (5)$$

$$i_s - i_{s-1} \leq 1, \quad (6)$$

$$j_s - j_{s-1} \leq 1, \quad (7)$$

$$i_1 = 1,\ i_k = n, \quad (8)$$

$$j_1 = 1,\ j_k = m, \quad (9)$$

$$|i_s - j_s| \leq r, \tag{10}$$

where $r > 0$ is the window length.

Equations 3 and 4 are the monotonicity constraints and guarantee that the path proceeds forward in time index. Equations 5 and 6 are the continuity constraint and guarantee that the path just moves one step in the time index. Equations 7 and 8 are the boundary constraints and ensures that the aligned path starts from bottom left and end at top right of the grid. Equation 9 ensures that the path does not wander very far from the diagonal [35].

Solving Equation 2 gives us the optimal path and knowing the optimal path, the distance between the two sequences can be calculated using equation 1. If A and B are the results of a similar maneuver, then the distance will be small, if different, then the distance is expected to be large. This method is applied to identify the three maneuver types.

4.2 K-medoids Clustering

K-medoids clustering is applied to time series data of gyroscope readings from different trip to cluster similar type of maneuver together. K-mediods clustering steps are as follows [37]:

1) Assume one point from each cluster as a representative object of that cluster.

2) Find the Euclidean distance of each object from this representative object.

3) The objects with minimum distance from the cluster representative are selected as the members of each cluster.

4) Estimate the mean of the minimum distances for each point from its cluster representative object.

5) Select 2 new objects as representative objects and repeat steps 2-4.

6) Swapping cost is calculated by subtracting previous mean distance from new mean distance.

7) If swapping cost is negative then new medoids are the new representative objects and go to step 5.

8) If cost is more than 0, discard points and select new representatives in step 5 keeping original points.

9) These steps will be repeated until cluster variation does not change.

CHAPTER 5
DATA DESCRIPTION AND PREPOSSESSING

This section details the description of the data, the data collection system, the prepossessing requirements, and the data preprocessing and validation techniques applied.

5.1 Data Description

The data were collected using the GoGreen app. GoGreen is a custom Android application developed by the Transportation Research Institute at Old Dominion University. The data collection system consists of the GoGreen app, GPS satellite, On Board Diagnostic (OBD) device, and a database. The app was developed in Android smartphone environment. In order to collect data using this app during a trip, the user has to start the app on a smartphone. The app collects inertial sensor data, such as linear acceleration, gravity acceleration, gyroscope readings, etc. The user can choose to turn the GPS of the smartphone on in order to get latitude and longitude information of the vehicle or can choose to keep the GPS off. Similarly, the OBD device can be turned on and off. If on, the OBD device measures a vehicle's speed, throttle position, fuel level, and RPM [mph]. The OBD device transmits data via Bluetooth to the smartphone.

Once the application is running with GPS and OBD on, it will collect the following readings:

1. Video
2. Accelerometer data: Time stamped acceleration in a local X, Y, and Z axis [meter per second^2, ms^{-2}]

3. Gyroscope: Time stamped angular speed in a local X, Y, and Z axis [radian per second, rad/s]
4. Compass data: Time stamped magnetometer data in a local X, Y, and Z axis [micro tesla, μT]
5. OBD data: Speed, throttle position, fuel level, RPM [mph]
6. Orientation data: The orientation of the phone [degrees]
7. GPS data: Altitude, latitude, longitude, speed [m/s], time stamp
8. Linear acceleration data: Time stamped data extracted from the accelerometer data in a local X, Y, and Z axis [ms^{-2}]
9. Gravity data: Time stamped data extracted from the accelerometer data in a local X, Y, and Z axis [ms^{-2}]

The phone collects that data at varying rates. The raw data are interpolated to get 10 data points per second (data rate 10 Hz). Once the user stops the app, it will display the time the trip took, fuel consumption of the vehicle, and average speed of the vehicle. The data collected are saved to the phone's local database. It will also prompt the user to upload the data into the database. Once uploaded, the data can be acquired using a code written in Java for data acquisition. Fig. 15 shows the data collection system. The snapshot of GoGreen app and the view of data in database are shown in Fig. 16 and Fig. 17. All the inertial sensor data are collected based on smart phone's local coordinate system (Fig. 18). The phone's local coordinate frame is fixed in the center of the phone. The X axis of the local coordinate system is along the width of the phone and considered positive towards the right side. The Y axis is along the length of the phone and positive towards the top of the phone. The Z axis is perpendicular to the screen of the phone and positive towards outward direction.

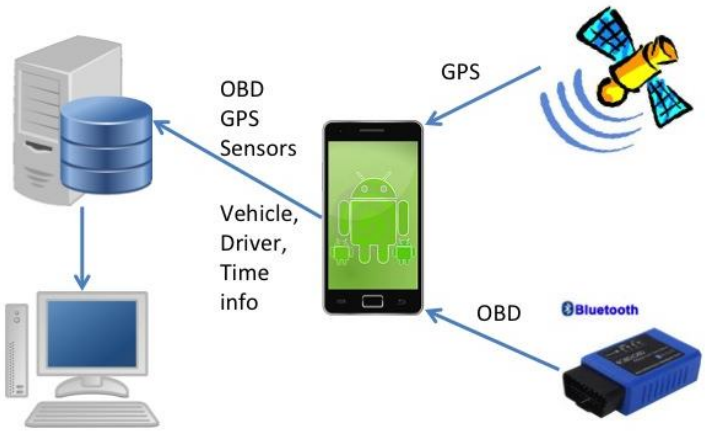

Fig. 15. Data collection System.

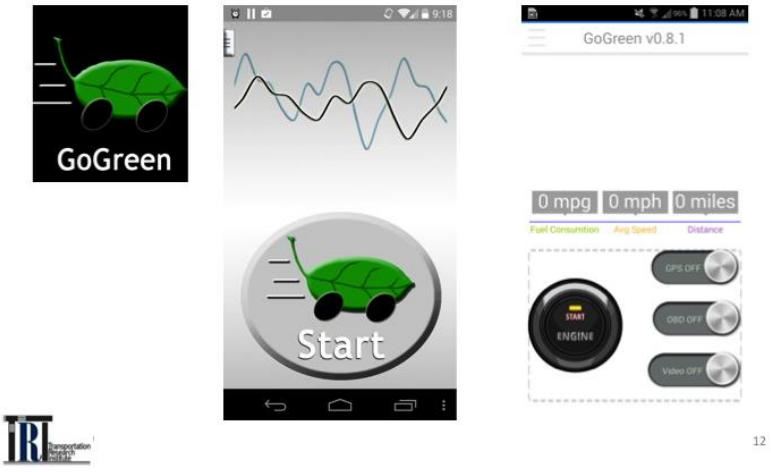

Fig. 16. Snapshot of GoGreen app.

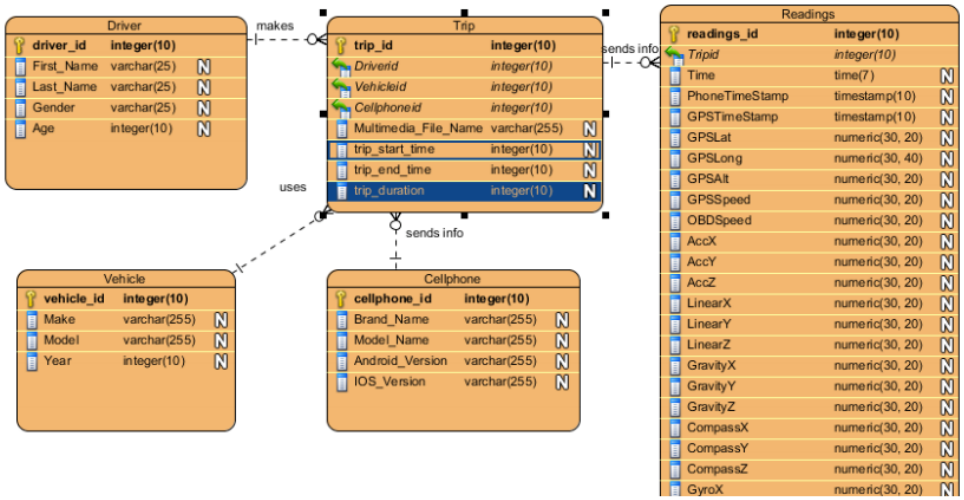

Fig. 17. The database of stored data.

The dataset for preliminary analysis was collected from two separate intersections in Norfolk and one intersection in Virginia Beach. The Norfolk intersections are the Boush-Brambleton intersection (mentioned as Intersection 1 in this dissertation) and Hampton Blvd-38th street intersection (mentioned as Intersection 2 in this dissertation). The Virginia Beach intersection is the Independence Blvd-Virginia Beach intersection (mentioned as Intersection 3 in this dissertation). The locations are shown in Fig. 19, Fig. 20, and Fig. 21. The dataset collected from Intersection 1 and Intersection 2 contained vehicle information for the trips made by 7 separate drivers with 10 and 12 different type of cellphones, individually. The dataset collected from Intersection 3 contained vehicle information for the trips made by 5 separate drivers with 7 different types of cellphones. The data summary is presented in TABLE 1. The smartphone is kept in arbitrary orientation inside the vehicle. The data are gathered from circular area with a radius 90 meter measured from the center of the intersection.

TABLE 1 Summary of Data collected from 3 different intersections

Location	#Drivers	#Cell phones	Left turn	Right turn	Through
Intersection 1	7	10	145	90	35
Intersection 2	7	12	124	133	157
Intersection 3	5	7	9	3	56

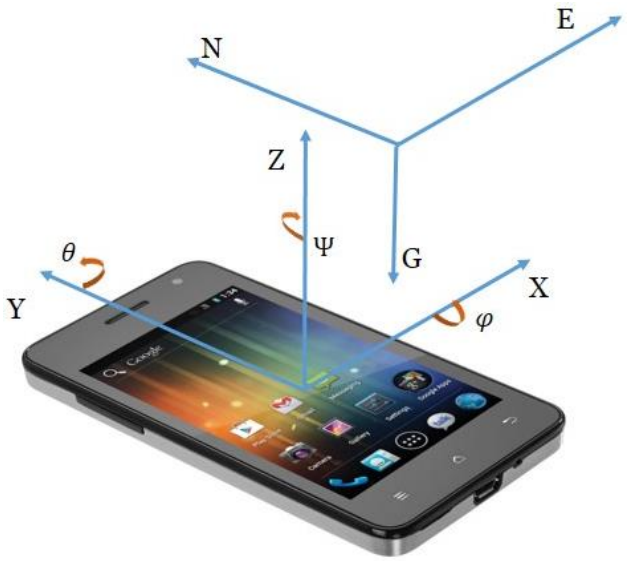

Fig. 18. Smartphone's local coordinate system, global coordinate system and orientation angles made by the phone.

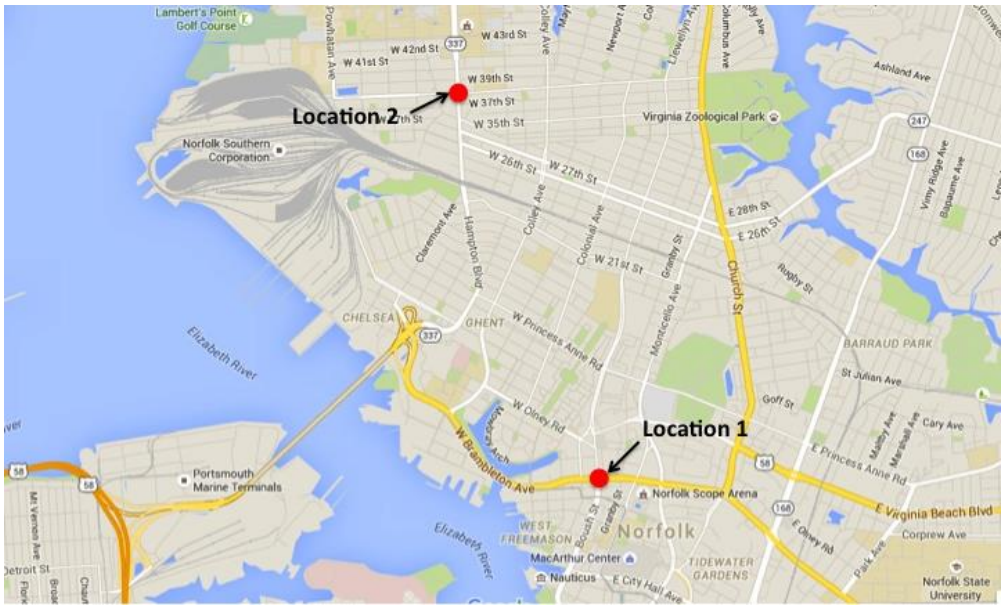

Fig. 19. Map of location of data collection for analysis showing Intersection 1 and 2.

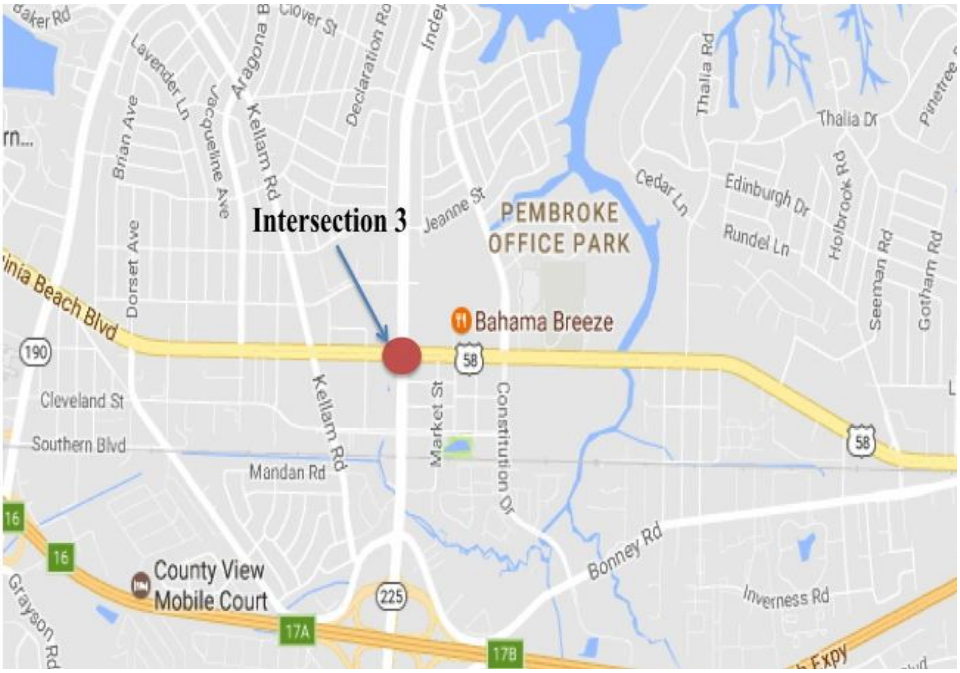

Fig. 20. Map of location of data collection for analysis showing Intersection 3.

Fig. 21. Intersections for data collection for preliminary analysis. Top Left: Location 1: Boush-Brambleton intersection. Top right: Location 2: Hampton-38th Street intersection. Bottom: Intersection 3: Independence Blvd-Virginia Beach Blvd.

5.2 Data Preprocessing Requirements

The smartphone can remain in any arbitrary orientation inside a vehicle. The smartphone provides the sensing data in the X, Y, and Z axes of its local coordinate. Based on the orientation of the phone and the direction of movement of the vehicle, change in forces is experienced along different axes. For example, the gyro sensor detects the change in angular speed of the phone.

The gyroscope is calibrated such that when laid in horizontal plane, the gyroscope reading is zero. Any change in the orientation of the phone is detected by the gyroscope and change of angular speed corresponding to phone's local axis is provided in a XYZ coordinate in radian/seconds. When a vehicle undergoes a turning maneuver, the gyroscope sensor inside the smartphone detects the turning motion and a change is expected to be observed in the reading along the Z axis of phone's local coordinate provided that the phone is laid on horizontal plane. However, as the phone can remain in an arbitrary position, the variation of the gyroscope reading due to turning movement can be observed at any axis. Such a scenario is described in Fig. 22. Fig. 22(a) and Fig. 22(d) show two different orientations of phone inside a vehicle. Plots of raw three-Axis gyroscope readings during two left turn trips for these two orientations are presented in Fig. 22(b) and Fig. 22(e), respectively. The variation in raw gyroscope readings for left turn was observed along Y axis of phone's local coordinate (Fig. 22 (b)) for orientation 1, while it was observed along Z axis (Fig. 22 (e)) for orientation 2, highlighted by the brown box in the figures. As the goal is to find unique signature of vehicle maneuvers for all phones regardless of orientation, orientation correction on raw sensor data is applied. This will align a phone's local coordinate system to a global coordinate system. A global coordinate is considered as North, East, and Downward positive. Thus a phone's positive X, Y, and Z axis will align to East, North and downward direction, respectively. Fig. 18 shows the phone's local coordinate system and global coordinate system.

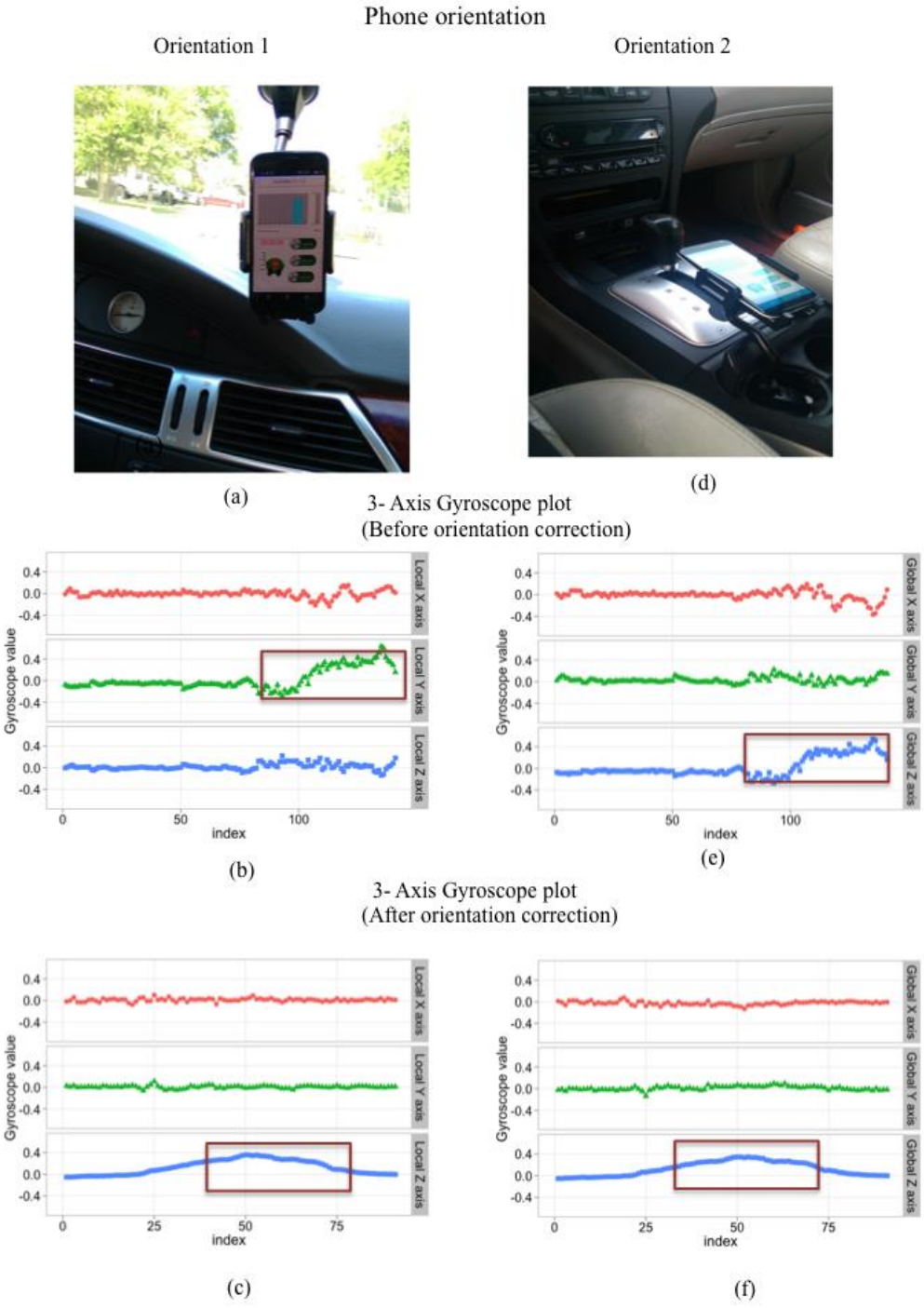

Fig. 22. Three-axis gyroscope plots for before and after orientation correction during two left turn trips while phone kept in two different orientations.

Finally, smartphone sensors are very sensitive and capture any movements experienced by a phone; thus are prone to noise in the data. During the data collection, the phones are kept in a stationary position. However, the phone might still experience some jerks or vibrations along directions other than the direction of vehicle's movement. It is necessary to filter out such instances so that the forces experienced by the phone are only due to vehicular movement. Thus, it is required to apply appropriate techniques for data filtering to clear out such noise.

5.3 Data Preprocessing

In this section, the data pre-processing techniques used for noise removal and orientation correction of the smartphone sensor data are described.

5.3.1 Noise Removal

Two techniques have been employed to reduce noises in the sensor data. For the first technique, the gravity acceleration extracted using 3-axis accelerometers has been utilized to filter out the sensor data points within the time series. Since a smartphone can remain in any arbitrary orientation inside a vehicle, gravity acceleration measured by the phone is divided among the X, Y, and Z axes along a phone's local coordinate system. The total gravity acceleration, which is the square root of gravity acceleration values measured in each direction, is expected to be 9.8 ms^{-2}. Due to potholes on the road surface, the vehicle may experience sudden jerky movement along the direction perpendicular to the horizontal plane. As a result, the phone will also experience sudden movement along the vertical direction, which, in turn, can cause a sudden dip or spike in total gravity acceleration values. This understanding is employed to filter out such instances in the time series by taking only the instances in which the square root

of gravity acceleration values measured in each direction is higher than 9.6 ms^{-2} and lower than 9.98 ms^{-2}. The following filter is employed to each instance of the time series:

$$9.6\, ms^{-2} \leq \sqrt{g_X^2 + g_Y^2 + g_Z^2} \leq 9.98\, ms^{-2}, \quad (11)$$

where

g_X Gravity acceleration value measured by phone along a phone's local X axis,

g_Y Gravity acceleration value measured by phone along a phone's local Y axis,

g_Z Gravity acceleration value measured by phone along a phone's local Z axis.

The threshold value of g is taken to be 9.6 ms^{-2} instead of 9.8 ms^{-2}, as the precision of smartphone accelerometer can vary up to $0.018g$, resulting the value of g measured to be within the range from 9.6 to 9.98 ms^{-2} [38]. It was found that only 2 trips out of 775 (0.25%) trips had such instances where the gravity value was out of range.

Next, a median filter is applied on 1-second window of sensor data. A median filter is chosen as the data smoothening technique as it can effectively reduce noise without losing important features of the data. Each data point is replaced by the median value of consecutive 10 data points. The equation for the median filter is as follows:

$$x_i = median\,\{x_i, x_{i+1}, \ldots, x_{i+9}\}, \quad (12)$$

where

i index of the data point.

The Z component of gyroscope readings plot before and after noise removal is shown in Fig. 23.

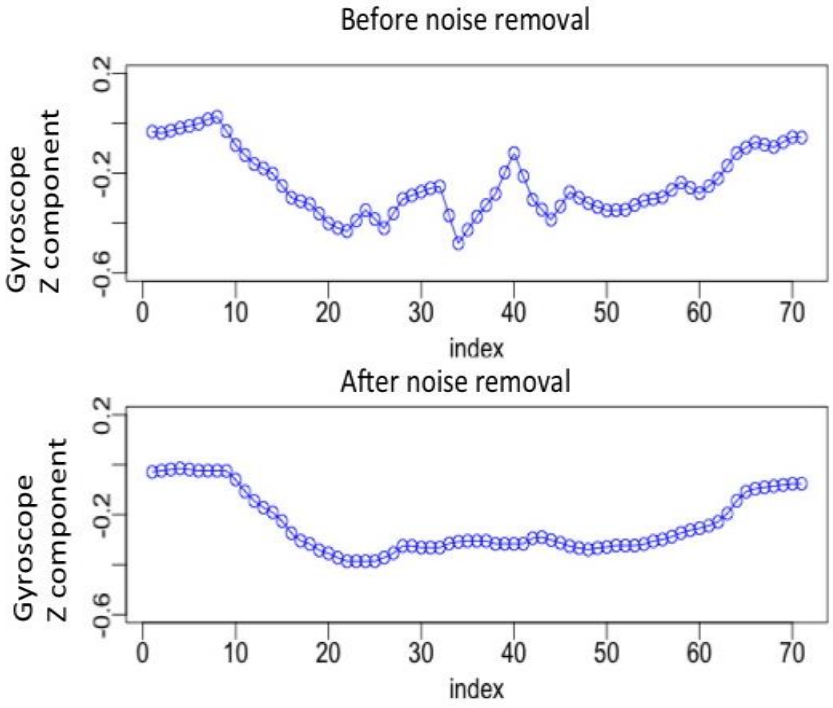

Fig. 23. Z component of gyroscope readings plot before and after noise removal.

5.3.2 Orientation Correction

A two-step orientation correction method is applied on the smartphone collected raw data to transform the data to a global coordinate system. The first step is to identify the orientation of the phone with respect to its global coordinate; the second step includes the transformation of the raw sensor data from a local to a global coordinate.

For the first step, the technique developed by Ozyagcilar is employed [39]. Let, φ, θ and ψ are the pitch, roll, and yaw angles of a phone respectively in Fig. 18. φ, θ, and ψ represent the rotation along X, Y and Z axis of phone's local coordinate respectively. ψ is also the angle made by the phone relative to North.

Let, the gravity component, *g* measured by the phone at its current location is

$$\begin{bmatrix} g_X \\ g_Y \\ g_Z \end{bmatrix} \tag{13}$$

and the magnetometer signal, *m* at the current orientation and location is

$$\begin{bmatrix} m_X \\ m_Y \\ m_Z \end{bmatrix}. \tag{14}$$

Then, according to Ozyagcilar [39],

$$\text{Pitch}(\varphi) = \tan^{-1}\left(\frac{g_Y}{g_Z}\right), \tag{15}$$

$$\text{Roll}\ (\theta) = \tan^{-1}\left(\frac{-g_X}{g_Y \sin\varphi + g_Z \cos\varphi}\right), \tag{16}$$

$$Yaw\,(\psi) = tan^{-1}\left(\frac{m_X \cos\theta + m_Y \sin\theta \sin\varphi + m_Z \sin\theta \cos\varphi}{m_Z \sin\varphi - m_Y \cos\varphi}\right). \tag{17}$$

Further restriction on the angles is provided to assure that for one orientation of the phone the pitch, roll, and yaw has only one unique solution. The pitch and yaw angles are restricted to have values from -180 to 180 degree, while roll angles are restricted to have values from -90 to 90 degrees.

The second step of the orientation correction technique involves applying transformation operation on raw data to transform it from local to global coordinate system. After finding the orientation angles, a quaternion transformation equation is developed and applied on the smartphone-collected data. It can also be done by applying rotation matrix to rotate a phone's local coordinates to global coordinates, where, a local X, Y, and Z axis will be rotated by $\varphi, \theta,$ and γ amount, respectively, to align with a global X, Y, and Z axis. The rotation matrixes will be as follows:

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{bmatrix}, \tag{18}$$

$$R_Y = \begin{bmatrix} cos\,\theta & 0 & -sin\,\theta \\ 0 & 1 & 0 \\ sin\,\theta & 0 & cos\,\theta \end{bmatrix}, \tag{19}$$

$$R_Z = \begin{bmatrix} cos\,\psi & sin\,\psi & 0 \\ -sin\,\psi & cos\,\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{20}$$

Let's suppose, x̃ỹz̃ is a new coordinate system whose orientation w.r.t xyz is a rotation by ψ along z − axis. The coordinates of point r in new system

$$\begin{pmatrix} \tilde{r_1} \\ \tilde{r_2} \\ \tilde{r_3} \end{pmatrix} = R_Z(\psi) \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} = \begin{bmatrix} cos\,\psi & sin\,\psi & 0 \\ -sin\,\psi & cos\,\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}. \tag{21}$$

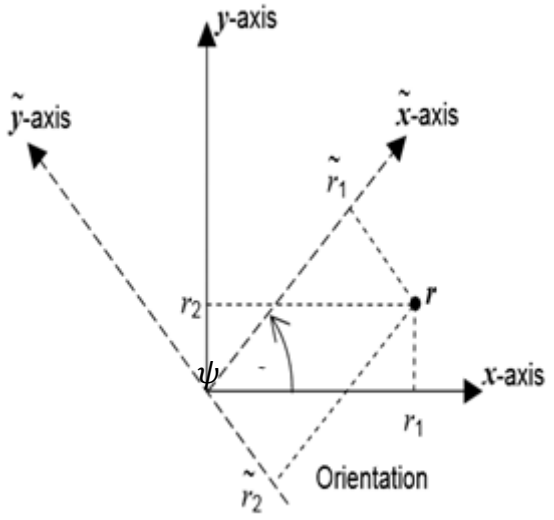

Fig. 24. New coordinate system.

$\tilde{x}\tilde{y}\tilde{z}$ is a new coordinate system whose orientation w.r.t xyz is a rotation by ψ along z axis, φ along x axis and θ along y axis (Fig. 24).

The coordinates of point r in new system

$$\begin{pmatrix}\tilde{r_1}\\ \tilde{r_2}\\ \tilde{r_3}\end{pmatrix} = R_z(\psi)\, R_y(\theta)\, R_x(\varphi) \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \tag{22}$$

$$= \begin{bmatrix} cos\,\psi & sin\,\psi & 0 \\ -sin\,\psi & cos\,\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\,\theta & 0 & -sin\,\theta \\ 0 & 1 & 0 \\ sin\,\theta & 0 & cos\,\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\,\varphi & sin\,\varphi \\ 0 & -sin\,\varphi & cos\,\varphi \end{bmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}. \tag{23}$$

$$R_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\,\varphi & sin\,\varphi \\ 0 & -sin\,\varphi & cos\,\varphi \end{bmatrix} \tag{24}$$

can be represented as

$$\text{Quaternion, } q_x(\varphi) = \left(cos\frac{\varphi}{2}, sin\frac{\varphi}{2}x\right) \tag{25}$$

$$= \left(cos\frac{\varphi}{2} + \frac{(i.1+j.0+k.0)}{\sqrt{1+0+0}} sin\frac{\varphi}{2}\right) \tag{26}$$

$$= \left(cos\frac{\varphi}{2} + i\, sin\frac{\varphi}{2}\right). \tag{27}$$

$$\text{Similarly, } q_y(\theta) = \left(cos\frac{\theta}{2} + j\, sin\frac{\theta}{2}\right) \tag{28}$$

$$q_z(\psi) = \left(cos\frac{\psi}{2} + k\, sin\frac{\psi}{2}\right). \tag{29}$$

Rotations along z-y-x axis can be represented by

$$q = \left(cos\frac{\psi}{2} + k\, sin\frac{\psi}{2}\right)\left(cos\frac{\theta}{2} + j\, sin\frac{\theta}{2}\right)\left(cos\frac{\varphi}{2} + i\, sin\frac{\varphi}{2}\right) \quad (30)$$

$$= \left(cos\frac{\psi}{2}cos\frac{\theta}{2} + k\ sin\frac{\psi}{2}cos\frac{\theta}{2} + j\ cos\frac{\psi}{2}sin\frac{\theta}{2} - i\, sin\frac{\psi}{2}sin\frac{\theta}{2}\right)\left(cos\frac{\varphi}{2} + i\, sin\frac{\varphi}{2}\right) \quad (31)$$

$$=\left(cos\frac{\psi}{2}cos\frac{\theta}{2}cos\frac{\varphi}{2} + cos\frac{\psi}{2}cos\frac{\theta}{2}sin\frac{\varphi}{2} + k\, sin\frac{\psi}{2}cos\frac{\theta}{2}\ cos\frac{\varphi}{2} + j\, sin\frac{\psi}{2}cos\frac{\theta}{2}sin\frac{\varphi}{2} + j\, cos\frac{\psi}{2}sin\frac{\theta}{2}cos\frac{\varphi}{2} - k\ cos\frac{\psi}{2}\ sin\frac{\theta}{2}sin\frac{\varphi}{2} - i\, sin\frac{\psi}{2}sin\frac{\theta}{2}cos\frac{\varphi}{2} + \ sin\frac{\psi}{2}sin\frac{\theta}{2}sin\frac{\varphi}{2}\right) \quad (32)$$

$$= \left(cos\frac{\theta}{2}cos\frac{\varphi}{2}cos\frac{\psi}{2} + sin\frac{\varphi}{2}sin\frac{\theta}{2}sin\frac{\psi}{2}\right) + \left(sin\frac{\varphi}{2}cos\frac{\theta}{2}cos\frac{\psi}{2} - cos\frac{\varphi}{2}\ sin\frac{\theta}{2}sin\frac{\psi}{2}\right)i$$

$$+ \left(cos\frac{\varphi}{2}sin\frac{\theta}{2}cos\frac{\psi}{2} + sin\frac{\varphi}{2}cos\frac{\theta}{2}\ sin\frac{\psi}{2}\right)j + \left(cos\frac{\varphi}{2}cos\frac{\theta}{2}sin\frac{\psi}{2} - \ sin\frac{\varphi}{2}sin\frac{\theta}{2}cos\frac{\psi}{2}\right)k. \quad (33)$$

Thus, Equation 19 can be written as quaternion

$$\text{q} = \begin{bmatrix} cos\left(\frac{\varphi}{2}\right)cos\left(\frac{\theta}{2}\right)cos\left(\frac{\psi}{2}\right) + sin\left(\frac{\varphi}{2}\right)sin\left(\frac{\theta}{2}\right)sin(\frac{\psi}{2}) \\ sin\left(\frac{\varphi}{2}\right)cos\left(\frac{\theta}{2}\right)cos\left(\frac{\psi}{2}\right) - cos\left(\frac{\varphi}{2}\right)sin\left(\frac{\theta}{2}\right)sin(\frac{\psi}{2}) \\ cos\left(\frac{\varphi}{2}\right)sin\left(\frac{\theta}{2}\right)cos\left(\frac{\psi}{2}\right) + sin\left(\frac{\varphi}{2}\right)cos\left(\frac{\theta}{2}\right)sin(\frac{\psi}{2}) \\ cos\left(\frac{\varphi}{2}\right)cos\left(\frac{\theta}{2}\right)sin\left(\frac{\psi}{2}\right) - sin\left(\frac{\varphi}{2}\right)sin\left(\frac{\theta}{2}\right)cos(\frac{\psi}{2}) \end{bmatrix}. \quad (34)$$

Quaternion operation is applied on the smartphone raw data to convert the data along their local coordinate to global coordinate.

Now the equation for the conversion is as follows:

$$a^{global} = qaq^{*}, \quad (35)$$

where

a is the parameter in smartphone local coordinate and consists of

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}, \quad (36)$$

q the quaternion,

q^{*} the conjugate of q.

Equation 35 can be elaborated as following:

$$= (q + q_x\hat{\imath} + q_y\hat{\jmath} + q_z\hat{k})(a_x\hat{\imath} + a_y\hat{\jmath} + a_z\hat{k})(q - q_x\hat{\imath} - q_y\hat{\jmath} - q_z\hat{k})$$

$$= \{(qa_x\hat{\imath} + qa_y\hat{\jmath} + qa_z\hat{k}) + q_xa_x(-1) + q_xa_y\hat{k} + q_ya_z\hat{\imath} + q_za_x\hat{\jmath} + q_xa_z(-\hat{\jmath}) +$$

$$q_ya_x(-\hat{k}) + q_ya_y(-1) + q_za_y(-\hat{\imath}) + q_za_z(-1)\}(q - q_x\hat{\imath} - q_y\hat{\jmath} - q_z\hat{k})$$

$$= (q^2a_x - 2qa_yq_z + 2qq_ya_z + q_x^2a_x + 2q_xa_yq_y + 2q_xa_zq_z - q_y^2a_x - q_z^2a_x)\hat{\imath} + (2qq_za_x$$

$$+ q^2a_y - 2qa_zq_x + 2q_xa_xq_y - q_x^2a_y + q_y^2a_y - q_z^2a_y + 2q_yq_zq_z)\hat{\jmath} + (-2qq_ya_x + 2qq_xa_y$$

$$+ q^2a_z + 2q_xa_xq_z - q_x^2a_z + 2q_yq_za_y + q_z^2a_z$$

$$- q_y^2a_z)\hat{k}. \quad (37)$$

Equation 37 will provide the values of a along the global X, Y and Z axis. This will ensure that the equations or techniques developed for maneuver identification will be same regardless of phone's orientation. It is noteworthy that similar quaternion conversion technique is applied by [13], however the final quaternion for orientation correction found by their method is different than the quaternion found in this research.

The whole conversion process can be expressed by the flow chart in Fig. 25.

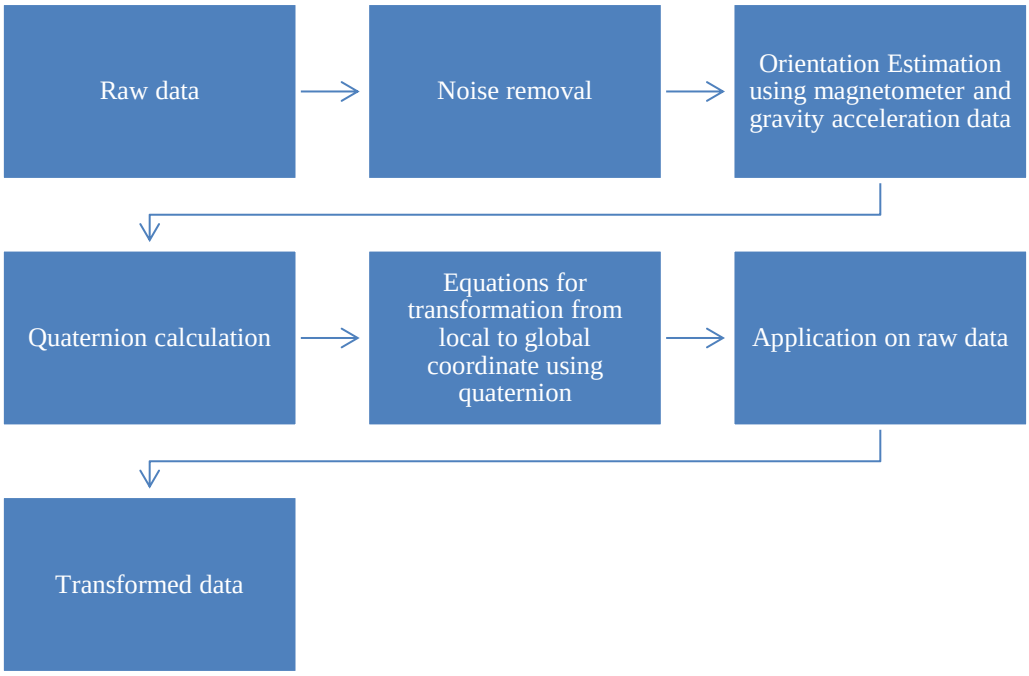

Fig. 25. Flow chart of orientation correction process.

After quaternion conversion the same two trips shown in Fig. 22 with different phone orientations, initially, shows a major variation of gyroscope readings along the Z axis of the global coordinate (Fig. 22 (e) and Fig. 22(f)).

5.3.3 Orientation Correction Method Validation

In order to check the accuracy of the orientation correction method, a validation technique is required. Validation is done by transforming raw gravity data obtained along a phone's local coordinate to a global coordinate. For a randomly oriented phone making angles with a horizontal surface, the raw gravity data are divided among local the X, Y, and Z axes, resulting in a component value of 9.8 ms^{-2}. After orientation correction, the transformed data represent data from a phone lying on a horizontal surface. For a phone lying on a horizontal plane, the gravity value should be 9.8 ms^{-2} along the global Z axis (perpendicular to the

horizontal plane) and near to 0 along the global X and Y axes. Thus, after orientation correction, if the transformed gravity readings are found to be 9.8 ms^{-2} along the global Z axis (perpendicular to the horizontal plane) and near to 0 along the global X and Y axes, then this validates the orientation correction technique.

This validation technique was applied and it was found that the transformed gravity readings of the Z component of the gyroscope readings were around 9.8 ms^{-2}, while the X and Y components were near to 0 (shown in TABLE 2). Thus, the orientation correction method was accurate.

TABLE 2 Gyroscope readings of raw data and data after orientation correction showing proof of orientation correction method validation

	Gravity acceleration readings (raw)			Gravity acceleration readings (after orientation correction)		
Index	X component	Y component	Z component	X component	Y component	Z component
1	1.19	9.312	2.833	5.13E-16	3.45E-15	9.807
2	1.18	9.337	2.754	-2.86E-16	8.03E-16	9.806
3	1.15	9.374	2.642	-8.30E-17	3.47E-17	9.807
4	1.14	9.394	2.574	-2.84E-16	7.96E-16	9.806
5	1.13	9.412	2.509	4.13E-16	6.99E-16	9.807
6	1.14	9.427	2.450	-5.67E-16	-1.60E-15	9.806

CHAPTER 6
MANEUVER IDENTIFICATION

At an intersection, vehicles typically execute 3 different maneuvers: left turn, right turn, and through. As the vehicle makes a turn, a phone's gyroscope experiences a high rotation rate as opposed to when the vehicle makes a through movement. This knowledge is used to differentiate the three maneuvers. At first, gyroscope data collected during multiple trips from Intersection 1 are plotted to see if there exists any pattern that could be used to differentiate the maneuvers. As the trip data for analysis are labeled data, the ground truth of the maneuver type is already known. Using the ground truth for each type of maneuver, gyroscope readings along the global Z axis from multiple trips are plotted. Fig. 26, Fig. 27and Fig. 28 show the plot for multiple left turn, right turn, and through trips. The limit for y axis is from -0.4 radian/second to 0.4 radian/second. The limit for x axis is from 1 to the number of data points in each trip. Observing these plots, a pattern is found. For left turn maneuvers, the gyroscope reading shows a hike in the time series, while for a right turn, it shows a depression in the time series. For through maneuvers, no such pattern is observed. This pattern can be explained by understanding the gyroscope data collection mechanism. As the vehicle makes a left turn, a phone is kept at stationary position in a random orientation inside the vehicle experiences rotation along its local axis. After orientation correction, the transformed data are such that the phone is laid on horizontal plane inside the vehicle. Thus, due to a vehicle making a left turn, the phone experiences anti-clockwise rotation along global Z axis. As the anti-clockwise direction is positive, as a result, positive readings of gyroscope data are generated and a spike in time series along the global Z axis in observed. Theoretically, as the vehicle executes the left turn gradually,

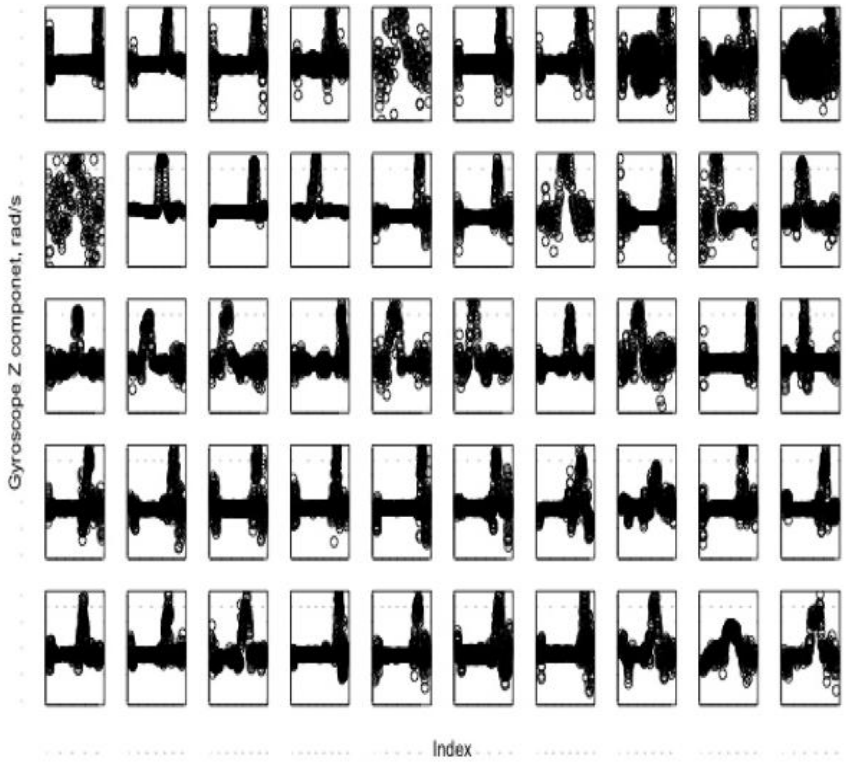

Fig. 26. Gyroscope plot for 50 left turn trips at Intersection 1.

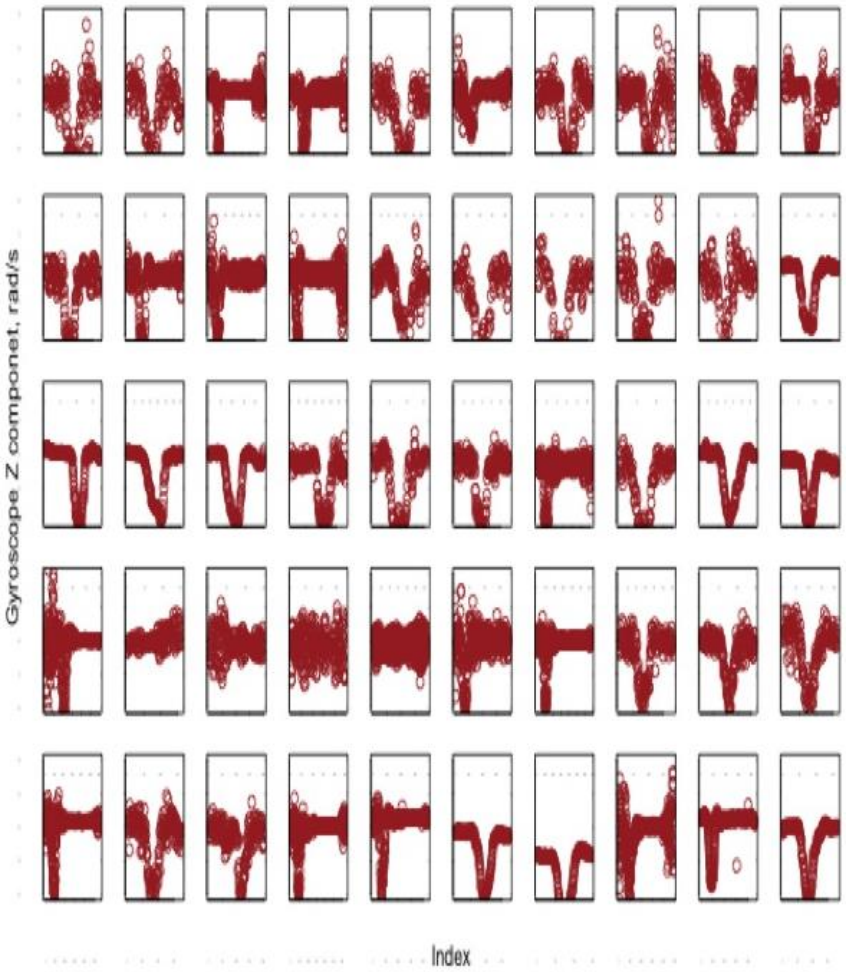

Fig. 27. Gyroscope plot for 50 right turn trips at Intersection 1.

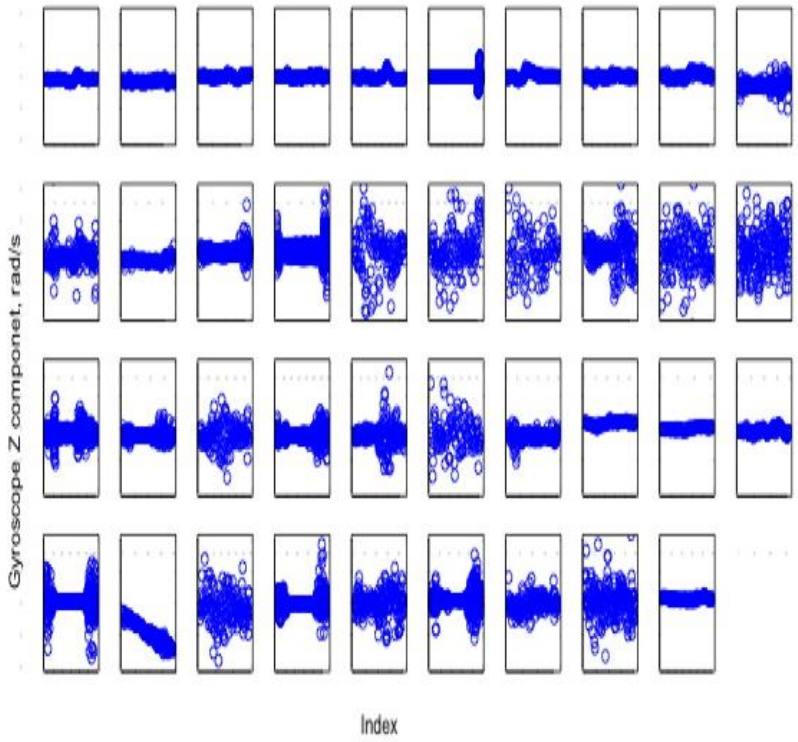

Fig. 28. Gyroscope plot for 39 through trips at Intersection 1.

the phone experiences anti-clockwise rotation gradually, thus, the signature pattern for the gyroscope reading along the global Z axis should be a concave shape. Similarly, when the vehicle makes a right turn, the phone inside it experiences clockwise rotation along global Z axis, thus generating a negative reading. Theoretically, as the vehicle executes a right turn gradually, the phone also experiences a clockwise rotation gradually, thus, the signature pattern for gyroscope reading along the global Z axis should be a convex shape. A depression in the gyroscope time series data along global Z axis is observed as a result. For through maneuvers, the phone does not experience any rotation, thus no spike or depression in gyroscope time series data should be observed. Theoretically, a straight line should be observed in gyroscope readings. However, in the plot for multiple through trips, there are some trips that show variation in

gyroscope time series data, which indicates that due to noise in data, the pattern may not always be observed. It is important to mention that, the time series for different trips contains different numbers of data points.

Not all trips contain clear patterns representing maneuver type. For the three intersections, number of trips showing no clear pattern in gyroscope global Z component readings is presented in TABLE 3. Among 774 total trips, 111 trips (14%) trips show no clear pattern.

TABLE 3 Number of trips for each intersection that shows no clear pattern in gyroscope readings

	# of trips with no clear pattern in gyroscope reading		
	Intersection 1	Intersection 2	Intersection 3
Left turn	19	15	1
Right turn	6	16	1
Through	18	35	0

The plots shown in Fig. 26, Fig. 27 and Fig. 28 are for transformed gyroscope reading along global Z axis after orientation correction. The local axis along which the maximum variation of gyro reading occurs depends on the orientation of the phone inside the vehicle (as mentioned in the data description section). Fig. 29(a) and Fig. 30(d) shows the gyroscope plot along a phone's local axis that experienced a high rotation rate during 3 different trips for left turns and right turns, respectively. For example, a plot of raw gyroscope readings for Trip 1 presented in Fig. 29(a) shows high rotation rate along phone's local X axis while for Trip 2, it is observed in the phone's local Y axis. Also, the signature pattern between left turn and right turn is not distinctive due to noise in the data and arbitrary orientation of the phone. Thus it is necessary to reduce

noise and apply orientation correction. Fig. 29(b) and Fig. 30(e) shows gyroscope plot for the same trips after noise reduction and orientation correction.

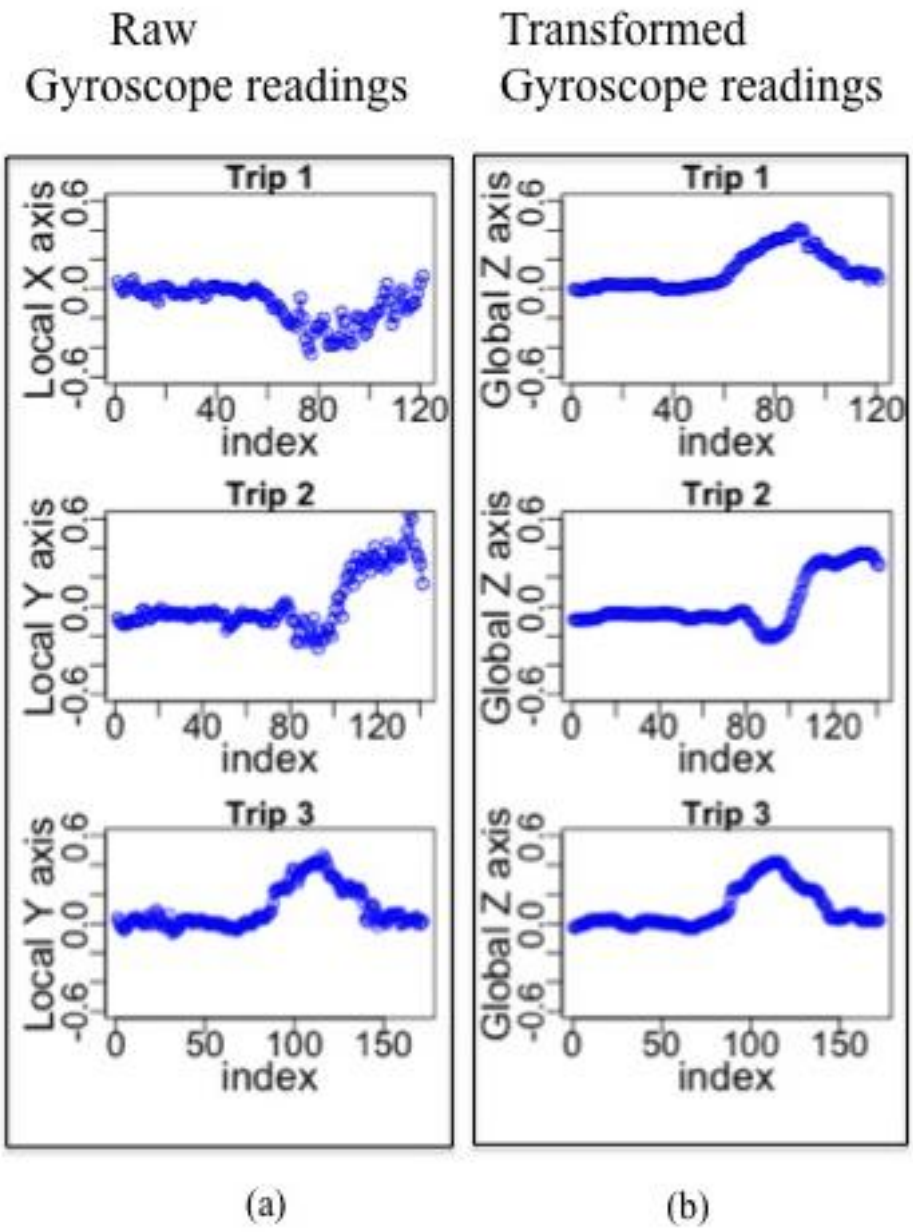

Fig. 29. Gyroscope plot for 3 different trips during left turn (a) before transformation and (b) after transformation.

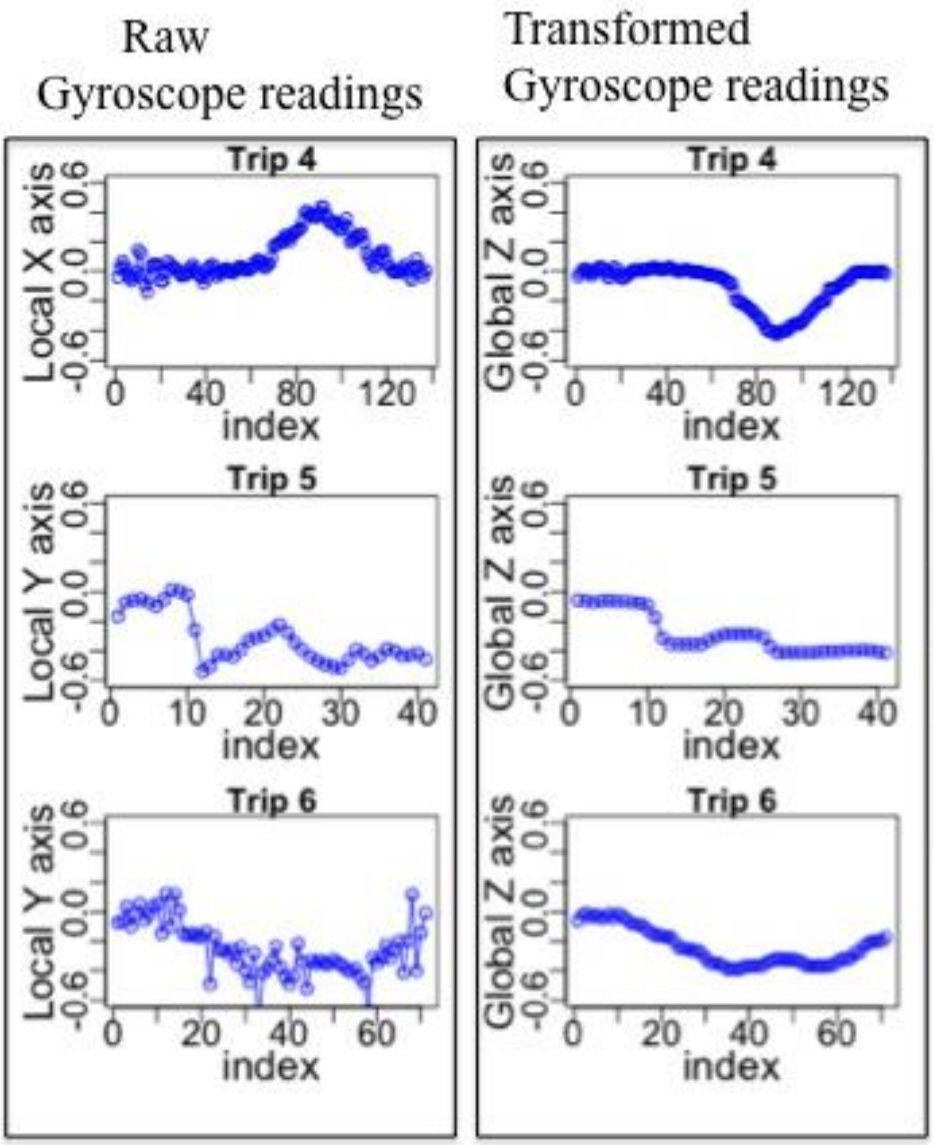

Fig. 30. Gyroscope plot for 3 different trips during right turn (d) before transformation and (e) after transformation.

As observed in Fig. 29(b) and Fig. 30(e), the plot of the transformed gyroscope readings show higher rotation rate along the global Z axis for turning vehicles and a particular pattern can be found for left turns and right turns. For left turns, the transformed gyroscope time series plot showed a concave shape [Fig. 29 (b)], while for right turns, a convex shape is observed [Fig. 30 (e)]. As the vehicle makes a left turn, a phone lying in a horizontal plane experiences a counter clockwise rotation along global Z-axis, thus generating positive readings and a concave shape in the gyroscope time series plot. If the vehicle is making a right turn, then the phone experiences clockwise rotation and generates negative readings in gyroscope and a convex shape in the gyroscope time series plot. The signature pattern for left turns and right turns is presented in Fig. 31 and Fig. 32. For through maneuvers, the phone does not experience any rotation and thus no high rotation rate will be observed in the gyroscope reading.

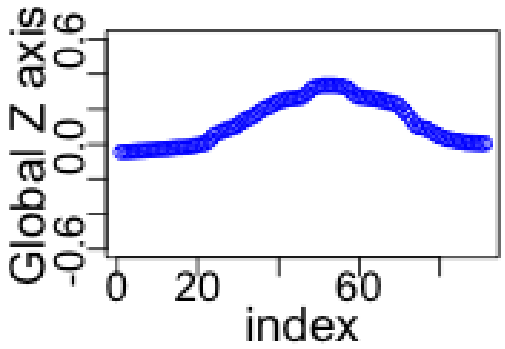

Fig. 31. Signature pattern for left turn.

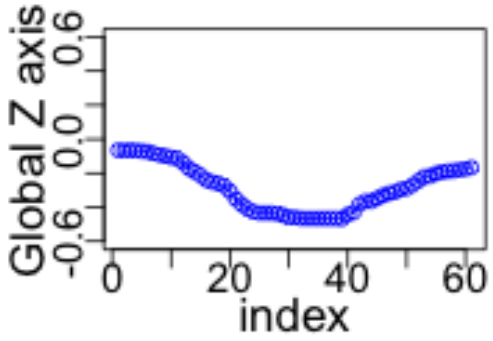

Fig. 32. Signature pattern for right turn.

6.1 Distinctive Feature Identification

Feature identification is an important step for algorithm development as features impact the accuracy of classification. There are several methods for feature evaluation and selection. A simple method of assessing feature quality is visualization of the features. In this research, boxplots are employed as a means for visualizing the distribution of data and identifying distinctive features. As observed, the distinctive pattern is found for the Z component of the transformed gyroscope data for the three maneuver types. Thus, the distribution of the Z component of transformed gyroscope data is explored using the boxplots. Data from three intersections have been analyzed. The features that are selected to visualize are: mean, median, maximum, minimum, and range. Based on ground truth, left turns, right turns, and through maneuvers are separated. For each trip, the mean, median, maximum, minimum, and range values have been calculated. For a particular intersection, the mean value of the selected 5 features for all left turn trips, right turn trips, and through trips have been calculated. Using these as inputs, boxplots for each type of maneuvers in an intersection, have been generated for the selected 5 features.

Fig. 33, Fig. 34 and Fig. 35 show the boxplots for mean, median, maximum, minimum, and range for Intersection 1. The central rectangle equals to the inter quartile range (IQR) is calculated as follows:

$$IQR = Q_3 - Q_1 , \tag{38}$$

where Q_3 3rd quartile

Q_1 1st quartile.

The rectangular box represents the 1st quartile, median, and 3rd quartile of the particular feature. The upper whisker is located at the smaller of the maximum x value and Q_3 + 1.5*IQR, lower whisker is located at the larger of the smallest x value and Q_1 – 1.5*IQR. The circular points represent the outliers that lie outside the whiskers.

For Intersection 1, the boxplots are shown in Fig. 33. The boxplot for maximum values of Z component of transformed gyroscope readings show that for the distribution for gyroscope readings during left turns does not have overlap with the distributions for gyroscope readings during right turns and through maneuvers. However, there is overlap between the distribution of gyroscope readings during right turn and through maneuver. That indicates that the maximum value of the gyroscope readings will not be a distinctive feature for maneuver classification. The boxplot for the minimum values of the Z component of the transformed gyroscope readings show that there is overlap between the distribution of gyroscope readings during left turns and through maneuvers. The boxplots for mean and median values show that there is overlap among the distributions of gyroscope readings during all three maneuvers. The least overlap between the distributions of gyroscope readings during left turns, right turns, and through maneuvers is observed for range values. Thus, it is identified as the most distinctive feature for maneuver classification. Similar observations were made for the boxplots for Intersection 2 (Fig. 34) and

Intersection 3 (Fig. 35). For all the 3 intersections, range was found to be the most distinctive feature for maneuver classification.

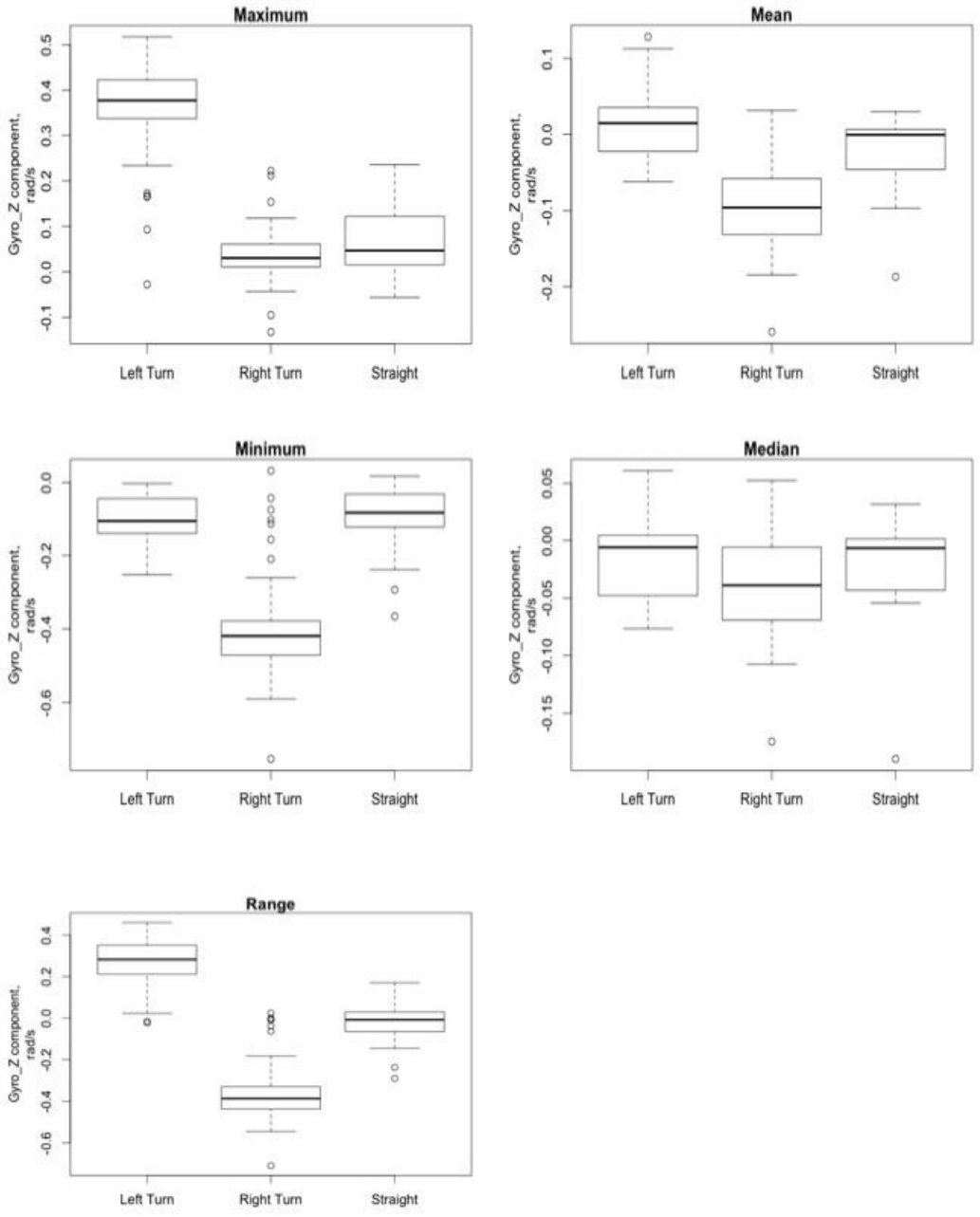

Fig. 33. Boxplots for features of gyroscope values for trips at Intersection 1.

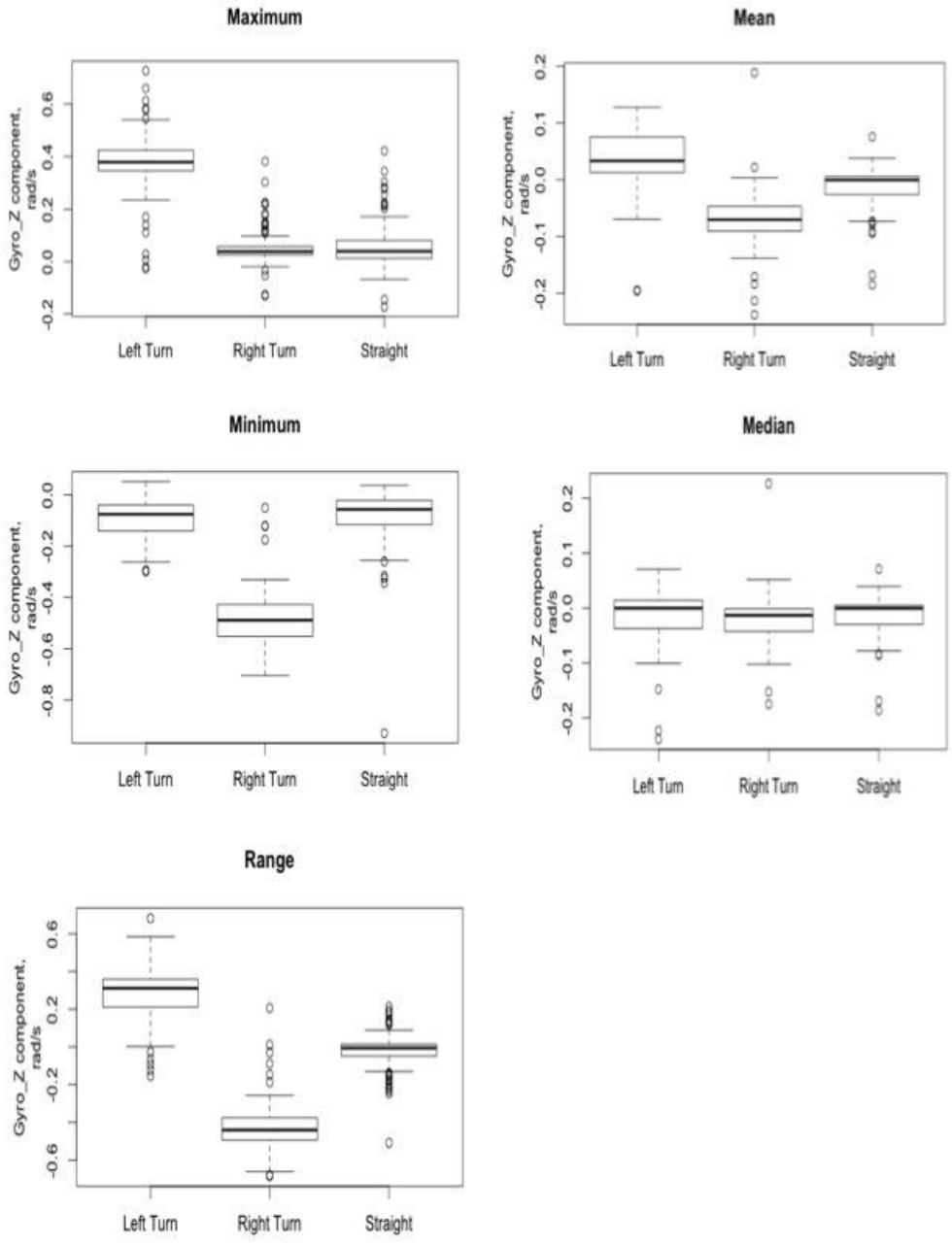

Fig. 34. Boxplots for features of gyroscope values for trips at Intersection 2.

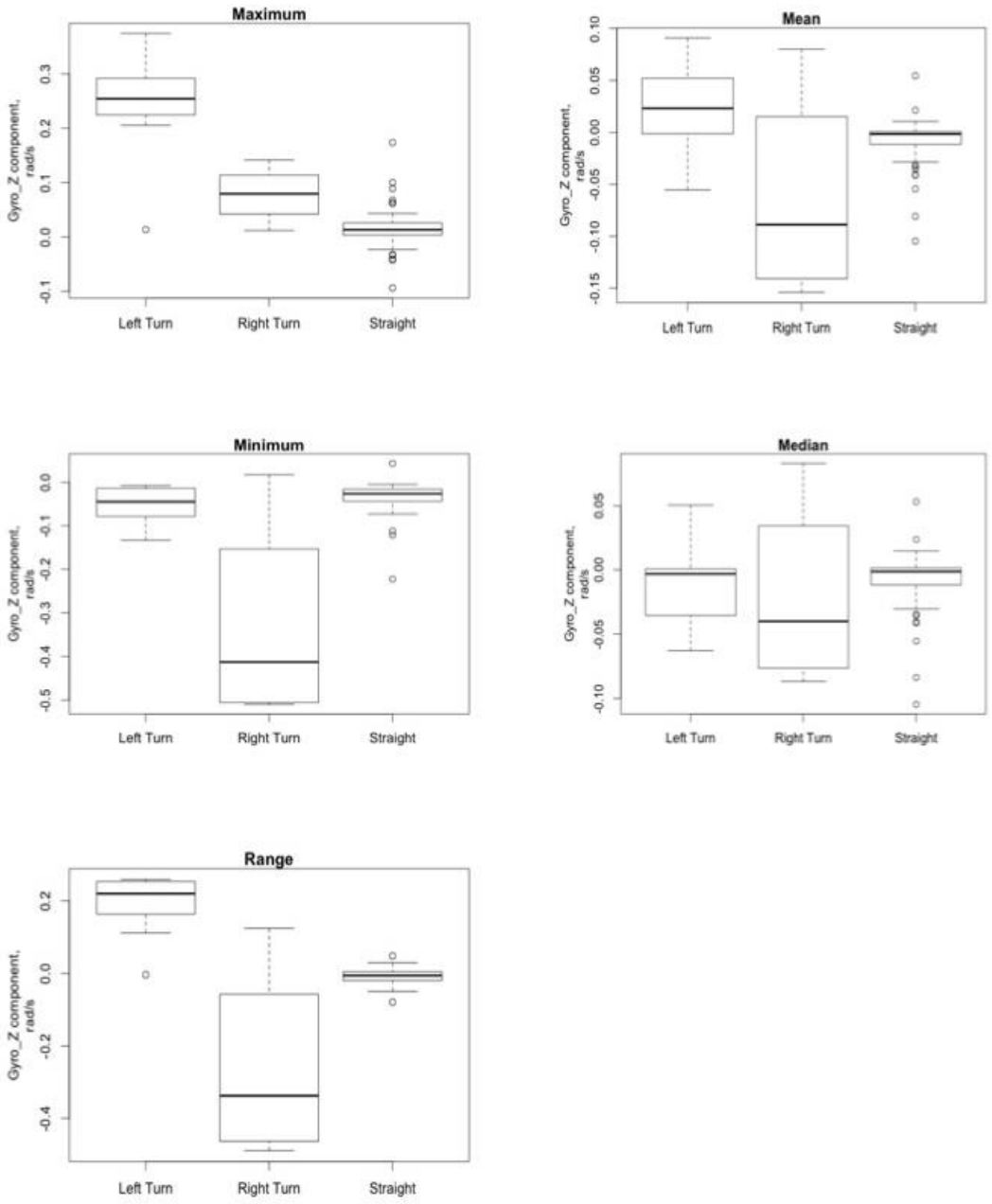

Fig. 35. Boxplots for features of gyroscope values for trips at Intersection 3.

6.2 Maneuver Identification Algorithm Based on Threshold of Gyroscope Reading

An algorithm is empirically developed to identify the 3 different types of maneuver. The range value of the Z component of transformed gyroscope readings are employed for the algorithm development.

For any particular trip, the range, r is calculated as in

$$\text{Range, } r = \max\{x_1, x_2, x_3, \dots\dots, x_n\} - \min\{x_1, x_2, x_3, \dots\dots, x_n\}, \quad (39)$$

where *x* is the Z component of gyroscope reading for a trip with 1 to n data points.
For a particular trip, if r is greater than 0.29, the maneuver executed in the trip is identified as turn, else identified as through. For any particular trip with maneuver identified as turns, if the absolute value of maximum positive gyroscope reading is higher than the absolute value of maximum negative gyroscope reading, then the turn is identified as a left turn (LT), else it is identified as a right turn (RT). The value of r for the algorithm is set to 0.29 empirically and based on sensitivity analysis done on a training data set consisting of 35 trips. The results for sensitivity analysis is presented in TABLE 4, TABLE 5 and TABLE 6. Among the 35 trips, 8 were left turns, 8 were right turns, and 19 were through maneuvers. The value of r has been varied from 0.26 to 0.34 to see the accuracy of the algorithm on identifying the maneuver type of the training data set. It was observed that for r=0.26, all the left turn trips were correctly identified (100% accuracy), 7 right turn trips were correctly identified (88% accuracy) and 13 through trips were correctly identified (68% accuracy) (TABLE 4). The overall accuracy of the maneuver identification algorithm for 0.26 was 85%. For r values ranging from 0.27 to 0.28, the algorithm correctly identified 8 left turn trips, 7 right turn trips, and 16 through trips with an accuracy of 100%, 88% and 84%, respectively (TABLE 5). The overall accuracy of the algorithm for the range of values was 91%. For r values 0.29, the algorithm correctly identified 8

left turn trips, 7 right turn trips and 19 through maneuvers with an accuracy of 100%, 88% and 100% respectively. The overall accuracy of the algorithm was 96%. The accuracy of the algorithm was stabilized at 96% and did not improve further with increase in r values (TABLE 6). Thus the value of r is selected to be 0.29.

TABLE 4 Sensitivity analysis results for maneuver identification algorithm with $r = 0.26$

		Identified				
r		Left Turn	Right Turn	Through	%Accuracy	Overall Accuracy%
0.26	Left Turn	8	0	0	100	85
	Right Turn	0	7	1	88	
	Through	3	3	13	68	

TABLE 5 Sensitivity analysis results for maneuver identification algorithm with *r* values varying from 0.27 to 0.28

		Identified				
r		Left Turn	Right Turn	Through	%Accuracy	Overall Accuracy%
0.27 - 0.28	Left Turn	8	0	0	100	91
	Right Turn	0	7	1	88	
	Through	2	1	16	84	

TABLE 6 Sensitivity analysis results for maneuver identification algorithm with *r* values 0.29 and higher

		Identified				
r		Left Turn	Right Turn	Through	%Accuracy	Overall Accuracy%
>= 0.29	Left Turn	8	0	0	100	96
	Right Turn	0	7	1	88	
	Through	0	0	19	100	

6.3 Results of Maneuver Identification Based on Threshold of Gyroscope Readings

The above-described algorithm is applied to smartphone data collected during trips from 3 four-legged intersections situated in Norfolk, VA and Virginia Beach, VA. The accuracy of the algorithm is assessed by matching the identified maneuver type with actual maneuver made by the vehicle at any particular intersection. The GPS coordinates are used as ground truth for detecting the actual maneuver made by the vehicle through the intersection for any particular trip.

From Intersection 1, a total of 274 trips data were collected, of which 145 of the trips consisted of left turns only, 90 trips were of right turns only, and 39 trips were of through maneuvers only along the intersection. The duration of the trips varied from minimum 14 seconds to 55 seconds. The sampling rate was set to 10 samples per second. Thus, each trip contained a minimum 140 to a maximum 550 data points. Seven different drivers made those trips. The results in TABLE 7 are presented as a confusion matrix where each row represents the number of left turns, right turns, and through maneuvers in an actual class, while each column represents the number of left turns, right turns, and through maneuvers in an identified class. Of

the 145 actual left turns, the algorithm accurately identified 143 trips as left turns, while it inaccurately identified 2 left turns as through maneuvers. Similarly, out of 90 right turn trips, the algorithm correctly identified 85 trips as right turns and inaccurately identified 5 trips as through maneuvers. Of the 39 actual through trips, the algorithm accurately identified 36 trips as through maneuvers and inaccurately identified 2 trips as left turns and 1 trips as a right turn. The algorithm had 99% accuracy for identifying left turns, 95% accuracy for identifying right turns, and 93% accuracy for identifying through maneuvers. The results for Intersection 2 are shown in TABLE 8. 7 different drivers executed the trips. Of the 124 actual left turns, the algorithm accurately identified 118 trips as left turns with 100% accuracy. Out of 133 right turn trips, the algorithm correctly identified all 133 trips as right turns (100% accuracy). Of the 157 actual through trips, the algorithm accurately identified 154 trips as through (99% accuracy).

The results for Intersection 3 are presented at TABLE 9 as a confusion matrix. Of the 9 actual left turns, the algorithm accurately identified 5 trips as left turns (56% accuracy). The algorithm correctly identified all 3 right turn trips (100% accuracy). Of the 56 actual through trips, the algorithm accurately identified all the 56 trips as through maneuvers (100% accuracy). Investigating the reason of the inaccurate identification of left turn trips revealed that for those trips, the gyroscope did not sense much rotation while the vehicle executed the turns. Intersection 3 is a wide intersection and the vehicles were executing the turn very slowly. As a result, the phone were experiencing very little rotation per second, which was below the defined range (0.29 rad/second) to identify turns. This indicates that r value should be selected based on multiple intersections with various intersection widths.

The drawback of the algorithm is that the algorithm relies on the threshold of gyroscope values. A gyroscope is a sensitive sensor. Although during the data collection process, the

smartphones are kept in stationary positions, they were not always kept fixed by a smartphone holder. Sometimes the smartphones were just kept lying in a cup holder. Due to potholes in the road surface or sudden braking of the car, the smartphone may experience some movement, which might result into higher values in the gyroscope readings. In such cases, even though the vehicle did not make any turn, due to the phone experiencing some movements, the developed algorithm falsely identify the through maneuvers as turns. This might be the reason of identifying a number of through maneuvers as left and right turns. In addition as mentioned, if the phone experienced very little rotation due to the vehicle executing the turn very slowly, as happened in Intersection 3, in such cases, this threshold method will fail to identify turns correctly.

TABLE 7 Results for maneuver identification by threshold of gyroscope readings for Intersection 1

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	143	0	2	99
Right Turn	0	85	5	95
Through	2	1	36	93

TABLE 8 Results for maneuver identification by threshold of gyroscope readings for Intersection 2

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	118	2	4	96
Right Turn	0	133	0	100
Through	3	0	154	99

TABLE 9 Results for maneuver identification by threshold of gyroscope readings for Intersection 3

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	5	0	4	56
Right Turn	0	3	0	100
Through	0	0	56	100

6.4 Maneuver Identification by DTW Distance

In this section, a clustering based algorithm for maneuver classification is described. As mentioned previously, there are similarities in patterns of gyroscope readings for each type of maneuver. Thus, it is invaluable to explore if the maneuvers can be classified based on the patterns in the gyroscope readings. For each intersection, a trip is selected as template for each

maneuver. Observing the gyroscope global Z component plots for each maneuver and visually selecting the trip that matches closely to the signature pattern, the template trip is selected. The gyroscope Global Z component plot for template trip representing left turn, right turn, and through maneuver, respectively for Intersection 1 is shown in Fig. 36, Fig. 37 and Fig. 38. The distance between the times series of each trip with the three template trip is calculated using DTW algorithm. The distance between the template left turn trip and any particular trip making a left turn maneuver is expected to be small, as it is expected that the gyroscope time series pattern will be similar. On the other hand, distance between the template left turn tip and any particular trip making right turn or through maneuver is expected to be large. Thus, each trip is categorized based on the maneuver type of template trip that has smallest distance with that trip. For example, if any particular trip is found to have smallest DTW distance between the template trip for a left turn maneuver, then that particular trip is identified as a left turn. In R software, “dtw” package is used for calculating the DTW distance between the gyroscope readings of trips [40].

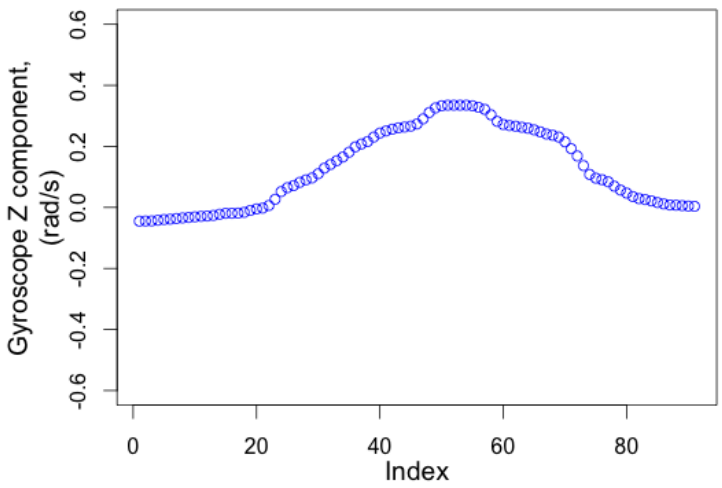

Fig. 36. Gyroscope time series plot of template trip representing left turn maneuver at Intersection 1.

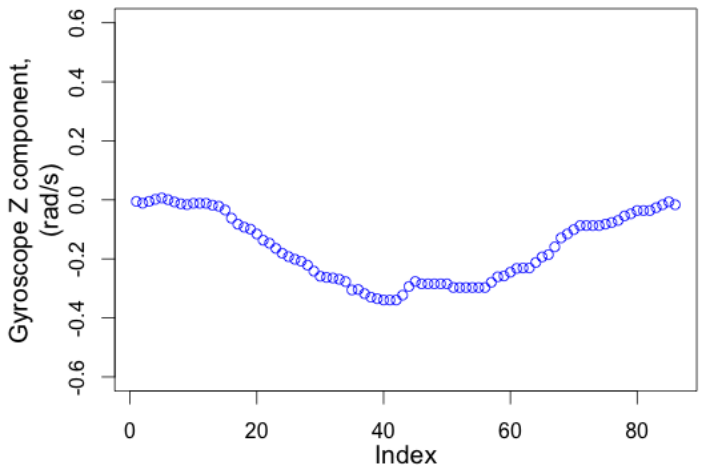

Fig. 37. Gyroscope time series plot of template trip representing right turn maneuver at Intersection 1.

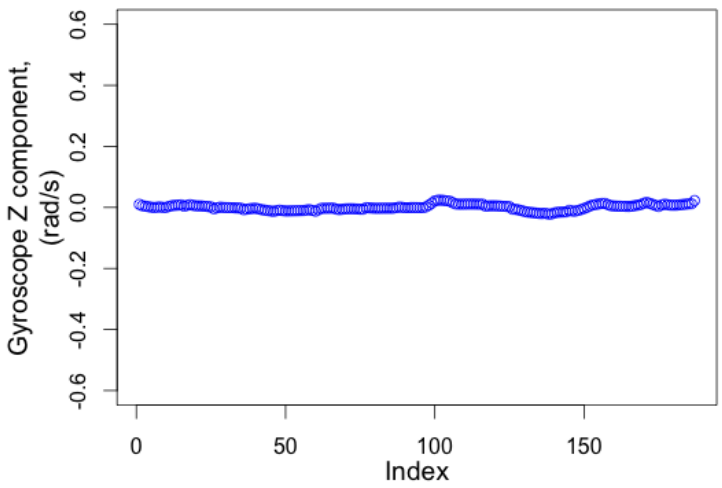

Fig. 38. Gyroscope time series plot of template trip representing through maneuver at Intersection 1.

In order to calculate the DTW distance between gyroscope time series of a template trip and a candidate trip, the warping matrix, w is calculated (Fig. 39). The element of w, w(i, j) is calculated as follows:

$$w(i,j) = d(a_i, b_j) + \min\{w(i-1,j-1), w(i-1,j), w(i,j-1)\}, \quad (40)$$

where $d(a_i, b_j) = |a_i - b_j|$. (41)

The conditions of $d(a_i, b_j)$ are as follows:

$$d(a_i, b_j) = \begin{cases} 0, \text{if } i = 0, j = 0 \\ \text{Infinity, if } i = 0 \text{ or } j = 0 \\ d(a_i, b_j), \text{if } i = 1 \text{ and } j - 1 \end{cases}. \quad (42)$$

Minimum path is found by starting from w(1, 1) and continuing to adjacent cell with minimum w(i, j) path following monotonicity, continuity, and boundary constraints that are discussed in Chapter 4. The total DTW distance between the template trip and candidate trip is then

$$\text{DTW distance} = \frac{w(n,m)}{k}, \tag{43}$$

where k is number of cells in the path.

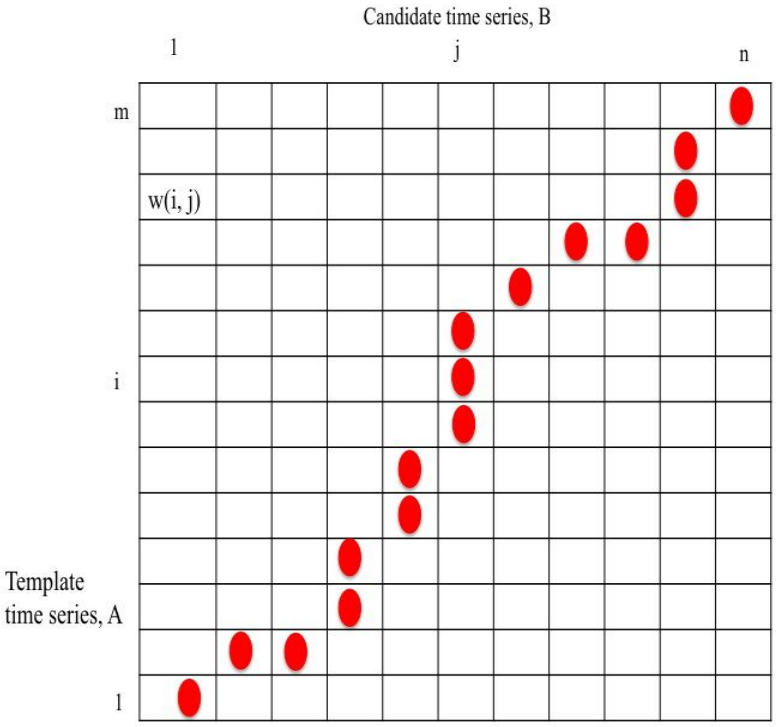

Fig. 39. Warping matrix for calculating DTW distance.

6.5 Results of Maneuver Identification by DTW Distance

The results of maneuver identification based on DTW distance are presented as a confusion matrix for the three intersections in TABLE 10, TABLE 11 and TABLE 12. For Intersection 1, the algorithm has a 70% accuracy of identifying left turns, a 96% accuracy of identifying right turns, and a 87% accuracy of identifying through maneuvers. For Intersection 2, the algorithm has a 91% accuracy of identifying left turns, a 99% accuracy of identifying right turns, and a 95% accuracy of identifying through maneuvers. For Intersection 3, the algorithm

has a 100% accuracy of identifying left turns, a 100% accuracy of identifying right turns, and a 76% accuracy of identifying through maneuvers. For Intersections 1 and 2, the method based on a DTW distance/dissimilarity matrix yielded less accuracy than the threshold of the gyroscope value algorithm. For this method to correctly identify maneuvers, it is expected that the left turn and right turn have a signature pattern similar to Fig. 31 and Fig. 32. The left turn should have a concave curve while right turn should have a convex curve. However, this concave/convex pattern is not observed in all the turn trips. For example, as shown in Fig. 29(b), Trip 2, although it was a left turn trip, the gyroscope plot has a convex followed by concave shape. Similarly, in Fig. 30(b) Trip 2 shows a gyroscope plot for a right turn trip where no distinctive convex shape is observed. When the DTW distance between the gyroscope readings of a trip showing distinctive turn pattern and that of a trip with no distinctive pattern is calculated, the distance is expectedly found to be large. Thus the algorithm might inaccurately identify the maneuvers. For Intersection 3, the accuracy for left turns and right turns identification is 100%. This indicates that, although the threshold of gyroscope reading was not large enough to accurately identify the turns (as seen in Section 6.1.2), the pattern for left and right turns existed in the time series.

Overall, this method has less accuracy than the method based on threshold value of gyroscope readings for maneuver identification.

TABLE 10 Results for maneuver identification by DTW distance for Intersection 1

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	101	10	33	70
Right Turn	0	85	4	96
Through	4	1	33	87

TABLE 11 Results for maneuver identification by DTW distance for Intersection 2

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	112	2	9	91
Right Turn	0	131	1	99
Through	4	4	148	95

TABLE 12 Results for maneuver identification by DTW distance for Intersection 3

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	8	0	0	100
Right Turn	0	2	0	100
Through	11	2	42	76

6.6 Maneuver Identification by Clustering Based on Maximum and Minimum Value of Gyroscope Reading

As mentioned in Section 6.1, the boxplots for the gyroscope readings for the maneuvers indicated that the range is the most distinctive feature for identifying maneuver type. Here, using the ground truth value for maneuver type, the maximum and minimum values of the Gyroscope Z component readings for each trip are plotted together for each intersection to see if any cluster can be seen visually. The plots are presented in Fig. 40, Fig. 41, and Fig. 42. The X axis and Y axis of the plot represents the maximum and minimum of the gyroscope global Z component readings for each trip respectively. Each circle in the plot represents each different trip for that particular intersection. The plots show that the trips from each maneuvers form 3 clusters of trips in each intersection. However, in real scenarios, the objective is to identify the type of maneuver for each trip, as no ground truth value for the maneuver type is known. Thus, an appropriate clustering technique can be effective in identifying the cluster representative trip and the member trips for each cluster. After identification of the members forming each cluster, another algorithm will be applied to identify the maneuver that each cluster represents.

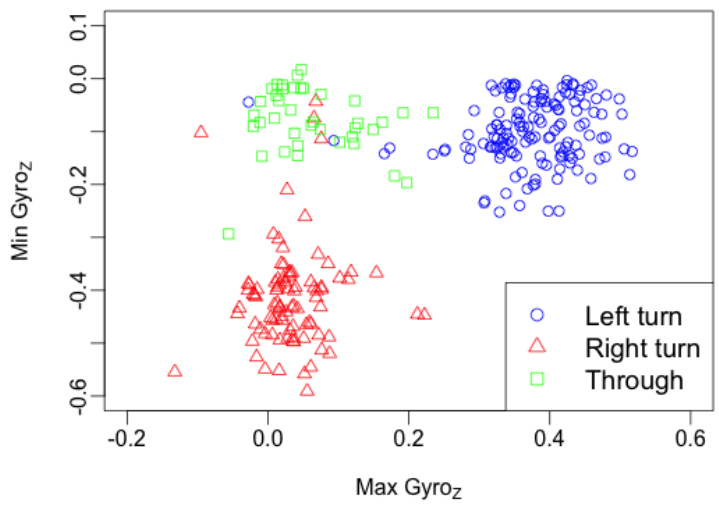

Fig. 40. Plot of maximum and minimum gyroscope Z component readings for trips at Intersection 1.

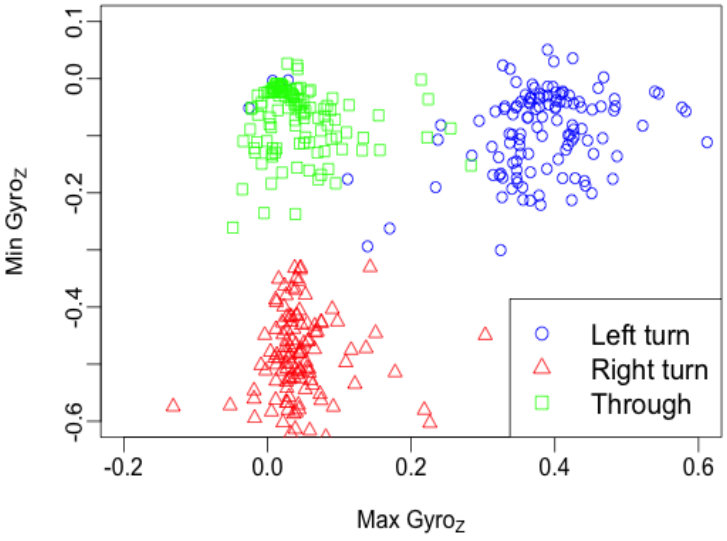

Fig. 41. Plot of maximum and minimum gyroscope Z component readings for trips at Intersection 2.

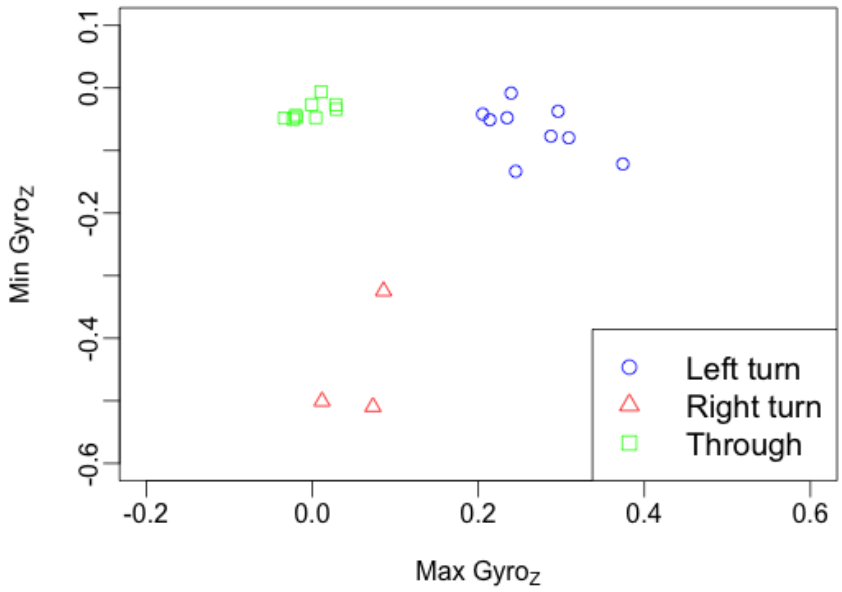

Fig. 42. Plot of maximum and minimum gyroscope Z component readings for trips at Intersection 3.

A K-medoids clustering based on the minimum and maximum of gyroscope global Z component readings is employed to see the effectiveness of clustering the three maneuver types. The maximum and minimum values of gyroscope readings from each trip has been provided as input and k-medoids clustering is done on that input. In R software, “PAM” (Partitioning Around Medoid) toolbox is used for the clustering [41]. The term “medoid” refers to an observation for a particular cluster. The sum of the distances between that medoid and all the other members in that particular cluster is minimum. “PAM” toolbox identifies the clusters and produces the number of members of each cluster.

To assess the accuracy of the clustering or the fitness of each member in the assigned cluster, a silhouette plot is employed. A silhouette plot is a plot of a measure known as silhouette width [42]. Silhouette width is the ratio of the closeness of a particular member to other members

in its own cluster to the closeness of it to members in other clusters. Suppose, j is a particular observation which belongs to a cluster K. d(j) is the average dissimilarity between j and all other members of the cluster K. For all other clusters m, d(j,m) is the average dissimilarity of j to all members of m. The dissimilarity between j and its "neighbor" cluster, i.e., the nearest one to which it does not belong is d(j,m) is e(j). e(j) is the minimum average dissimilarity between j to other clusters except the one it belongs to[42].

The silhouette width s(j) is defined as follows:

$$s(j) = \frac{(e(j)-d(j))}{\max(d(j),e(j))} \ . \tag{44}$$

The value of s(j) near 1 means that the observation is well placed in its cluster, while value near 0 mean that it's likely that an observation might really belong to another cluster.

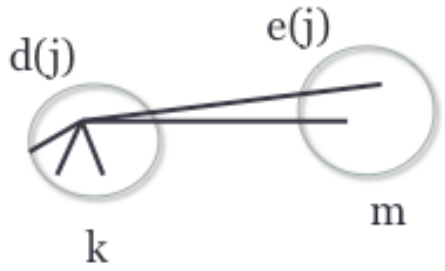

Fig. 43. Distance between a particular observation with other observations.

The silhouette plot is the plot of these silhouette width for each observation. For well fitted clusters, each member should have a silhouette width value close to 1. If the silhouette plot shows that for many observations the values are closer to zero then it means that the fit for each observations in its assigned cluster was not good. The overall fit is evaluated with the calculated

value "average silhouette width" which represents the strength of the structure of the cluster. TABLE 13 show the meaning of the values [42].

TABLE 13 Interpretation of silhouette width values [42]

Average silhouette width	Interpretation
0.71- 1.0	A strong structure has been found
0.51 – 0.70	A reasonable structure has been found
0.26 – 0.50	The structure is weak and artificial
< 0.25	No substantial structure has been found

In order to show the clusters in a plot more distinctively, PCA is applied on the bivariate data set. PCA yields first component in terms of correlations between the two variables, such that the greatest variance of the data set comes to lie on the first axis, i.e., the first principal component, the second greatest variance on the second axis, i.e., the second principal component and so on[43]. For example, for a data set consisting of X and Y, a scatterplot is shown in Fig. 44. The best fitted regression line in terms of X and Y and its orthogonal line is shown by dotted lines. This regression line is able to show the most variation of the data set. This line is the first principle component. The orthogonal line to the best fitted regression line is the second principle component (see Fig. 45). Plotting the first and second component will show clusters of the data, if there is any similarity within the data points. However, this method only divides the trips into clusters. It does not automatically identify which cluster represents which maneuver. For that identification, the maximum and minimum value of the gyroscope global Z component reading of each cluster representative trip is observed. As shown in the distinctive feature identification

section, maximum and minimum value will indicate the type of maneuver that cluster representative belongs to.

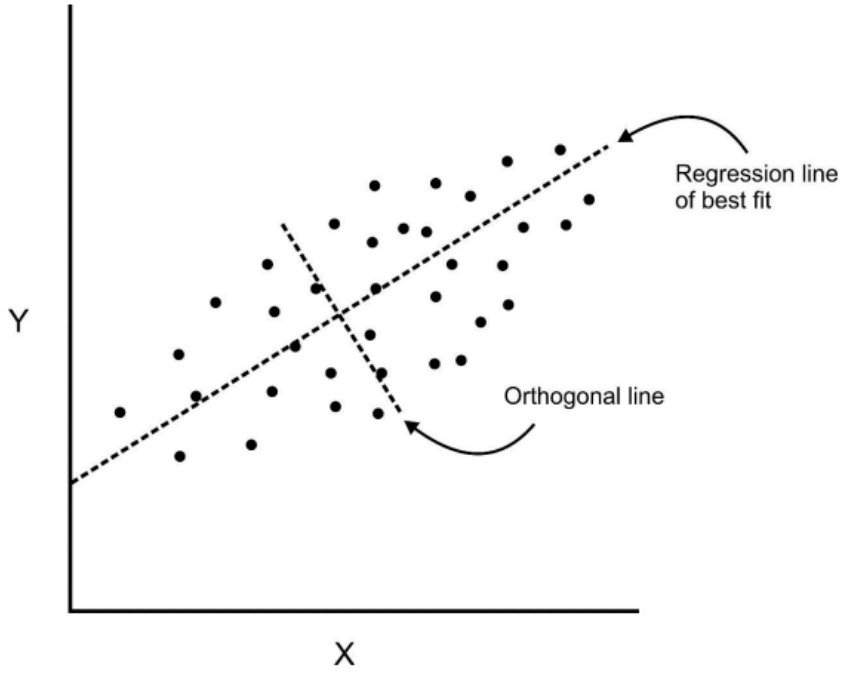

Fig. 44. Scatterplot of a bivariate dataset and the regression lines [43].

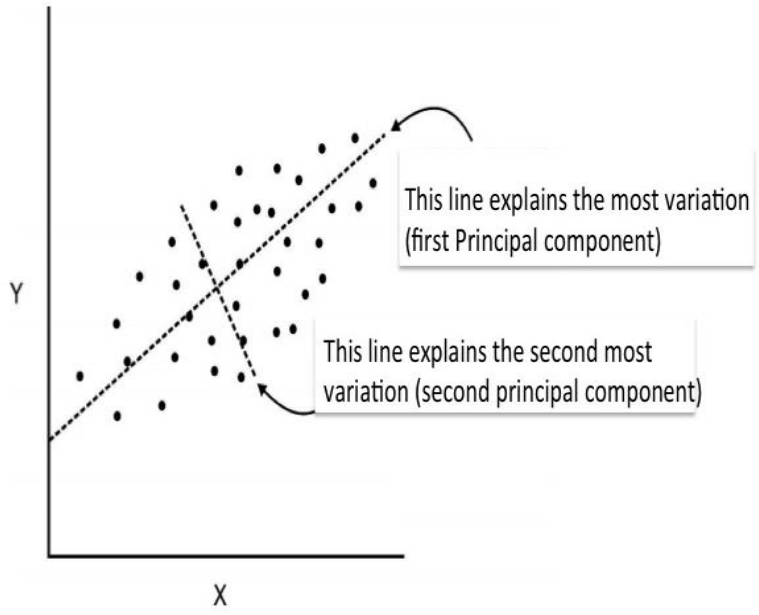

Fig. 45. The scatterplot showing the principal components [43].

6.7 Results of Maneuver Identification by Clustering Based on Maximum and Minimum Gyroscope Readings

Results of clustering based on the features (minimum and maximum gyroscope readings of each trip) performed well to cluster the trips. For Intersection 1, the clustering algorithm divided the trips into 3 clusters. The first cluster has 46 observations and average silhouette width 0.64. The second cluster has 142 observations and average silhouette width 0.67. The third cluster has 86 observations and average silhouette width 0.70. The silhouette width refers that the observations are well fitted to its assigned cluster. The overall average silhouette width 0.67 indicates that all these clusters have reasonably strong structure (Fig. 46). The cluster plot shows that the clusters are distinctive and does not have much overlap with each other (Fig. 47).

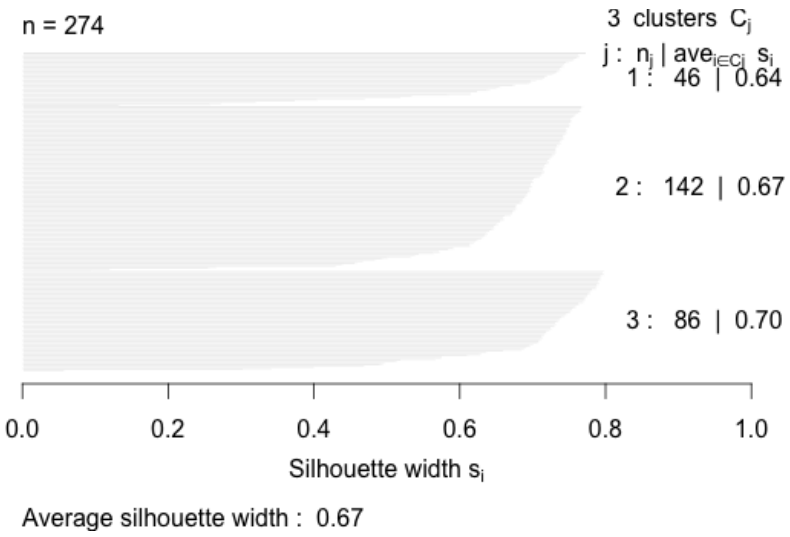

Fig. 46. Silhouette plot of clusters for trips at Intersection 1 for clustering based on features.

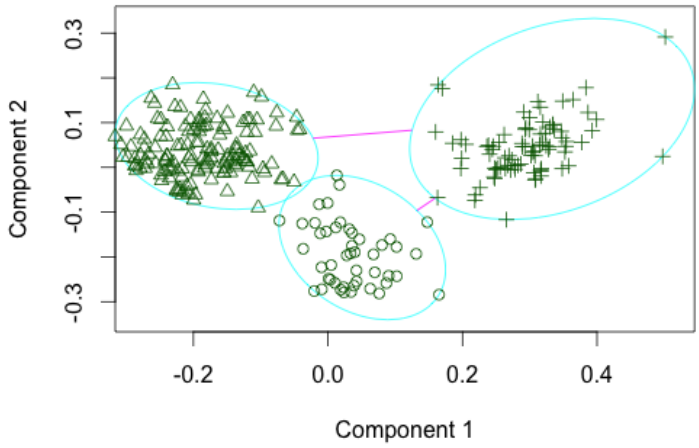

Fig. 47. Cluster plot of clusters for trips at Intersection 1 for clustering based on features.

Similar results were also observed for Intersection 2, where the k-medoids clustering based on maximum and minimum gyroscope values for each trip divided the trips into 3 clusters.

The first, second, and third clusters contain 156, 123, and 135 observations, respectively. The average silhouette width for the 3 clusters is 0.74, 0.64, and 0.69, respectively, which indicates that the observations are well fitted in their assigned cluster. The overall average silhouette width is 0.69, which also indicates that the clusters have strong structure (Fig. 48). The cluster plot also shows 3 distinctive clusters with some overlap (Fig. 49).

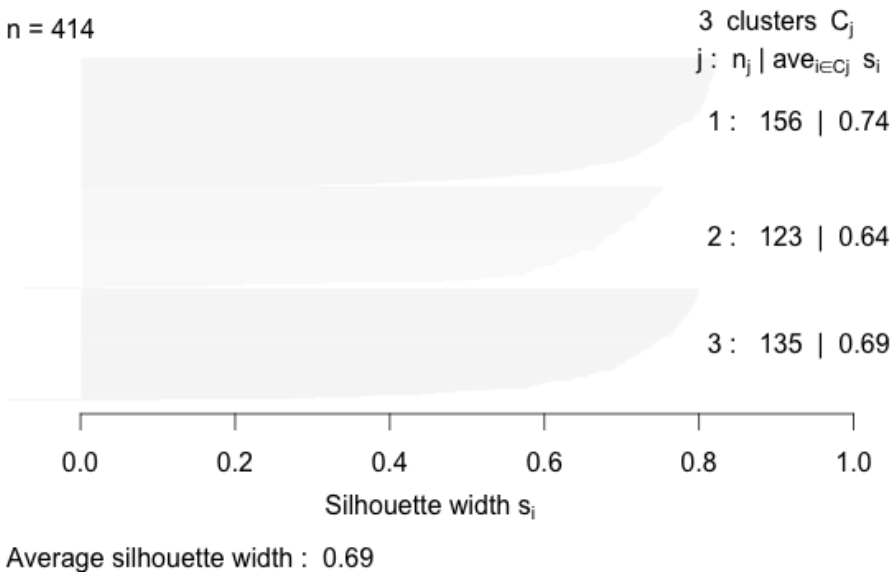

Fig. 48. Silhouette plot of clusters for trips at Intersection 2 for clustering based on features.

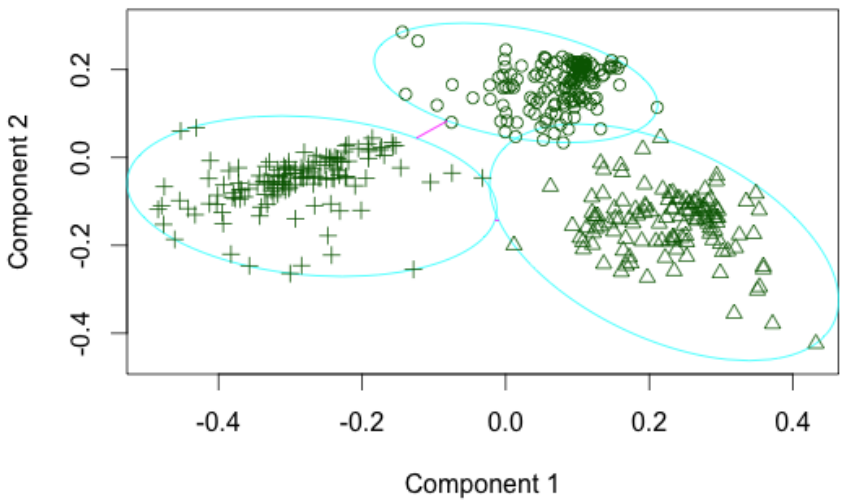

Fig. 49. Cluster plot of clusters for trips at Intersection 2 for clustering based on features.

For the data from Intersection 3, the k-medoids clustering performed well. The clustering divided the trips data into 3 clusters. The first, second, and third clusters contain 9, 56, and 3 observations, respectively. The average silhouette width for the 3 clusters is 0.67, 0.82, and 0.62, respectively, which indicates that the observations are well fitted in their assigned cluster. The overall average silhouette width is 0.79, which also indicates that the clusters have strong structure (Fig. 50). The cluster plot also shows 3 distinctive clusters with minimal overlap (Fig. 51). Thus k-medoids clustering based on maximum and minimum gyroscope values for each trip seem to be effective in identifying the clusters accurately.

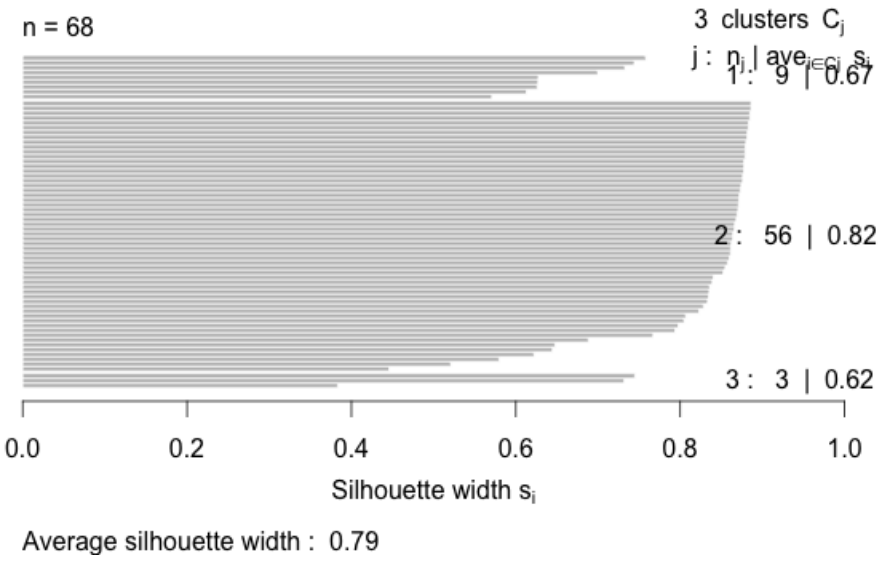

Fig. 50. Silhouette plot of clusters for trips at Intersection 3 for clustering based on features.

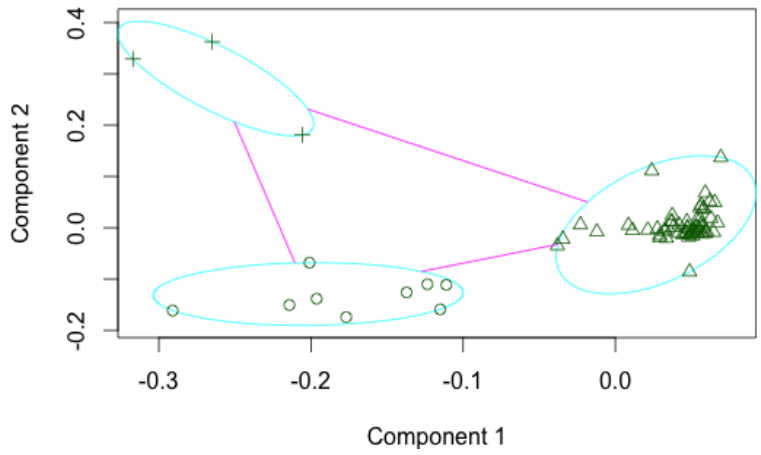

Fig. 51. Cluster plot of clusters for trips at Intersection 3 for clustering based on features.

However, this method does not automatically identify which cluster represents which maneuver. For that identification, maximum and minimum value of the gyroscope global Z component reading of each cluster representative trip is observed. Maximum and minimum values indicate the type of maneuver to which the cluster representative belongs. Using this method of

identification, the confusion matrix of results of maneuver identification is generated. The results for the three intersections are presented in TABLE 14, TABLE 15 and TABLE 16. For Intersection 1, the algorithm has a 98% accuracy of identifying left turns, a 96% accuracy of identifying right turns, and a 100% accuracy of identifying through maneuvers. For Intersection 2, the algorithm has a 99% accuracy of identifying left turns, 100% accuracy of identifying right turns, and a 99% accuracy of identifying through maneuvers. For Intersection 3, the algorithm has a 100% accuracy of identifying left turns, a 100% accuracy of identifying right turns, and a 100% accuracy of identifying through maneuvers. Overall, this method has high accuracy in maneuver identification. It indicates for some trips, although signature pattern in the gyroscope reading is not observed or the gyroscope threshold value may not be large enough for turn maneuver identification, the maximum and minimum values of the gyroscope global Z component are distinctive enough for accurate maneuver identification.

Further analysis is done to explore the trips from Intersection 1 and Intersection 2 that were inaccurately identified by this clustering technique. At Intersection 1, 3 left turn trips were identified as through maneuvers. Exploration of one such trip is provided here. First, the GPS plot of that particular trip is observed to make sure that the trip was indeed a left turn trip. The OBD speed plot was also done to see whether the vehicle was actually moving. The global Z component of gyroscope readings is plotted then to see if any expected variation is observed due to a turn. The raw gyroscope plot along X, Y and Z axis for that trip is also plotted. The plots are shown in Fig. 52. It becomes evident that, although the vehicle was moving and making a left turn, the gyroscope failed to capture the expected large variation in any of the local axis. The transformed gyroscope reading along global Z axis also does not show any variation. A similar observation was made in other turn trips that were inaccurately identified as through maneuvers.

This indicates that gyroscope might have malfunctioned during the trip. Although gyroscope sensor was capturing data, it was not accurately sensing the angular variations. For the through maneuvers that were inaccurately identified as turns, it was found that the gyroscope showed large variations even though the vehicle did not made any turning maneuver. This is probably due to the local movement of the phone inside the vehicle. Thus, the gyroscope sensor is very sensitive and may sometimes malfunction or capture local movements as well. The accuracy of the maneuver identification algorithm would be more if the gyroscope malfunctioning did not occur or the phone did not experience any local movement during the trip.

TABLE 14 Results for maneuver identification by clustering based on maximum and minimum gyroscope reading for Intersection 1

	Identified				
	Left Turn	Right Turn	Through	%Accuracy	Average silhouette width
Left Turn	142	0	3	98	0.67
Right Turn	0	86	4	96	
Through	0	0	39	100	

TABLE 15 Results for maneuver identification by clustering based on maximum and minimum gyroscope reading for Intersection 2

	Identified				
	Left Turn	Right Turn	Through	%Accuracy	Average silhouette width
Left Turn	123	0	1	99	0.69
Right Turn	0	133	0	100	
Through	0	2	155	99	

TABLE 16 Results for maneuver identification by clustering based on maximum and minimum gyroscope reading for Intersection 3

	Identified				
	Left Turn	Right Turn	Through	%Accuracy	Average silhouette width
Left Turn	9	0	0	100	0.79
Right Turn	0	3	0	100	
Through	0	0	56	100	

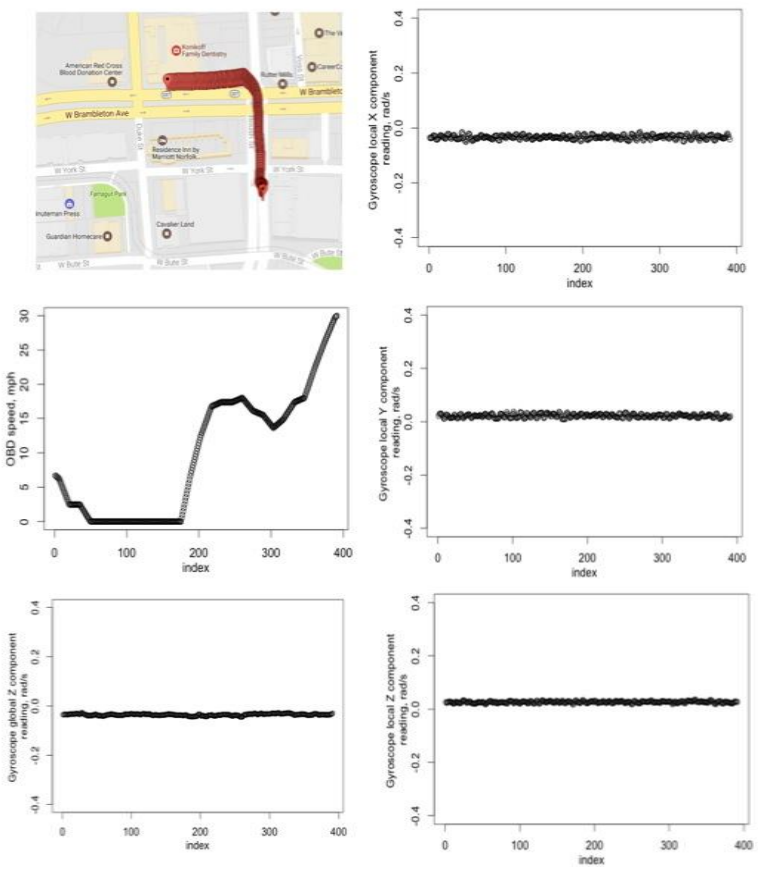

Fig. 52. GPS, OBD speed, raw gyroscope readings and gyroscope global Z component reading plots for a left turn trip identified as through by clustering method.

6.8 Evaluation of Magnetometer Data for Maneuver Identification

In this section, the magnetometer data from the trips are explored to see if they could be utilized to differentiate the three maneuver types. Using the magnetometer data, the orientation of the phone relative to a global coordinate can be calculated. As mentioned in the orientation correction section, gravity acceleration and magnetometer data from a phone's local coordinate system have been employed to identify a phone's orientation angles. A phone's orientation angle φ, θ, and ψ represent the rotation along X, Y, and Z axis of a phone's local coordinate respectively. ψ is also the angle made by the phone relative to Magnetic North. As the phone

remains stationary at its random orientation inside the vehicle, a deviation is expected in one of the orientation angles of the phone if a vehicle makes a turning movement. For example, suppose a vehicle is going North bound. The phone is oriented such that ψ angle of the phone is 55 degrees. If the vehicle makes a left turn at the intersection, then the phone experiences a positive 90 degree turn. As a result, the end orientation of the phone related to North will be 145 degrees. This deviation is expected when the entire time series of the angles of the phone during any particular trip executing a turning maneuver is plotted. To observe, if this assumption is indeed true, the phone angles of all the trips from the 3 intersections were plotted. Fig. 53 shows the plots for orientation angle and Z component of gyroscope readings during 3 different left turn trips. Fig. 53(a) shows the phone's angle related to North, ψ was initially around -150 degree and after the left turn, the angle became around 150 degrees. The gyroscope plot at Fig. 53(b) confirms that the vehicle's left turn maneuver was initiated at around the index number 125 and maneuver was completed at around index number 190. Thus, the change in angle related to North, ψ within that window is due to the turning maneuver. The φ, θ angles remains unchanged during the trip. Fig. 51(c) shows a different pattern in orientation angles, ψ. Initially the angle was -150 degrees. The angles deviated to 150 degrees between index number 25 to 80, then mostly remained at -150 degree until index number 150. They then deviated between -150 degree to 150 degrees until index number 225. After that the orientation angle remained around 100 degrees. However, the gyroscope plot showed in Fig. 53(d) that the turn was executed by the vehicle within index 160 to 250. Thus, there was fluctuations on orientation angle ψ all through the trip, although technically the fluctuation of angle, ψ should only be observed during index 160 to 250, while the vehicle was executing the turn. Fig. 53(e) shows that the orientation angle ψ changes from 150 degree to 100 degree in between index number 350 to 400. The gyroscope

plot (Fig. 53 (f)) confirmed that the vehicle executed the turn in between index number 350 to 400. Overall, a major deviation in orientation angle, ψ is observed if the vehicle executed a turn.

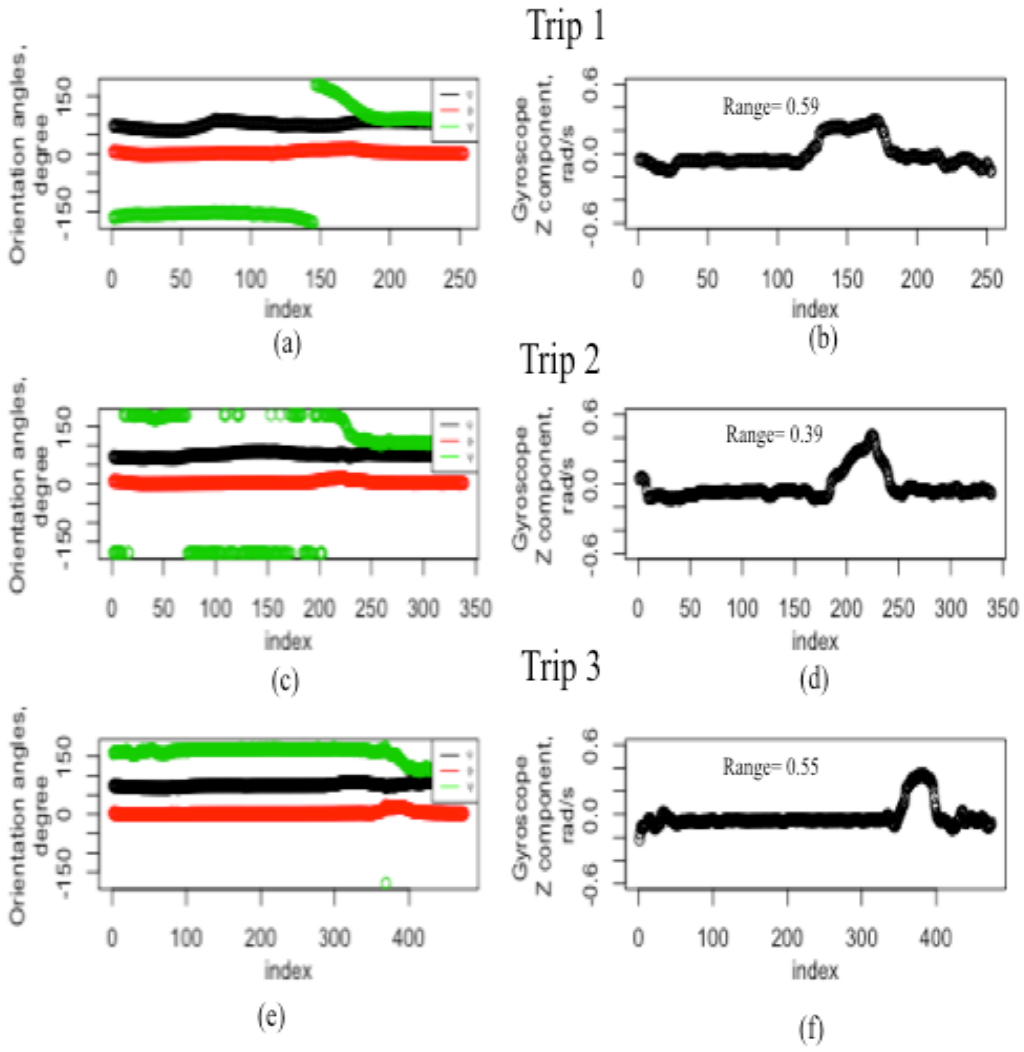

Fig. 53. Phone's orientation angle and Z component of gyroscope plots for 3 different left turn trips.

Next, orientation angles for through maneuvers is explored. It is expected that, for a through going vehicle, the orientation angle, ψ should remain unchanged. For example, if a

vehicle was going North bound initially and the orientation of the phone was such that the phone made a 50 degree angle related to North, then as the vehicle continued moving to North bound through the intersection, the orientation angle will not show any major deviation and remain the same at around 50 degrees. The plots of orientation angles and gyroscope Z component readings for 3 different through trips are presented in Fig. 54. Fig. 54(a) shows that the orientation angles did not show any major deviation as the vehicle made the through maneuver. The gyroscope plot shown in Fig. 54(b) also showed no deviation. However, in the trip shown in Fig. 54(c), the orientation angle, ψ changes from 150 degrees to -150 degrees between index number 150 to 200. The gyroscope plot shown in Fig. 54(d) showed no major variation in gyroscope readings within that period. There were some variations in gyroscope readings at the beginning and end of trip. The range, i.e. the difference between maximum and minimum gyroscope readings, was calculated to be 0.15. Based on the developed maneuver identification algorithm employing gyroscope readings, the range needs to be higher than 0.29 to identify a trip as a turn (Section 6.1.1). Thus the Trip 2, shown in Fig. 54, will be identified as a through trip if the decision is based on gyroscope readings only. But the orientation angles plots showed major deviation during the trip, that falsely implies the trip as a turn trip. A similar observation was made for the Trip 3, shown in Fig. 54. The plots for orientation angle, ψ showed a large deviation and the gyroscope readings also showed variation (Fig. 54(f)). However, the range of gyroscope readings was 0.22. This was less that the required range value 0.29 for identifying a trip as a turn. Thus, it became apparent that the orientation angles of the phone were sometimes giving inaccurate indication of turn maneuvers even though the vehicle was actually executing a through maneuver.

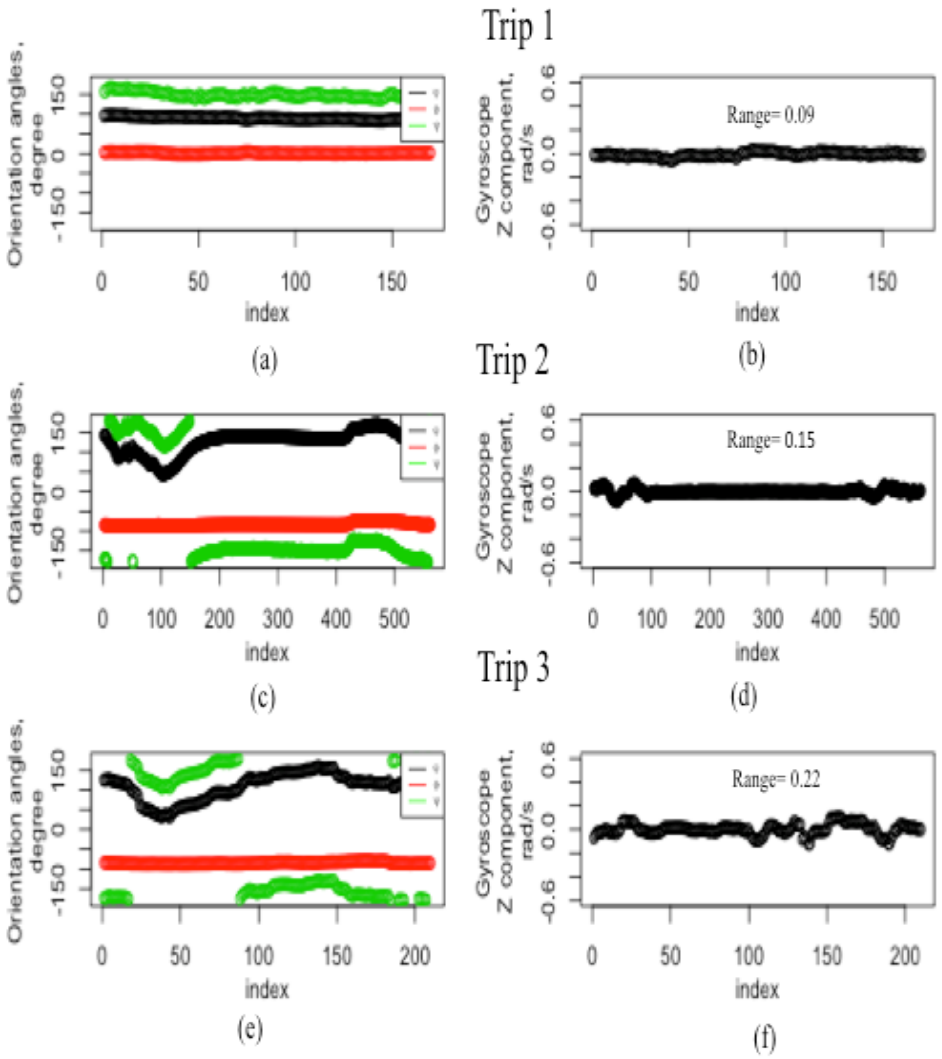

Fig. 54. Phone's orientation angle and Z component of gyroscope plots for 3 different through trips.

Another interesting case was found which is shown in Fig. 55. The orientation angle, ψ was found to be gradually decreasing from 150 degrees to -150 during the through trip and φ angle also showed large deviation during the trip. However, the gyroscope plot did not show any major variation during the trip (range 0.17). All these examples indicate that the orientation angle is not a good parameter for identifying maneuver type. The orientation angles are calculated

from magnetometer data. It seemed from observing the above mentioned cases that the magnetometer data have more vulnerability of being noisy than the gyroscope data. As a result, the orientation angles calculated using magnetometer data can become inaccurate.

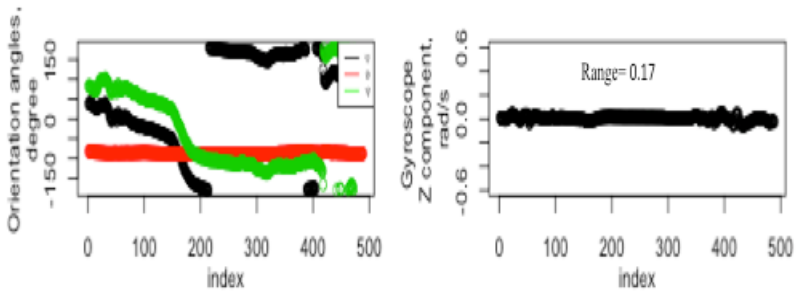

Fig. 55. Phone's orientation angle and Z component of gyroscope plots for a through trip.

6.9 Maneuver Identification Algorithm using Gyroscope Readings and Phone's Orientation Angles

In this section, an algorithm is developed employing the orientation angle of a phone and gyroscope readings to find if the employment of this two parameters together can improve maneuver identification accuracy. The range value of the Z component of transformed gyroscope readings and the difference in value of the orientation angle, ψ for 1st 3 seconds and the last 3 seconds of the trip are employed for the algorithm development. For any particular trip, the range of gyroscope reading, r is calculated as

$$\text{Range, } r = \max(x_1, x_2, x_3, \dots\dots, x_n) - \min(x_1, x_2, x_3, \dots\dots, x_n)\,, \qquad (45)$$

where x is the global Z component of gyroscope reading for a trip with 1 to n data points.

The range of orientation angle, s, for that particular trip is calculated as

$$S = \text{mean}(\psi_1, \psi_2, \psi_3, \dots\dots, \psi_{30}) - \text{mean}(\psi_{n-30}, \psi_{n-29}, \psi_{n-28}, \dots\dots, \psi_n) , \quad (46)$$

where ψ is the orientation angle for a trip with n data points.

For any particular trip, if r is equal or greater than 0.29, or alternately, s is greater than 60 degree, the maneuver executed in the trip is identified as a turn, else identified as a through maneuver. For any particular trip with a maneuver identified as a turn, if the absolute value of the maximum positive gyroscope reading is higher than the absolute value of the maximum negative gyroscope reading then the turn is identified as a left turn, else it is identified as a right turn. The results for this algorithm are discussed in the next section.

6.10 Results for Maneuver Identification using Gyroscope Readings and a Phone's Orientation Angles

Using both the gyroscope reading and orientation angles alternately for maneuver identification yielded poor results for identifying through maneuvers (TABLE 17). For Intersection 1, the algorithm had a 99% accuracy for identifying left turns, a 96% accuracy for identifying right turns, and a 75% accuracy for identifying through maneuvers. Similar results were observed for Intersection 2 (TABLE 18) and Intersection 3 (TABLE 19). The observations from the results are listed below:

- The algorithm using orientation angles and gyroscope readings alternately for maneuver identification showed less accuracy for through maneuver identification.
- The algorithm had high accuracy in turn maneuver identification.
- Overall, the accuracy of the algorithm is less than the algorithm, which used only gyroscope readings to identify maneuver type.

TABLE 17 Results for Intersection 1 by algorithm using gyroscope and orientation angles alternately for maneuver identification

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	143	2	0	99
Right Turn	0	86	4	96
Through	6	4	29	75

TABLE 18 Results for Intersection 2 by algorithm using gyroscope and orientation angles alternately for maneuver identification

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	118	2	4	96
Right Turn	0	133	0	100
Through	13	12	132	85

TABLE 19 Results for Intersection 3 by algorithm using gyroscope and orientation angles alternately for maneuver identification

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	7	0	2	78
Right Turn	0	3	0	100
Through	2	4	50	90

6.11 Results Comparison between Different Methods

For the purpose of comparison between the performances of each method for maneuver identification, a bar chart of each algorithms accuracy for identifying left, right, and through maneuvers is created for all three intersections, separately. The bar charts are presented in Fig. 56, Fig. 57, and Fig. 58, respectively. The findings are listed below:

- Using gyroscope data alone yields more accuracy of maneuver identification than using gyroscope and orientation angles together.
- Clustering by maximum minimum gyroscope reading yields best result.

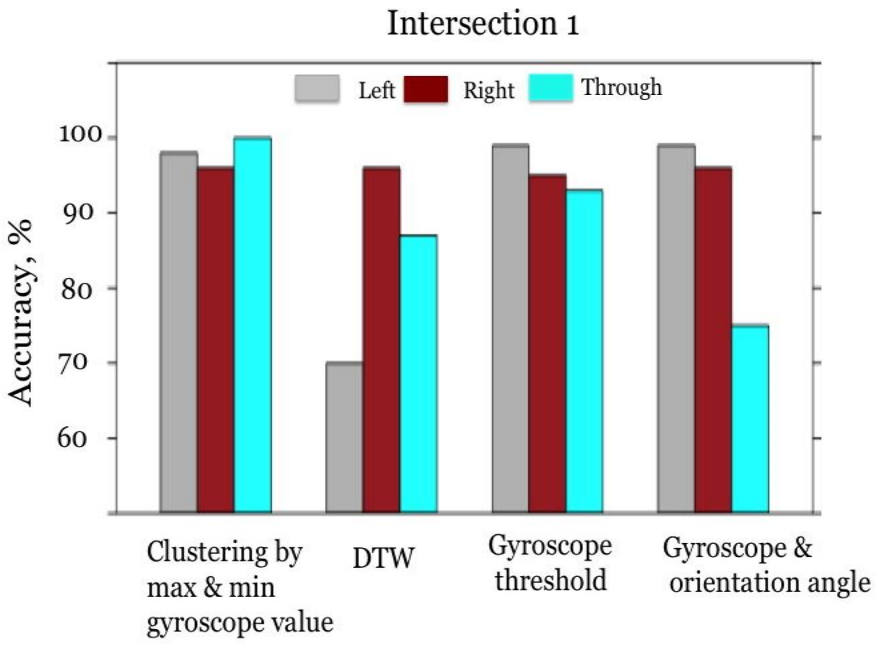

Fig. 56. Maneuver identification result comparison between different methods for trips at Intersection 1.

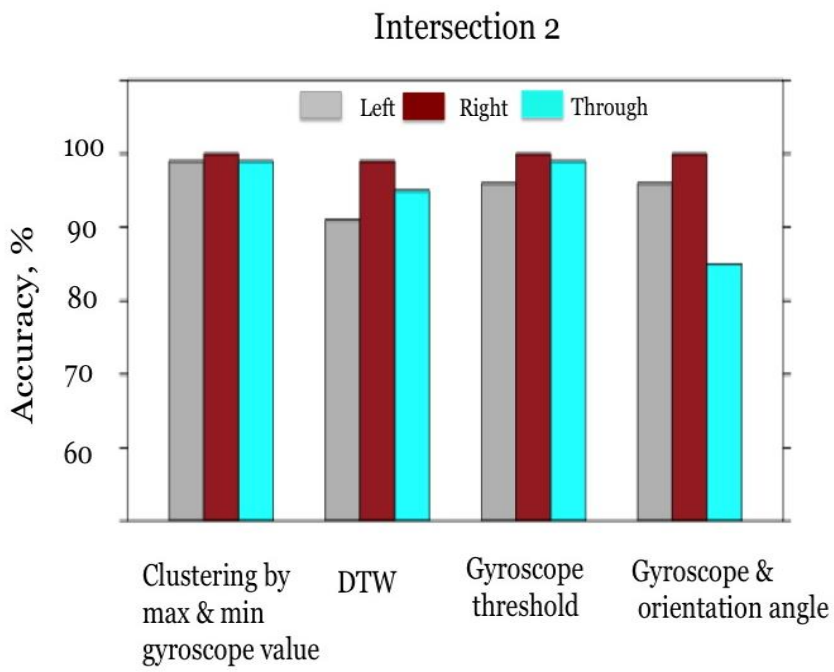

Fig. 57. Maneuver identification result comparison between different methods for trips at Intersection 2.

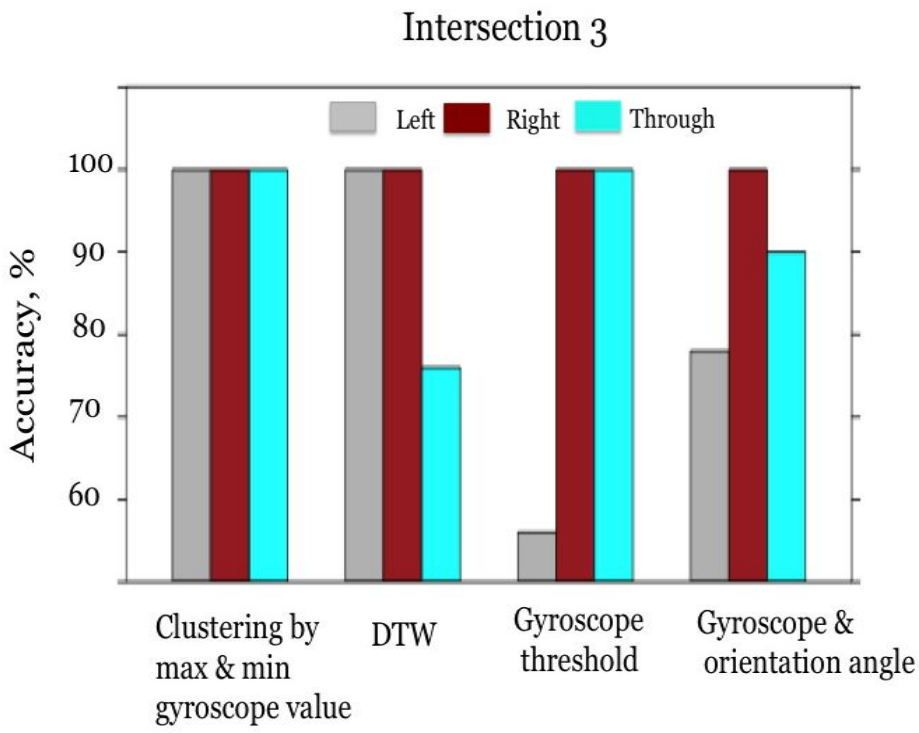

Fig. 58. Maneuver identification result comparison between different methods for trips at Intersection 3.

A hypothesis testing is done to see whether the accuracy results for k-medoids clustering by maximum and minimum gyroscope global Z component readings method is significantly different than other methods [44]. The total numbers of trips from all three intersections are summed up and the number of trips that are correctly identified by each method is calculated. This result can be expresses as a binomial distribution

$$X \sim Binomial(n, p), \tag{47}$$

where

n is the total number of trips,

p is the number trips correctly identified.

For example, among the 756 total trips from all three intersections, 746 trips are correctly identified using the k-medoids clustering method. This can be expressed as a binomial distribution $GT\_L \sim Binomial(756, 746)$. Similarly, for each maneuver, each method's accuracy can be expressed as binomial distribution. Thus, for each maneuver, there are four binomial distributions representing the accuracy of the four methods.

Let $X \sim Binomial(n_1, p_1)$ and $\widehat{p_1} = \frac{p_1}{n_1}$ and $Y \sim Binomial(n_2, p_2)$ and $\widehat{p_2} = \frac{p_2}{n_2}$ represents the results for two different methods for identifying a particular maneuver. To check, if the accuracy results are significantly different, a hypothesis is presented where the null hypothesis states that the accuracy of two methods are same. Thus,

$$null\ hypothesis, H_0: p_1 = p_2, \tag{48}$$

$$alternate\ hypothesis, H_1: p_1 \neq p_2, \tag{49}$$

The score test statistic for this null hypothesis is

$$TS = \frac{\widehat{p_1} - \widehat{p_2}}{\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}, \tag{50}$$

where

$$p = \frac{n_1\widehat{p_1} + n_2\widehat{p_2}}{n_1 + n_2}. \tag{51}$$

For a two tailed test, at 95% confidence level, the critical region value is 1.96. Thus, if absolute value of *TS* is greater than 1.96, the null hypothesis that the two method's accuracy is the same is rejected. Else, the null hypothesis is accepted. This method is applied to do one-on-one pair wise hypothesis testing between k-medoids clustering method with other methods. The results are presented in TABLE 20. It is found that k-medoids clustering by maximum and

minimum method gives significantly different accuracy result than gyroscope threshold method, DTW method and gyroscope and orientation angle method. Thus k-medoids clustering method is the most accurate method for maneuver identification as recognized by this dissertation.

TABLE 20 Hypothesis test results for k-medoids clustering with other three methods. The test statistics value is shown in parenthesis.

	Gyroscope threshold	DTW	Gyroscope and orientation angle
Clustering by max and min gyroscope	Significantly different (2.3)	Significantly different (8.0)	Significantly different (5.7)

CHAPTER 7
DIRECTION IDENTIFICATION

In this section the methodology for identifying the direction of the vehicle through the intersection and results are described.

7.1 Methodology 1: Using OBD Speed, Accelerometer, and Gyroscope Readings for Trip Direction Identification

The phone collects the accelerometer readings while the GoGreen app is running. As the phone is kept in stationary position inside the vehicle, the acceleration experienced by the phone is due to the movement of the vehicle. More specifically, the acceleration captured by the phone is the acceleration of the vehicle. The raw accelerometer readings recorded by the phone is based on the phone's local coordinate system and the axis that shows variation of acceleration values depends on the orientation of the phone inside the vehicle. As an orientation correction technique is applied, the raw accelerometer readings are converted to readings at global coordinate system. At the global coordinate system, the positive X axis points towards East and Positive Y axis points towards North direction. For a four-legged intersection that is aligned to magnetic North-South and East-West direction, the accelerometer readings, with respect to a global coordinate system, will indicate the direction of vehicle's movement. For example, if a vehicle is moving toward in an East bound direction, the transformed accelerometer readings will show variation along X axis, as the X axis of the global coordinate system represents the East-West direction. The Y axis will not show any significant variation as the phone is kept stationary and it only experiences the linear acceleration along the vehicle's direction of movement. Similarly, if a vehicle is moving towards North bound direction, then the transformed accelerometer readings

will show variation along Y axis of the global coordinate system, as the Y axis represents the North-South direction. This concept is used to identify the vehicle's direction in an intersection. As the vehicle can make a turning maneuver at the intersection the direction of the vehicle will change as a result. Therefore, it is critical to select a segment of the accelerometer readings during which the vehicle did not initiate any turning movement.

7.1.1 Description of Algorithm

A smartphone's accelerometer, gyroscope readings and instantaneous speed information from OBD have been utilized to identify the direction of a vehicle. The algorithm contains three-steps. In the first step, whether the vehicle was moving towards the North-South or East-West direction at the beginning of the trip was identified. First, the instantaneous speed information of the vehicle was employed to determine whether the vehicle was at accelerating, decelerating, cruising, or in a stopped state at a particular instance. For any instance, if the speed of the vehicle is higher than its immediate previous instance, then the vehicle is determined to be in an accelerating state (A). If the speed is lower, then the vehicle was in a decelerating state (D). If no change of speed is detected, than the vehicle was in a cruising state (C). Finally, if the instantaneous speed was 0, then the vehicle is determined to be in a stopped state (S) during that instance. Using a rolling window throughout the time series of linear accelerometer readings, 3 seconds time periods were identified where OBD state is consecutive accelerating (A) or decelerating (D).

In the second step, among the selected time periods from the first step, only one time period is required to be selected for further exploration for direction indication. Also, it needs to be ensured that during that time period the vehicle did not initiate any change of direction.

A threshold value is applied on gyroscope readings for the time periods to ensure that the vehicle did not initiate any change of direction within that time period. Thus, the objective was such that a time period needs to be selected in which the linear acceleration shows significant variation but the gyroscope reading are within a specific threshold value. It was considered here empirically that for a consecutive 5 second time period if the median value of the gyroscope reading was less than 0.03 and greater than -0.03 radian/sec, then the vehicle did not initiate any change of direction. Also, as the linear acceleration can show variation in either the X or Y axis, both the X and Y components of the acceleration reading are evaluated to see which time period has the maximum variation and at the same time, the median gyroscope readings are within the threshold. Thus, a time period is found with following objective functions:

Objective function 1:

$$Max(\sum_{j}^{j+29} a_{x_j}), \tag{52}$$

subject to $$\text{median}(G_{x_j}, G_{x_{j+1}}, G_{x_{j+2}}, \dots\dots, G_{x_{j+29}}) < 0.03, \tag{53}$$

$$\text{median}(G_{y_j}, G_{y_{j+1}}, G_{y_{j+2}}, \dots\dots, G_{y_{j+29}}) < 0.03, \tag{54}$$

$$\text{median}(G_{z_j}, G_{z_{j+1}}, G_{z_{j+2}}, \dots\dots, G_{z_{j+29}}) < 0.03, \tag{55}$$

where a_x is the X component of linear acceleration readings,

j is the index of the start of time period during which the vehicle was at consecutive A or D state,

G_x is the X component of gyroscope reading,

G_y is the Y component of gyroscope reading,

G_z is the Z component of gyroscope reading.

Objective function 2:

$$Max(\sum_{j}^{j+29} a_{y_j}), \tag{56}$$

$$\text{subject to} \quad \text{median}(G_{x_j}, G_{x_{j+1}}, G_{x_{j+2}}, \dots\dots, G_{x_{j+29}}) < 0.03, \tag{57}$$

$$median(G_{y_j}, G_{y_{j+1}}, G_{y_{j+2}}, \dots\dots, G_{y_{j+29}}) < 0.03, \tag{58}$$

$$median(G_{z_j}, G_{z_{j+1}}, G_{z_{j+2}}, \dots\dots, G_{z_{j+29}}) < 0.03, \tag{59}$$

where a_y is the Y component of Linear acceleration readings,

j is index of the start of time period during which the vehicle was at consecutive A or D state,

G_x is the X component of gyroscope reading,

G_y is the Y component of gyroscope reading,

G_z is the Z component of gyroscope reading.

From objective function 1, the time period having the maximum variation of the linear acceleration along the X axis is found. From objective function 2, the time period having the maximum variation of the linear acceleration along the Y axis is found. Comparing these two time periods, the period, *T*, that has more variation of linear acceleration is selected.

In the third step, the linear acceleration reading of time period, *T*, is explored to identify the axis of vehicle's movement. The axis associated with the time period, *T*, is the axis showing the (X or Y) most variation. The linear accelerometer readings will show more variation in one particular axis than the other. As the raw linear accelerometer readings have been transformed to a global coordinate system, for any perpendicular intersection, more variation of linear

accelerometer readings along X axis represents movements along an East-West direction, while more variation of linear accelerometer readings along Y axis represents movements along a North-South direction. Thus, observing the principal variation axis during the period, *T* provides information regarding the direction of the vehicle. For example, if more variation of a linear accelerometer is observed along Y axis for the selected 3 seconds time period when the vehicle was at an accelerating or at a decelerating state, then it indicates that the vehicle was going towards in either a North bound or a South bound direction and Y is the principal variation axis. On the other hand, if the maximum variation is observed along X axis, then it indicates that the vehicle was going towards in either an East bound or a West bound direction and X is the principal variation axis. In Figure 4, linear acceleration plots of two trips are presented. The top part of Figure 4 shows linear acceleration plot for Trip 1 in which the vehicle was initially at the North bound leg of intersection and made a through maneuver to go straight through the intersection. The bottom part of Figure 4 shows linear acceleration plot for Trip 2 in which the vehicle was initially at the South bound leg of intersection and went straight through the intersection executing a through maneuver. As expected, both the plots show more variation of linear acceleration along Y axis (blue curve) compared to X axis (red curve), indicating that the vehicle was either going in a North bound or a South bound direction.

In the fourth step of the algorithm, the sign of the linear accelerometer readings along the principal variation axis identified in step 3 for the 3 second time period is observed. As in the global coordinate system, a positive Y axis indicates North, thus, for a vehicle in an accelerating state, if the linear acceleration readings along the Y axis are positive this indicates movement in a North bound direction; else, if the acceleration readings along the Y axis are negative this indicates movement in a South bound direction. Similarly, as a positive X axis is East in a global

coordinate system, the positive acceleration readings along the X axis for a vehicle at accelerating state indicates movement in an East bound direction and negative readings indicate movement in a West bound direction.

For a vehicle in a decelerating state, negative linear acceleration readings along the Y axis indicate movement in a North bound direction and positive acceleration readings for a vehicle in a decelerating state indicate movement in a South bound direction. Similarly, as a positive X axis is East in a global coordinate system, the negative accelerometer readings along X axis for a vehicle in a decelerating state indicate movement in an East bound direction and positive readings indicate movement in a West bound direction. The logic is represented in TABLE 21. Thus, the direction of the vehicle can be pinpointed. The trips shown in Fig. 59 and Fig. 60 were in accelerating state for consecutive 3 second time period. The accelerometer readings along Y axis for that time period for Trip 1 were positive, thus indicating that the vehicle was moving towards Northbound direction (Fig. 59). For Trip 2, the accelerometer readings along Y axis for that time period were negative, indicating that the vehicle was moving towards Southbound (as shown in Fig. 60).

TABLE 21 The logic of direction identification based on linear acceleration readings

Major Variation Axis of Linear Acceleration	Vehicle's State	Sign of Linear Acceleration readings	Trip Direction
Y axis	Accelerating (Decelerating)	+ (-)	Northbound
	Decelerating (Accelerating)	+ (-)	Southbound
X axis	Accelerating (Decelerating)	+ (-)	Eastbound
	Decelerating (Accelerating)	+ (-)	Westbound

The fifth step determines whether the direction identified observing the linear accelerations from the selected time period indicates the initial direction of the vehicle (before executing a maneuver at an intersection) or a final direction of the vehicle (after executing a maneuver at an intersection). Using the maneuver identification algorithm, the maneuver type of the trip is already known. For trips that have been identified as through by the maneuver identification algorithm, the initial and final direction remains same. But for trips that have been identified as turn by the maneuver identification algorithm, the direction changes due to the maneuver. Comparing the index of start time of period *T,* and the index of maximum (minimum) Z component of gyroscope reading of left turn (right turn) will indicate whether the selected time period, *T*, is from before or after the vehicle executed the turning maneuver. For example, for a trip that has been identified as a left turn by the maneuver identification algorithm, the index of the maximum value of Z component of gyroscope reading is 150. The index of the starting of time period, *T*, found by the second step of the direction identification methodology is 20. Thus, it indicates that the selected time period, *T*, is before the vehicle executed the turning maneuver.

Therefore, the direction that is identified by further exploring the linear accelerations during time period, *T*, is the initial direction of the vehicle.

In the sixth step, the direction information from step 4, information regarding whether the direction is the initial or final direction of the vehicle (obtained from step 5) and maneuver type information obtained using the maneuver identification algorithm is merged to identify the vehicle's trajectory. For example, if the maneuver identification for a trip revealed that the vehicle made a through maneuver and the fourth step of the direction identification algorithm revealed that the vehicle was traveling in a Northbound direction, and the fifth step indicated that the direction was the initial direction, then merging these three information points provides the total trip direction of that trip along that particular intersection, which is North bound – North bound. This means that the vehicle was going in a North bound direction initially before the intersection and then executed a through maneuver at the intersection to go straight in a North bound direction.

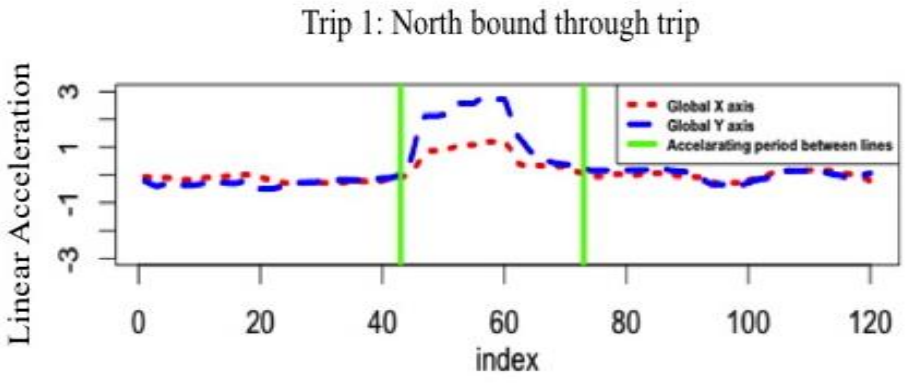

Fig. 59. Linear acceleration plot of a trip going North bound through the intersection.

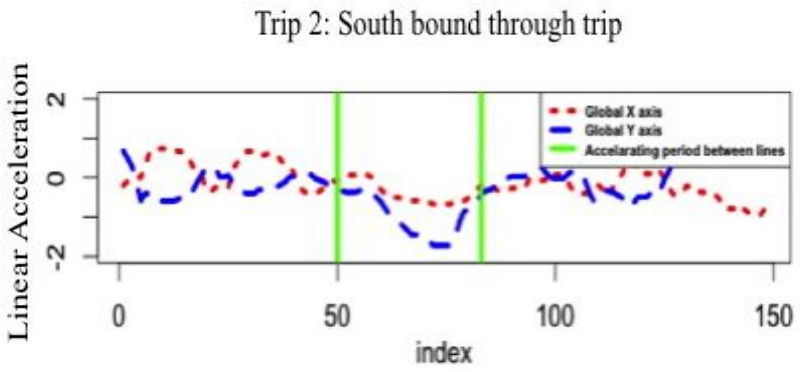

Fig. 60. Linear acceleration plot for Trip 2 going South bound direction through the intersection.

7.1.2 Results

At first, to assess the accuracy of the algorithm, the above-described algorithm is applied to trip data collected at a control condition. During the trip, a phone was held stationary in a phone holder. In addition, no lane changes were made during the trip at the intersection. This is done to ensure that the variation in acceleration and gyroscope data is experienced only because of the vehicle's motion along the direction of movement. The results are presented in TABLE 22. The results indicate that the algorithm correctly identified all 19 of the through trips and correctly identified 15 of 16 turn trips.

TABLE 22 Results of direction identification algorithm for trips at control condition

Maneuver type	#Trips	Direction correctly identified	%Accuracy
Through	19	19	100
Turn	16	15	94

While the phone collected trip data from the 3 intersections chosen for analysis, the phone was not always held in a phone holder, but rather may be placed loosely in a cup holder.

This might result into some movement of phone itself due to the movement of the vehicle. Also a driver may or may not execute lane changes during the trip. Thus, it is imperative to see how the algorithm performed on trip data collected from the three 4-legged intersections.

Intersection 1 had total 274 numbers of trips. Among those, 39 trips were going through the intersection and 235 trips were trips that made turning maneuver at the intersection. The direction identification algorithm has multiple steps and requires the trip data to meet certain requirements, i.e., the OBD state need to be in a consecutive 3 second acceleration or deceleration condition and, at the same time, the median value of the gyroscope readings during that 3 second period need to be within a specific threshold (<0.03 rad/s). Little of the trip data did not meet those requirements and thus did not get excluded from direction identification. The results of the direction identification algorithm are shown in TABLE 23. Of the 39 through trips, 29 trips met the consecutive accelerating or decelerating state and gyroscope threshold value requirements. Among, them the direction of 24 trips were identified correctly. The accuracy was 83%. Of the 235 turn trips, 178 trips met the consecutive accelerating or decelerating state and gyroscope threshold value requirements. Among them, direction of 131 trips was correctly identified with accuracy being 74%.

TABLE 23 Direction identification results for Intersection 1

Maneuver Type	Final # of trips for trip direction identification	Direction correctly identified	%Accuracy
Through	29	24	83
Turn	178	131	74

As there are some trip data losses due to the algorithm requirements, a further breakdown is presented to see in which step trip data is lost. The breakdown is presented in TABLE 24. Among the 39 through trips, 8 trips did not have the 3-second consecutive accelerating or decelerating state and thus did not pass the first step of the direction identification algorithm. Among the rest of the 31 trips, 2 trips had gyroscope readings outside the threshold value and thus did not pass the second step of the direction identification algorithm. Of the total 39 through trips, the rest of the 29 trips got qualified to go through steps 3 to 6 of the algorithm for direction identification. Among these 29 trips, the maneuver identification algorithm inaccurately identified 3 of the trips. Among the 235 turn trips, 4 trips did not have the 3-second consecutive accelerating or decelerating state and thus did not pass the first step of the direction identification algorithm. Fifty three trips had gyroscope readings outside the threshold value and thus did not pass the second step of the direction identification algorithm. Thus, of the total 235 turn trips, 178 trips got qualified to go through steps 3 to 6 of the algorithm for direction identification. Among these 178 trips, the maneuver identification algorithm inaccurately identified 7 of the trips.

TABLE 24 Breakdown of trip data loss for Intersection 1

Maneuver Type	Actual # of trips	No consecutive 3 seconds accelerating/ decelerating state found	Gyroscope outside of defined range	Final # of trips for trip direction identification	Maneuver correctly identified	Direction correctly identified
Through	39	8	2	29	26	24
Turn	235	4	53	178	171	131

The results for Intersection 2 and the breakdown of data loss are presented in TABLE 25 and TABLE 26. Of the 157 through trips, 120 trips met the consecutive accelerating or decelerating state and gyroscope threshold value requirements. Among, them the direction of 95 trips were identified correctly. The accuracy was 79%. Of the 257 turn trips, 242 trips met the consecutive accelerating or decelerating state and gyroscope threshold value requirements. Among them, the direction of 183 trips was correctly identified with an accuracy of 76%.

TABLE 25 Direction identification results for Intersection 2

Maneuver Type	Final # of trips for trip direction identification	Direction correctly identified	% Accuracy
Through	120	95	79
Turn	242	183	76

TABLE 26 Breakdown of trip data loss for Intersection 2

Maneuver type	Actual # of trips	No consecutive 3 seconds accelerating/ decelerating state found	Gyroscope outside of defined range	Final # of trips for trip direction identification	Maneuver correctly identified	Direction correctly identified
Through	157	19	18	120	117	95
Turn	257	0	15	242	236	183

The results for Intersection 3 are presented in TABLE 27. Of the 56 through trips, 49 trips met the consecutive accelerating or decelerating state and gyroscope threshold value requirements. Among, them the direction of 43 trips were identified correctly. The accuracy was 88%. Of the 12 turn trips, 11 trips met the consecutive accelerating or decelerating state and gyroscope threshold value requirements. Among them, direction of 5 trips were correctly identified with an accuracy of 46%. The reason for the low accuracy of turn identification is revealed when breakdown of trip data loss is performed (TABLE 28). Among the 11 turn trips that met the requirement of the direction identification algorithm, the maneuver, the maneuver identification algorithm inaccurately identified 4 of them. As mentioned in Section 6, the gyroscope was not able to capture strong rotation rate while the vehicle was making turn, the maneuver identification algorithm was not able to identify the turn maneuvers correctly. As a result the direction identification also yielded wrong results for those trips.

TABLE 27 Direction identification results for Intersection 3

Maneuver type	Final # of trips for trip direction identification	Direction correctly identified	%Accuracy
Through	49	43	88
Turn	11	5	46

TABLE 28 Breakdown of trip data loss for Intersection 3

Maneuver type	Actual # of trips	No consecutive 3 seconds accelerating/ decelerating state found	Gyroscope outside of defined range	Final # of trips for trip direction identification	Maneuver correctly identified	Direction correctly identified
Through	56	3	4	49	49	43
Turn	12	0	1	11	7	5

This work demonstrates the ability to identify a vehicle's trajectory without relying on GPS information. During the analysis, some trips were observed in which inaccurate trajectory information of the vehicle is found if only GPS coordinates are considered. Due to errors incurred with GPS coordinates, the position of a vehicle found by GPS contained deviation from its actual position. Plotting the GPS coordinates on a map showed a large deviation from the vehicle's actual trajectory. In such cases, the above-mentioned algorithm was able to correctly identify the trajectory of the vehicle. One such scenario is presented in Fig. 61. The actual trajectory of the vehicle is marked by green line whereas red balloons show the GPS plot of the trip. The GPS plots show large deviation from the actual trajectory. In reality, the vehicle was initially at West leg of intersection going in an East bound direction and made a right turn at the intersection to go directly towards South bound direction. The driver of the vehicle confirmed this trajectory information. The algorithms for maneuver and trip direction identification was able to accurately identify the trajectory. This signifies the importance of decreasing sole dependence on GPS and utilizing alternate sensors for trajectory identification.

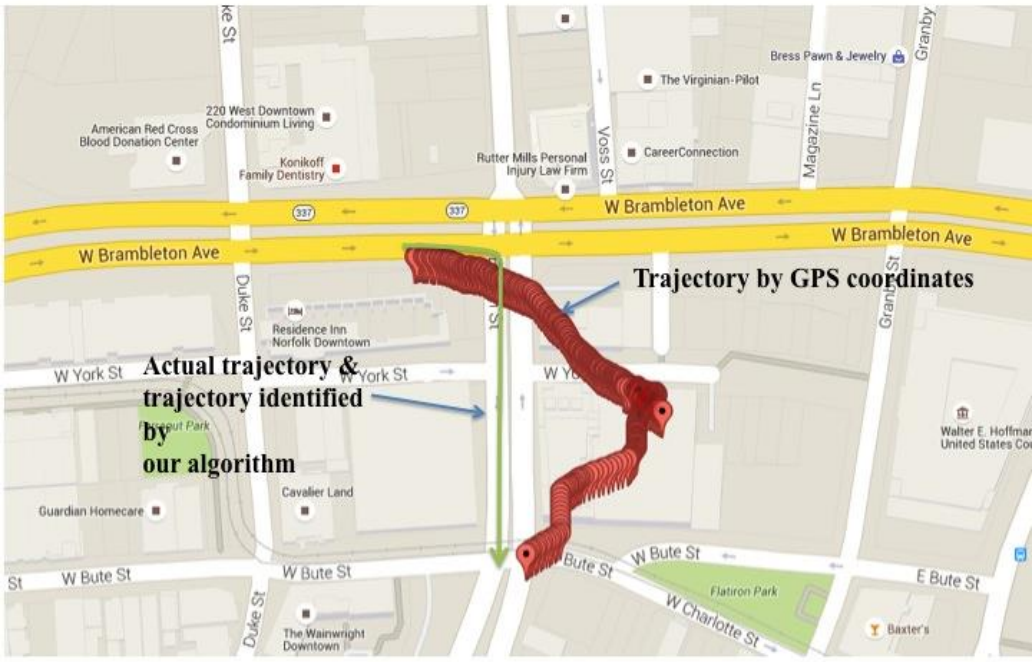

Fig. 61. GPS plot, actual trajectory and trajectory identified by our algorithm for a trip.

The algorithm has some drawbacks. The algorithm has some data loss issues as discussed above. Also the algorithm requires that a phone remain in stationary position inside a vehicle. If the phone moves inside the vehicle, then the direction identified using the accelerometer values would not be able to be identified accurately. In addition, the overall accuracy depends on accuracy of maneuver identification and direction identification as well. If the maneuver is incorrectly identified, the direction of the vehicle identified will be incorrect as well. Finally, if lane changes occurred during a trip, acceleration due to lane changes will affect the result. As for the trips in control condition, no lane changes occurred, the results for direction identification for through trips were 100% and turn trips were 94%. However, for trip data collected from the 3 intersections, lane change information was not available. Thus, it was possible that drivers might

have changed lanes during the trip at the intersection. In such cases, the direction identification will be less accurate.

7.2 Method 2: Using Magnetometer Data to Find Orientation of Vehicle

This method utilizes 3 coordinate systems: a phone's local coordinate system, a vehicle's coordinate system, and a global coordinate system. The phone collects the inertial sensor data in a phone's local coordinate system. These raw data are then converted to a vehicle's coordinate system. Then, using the transformed magnetometer data and gravity data, the orientation of the vehicle related to a global coordinate system is obtained. For transformation from a phone's coordinate system to a vehicle's coordinate system, a method developed by Wang et al. [14] is employed. A phone's local coordinate system and a vehicle's coordinate system is defined as shown in Fig. 62.

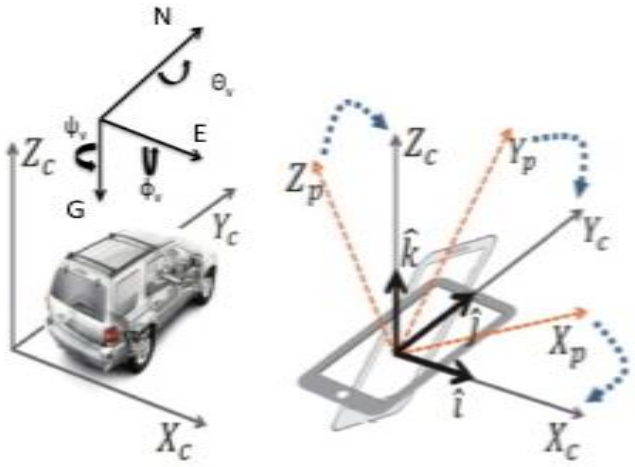

Fig. 62. A phone's coordinate system, a vehicle's coordinate system, and a global coordinate system [14].

Three unit coordinate vectors $\hat{\imath}, \hat{\jmath}$ and $\hat{k}$ is defined as for the X_c, Y_c, and Z_c axis, respectively. The rotation matrix R is defined as

$$R=\begin{pmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{pmatrix} . \quad (60)$$

The raw gravity acceleration readings and the linear acceleration readings (when the vehicle was decelerating) was employed to calculate the rotation matrix R [14]. The raw sensor readings, S_p are aligned from phone's local coordinate to vehicle's coordinate by applying R on S_p. The transformed readings, S_v at the vehicle's coordinates is thus found by following [14]:

$$S_c = S_p \times R. \quad (61)$$

For this dissertation, using this method, the magnetometer data are transformed from a phone's local coordinate to a vehicle's coordinates. The transformed magnetometer data are then used to determine the orientation of the vehicle with the following equations:

$$\text{Pitch}(\varphi_c) = \tan^{-1}\left(\frac{g_{Yc}}{g_{Zc}}\right), \quad (62)$$

$$\text{Roll}\ (\theta_c) = \tan^{-1}\left(\frac{-g_{Xc}}{g_{Yc}\sin\varphi_c + g_{Zc}\cos\varphi_c}\right), \quad (63)$$

$$\text{Yaw}\ (\psi_c) = \tan^{-1}\left(\frac{m_{Xc}\cos\theta_c + m_{Yc}\sin\theta_c\sin\varphi_c + m_{Zc}\sin\varphi_c\cos\varphi_c}{m_{Zc}\sin\varphi_c - m_{Yc}\cos\varphi_c}\right). \quad (64)$$

The angle, ψ_c is the angle that vehicle's forward direction makes related to global North. The direction is found by the logic shown in TABLE 29. Thus, the vehicle's direction is expected to be known using this algorithm.

TABLE 29 Vehicle's angle related to North and direction of movement

Angle related to North, ψ_c	Direction
-45 to 45	Northbound
>45 to 135	Eastbound
>135 to 180 or -180 to <-135	Southbound
< -45 to -135	Westbound

7.2.1 Results

The above described method is applied on a data set consisting of 35 trips. The location is the intersection between Powhatan Avenue and 49th Street in Norfolk, Virginia. One driver made all 35 trips. Before the start of each trip, the magnetometer is calibrated in order to remove any bias. The results are indicated in TABLE 30. Of the total 35 trips, 19 trips were through and 16 trips were turn trips. Of the 19 through trips, the algorithm correctly identified direction of 9 trips resulting in a 47% accuracy. Of the 16 trips, the algorithm correctly identified the direction of 11 trips resulting in a 67% accuracy. Although the magnetometer was calibrated before the beginning of each trip and the phone was not touched during the trip, the results indicated that due to the sensitive nature of the magnetometer, the direction found by using magnetometer values are not always accurate.

TABLE 30 Direction identification results using vehicle orientation related to North

Maneuver Type	Trips #	Direction correctly identified	%Accuracy
Through	19	9	47
Turn	16	11	69

CHAPTER 8

TRAJECTORY IDENTIFICATION

In this chapter, the method for trajectory identification of a vehicle is explained.

8.1 Method for Trajectory Identification

The gyroscope readings, linear acceleration readings and OBD speed values are utilized to identify a vehicle's trajectory in an urban grid network. The method is illustrated through an example. Let's suppose a vehicle's origin and destination are known (as shown in Fig. 63). The route taken by the vehicle to traverse from the origin to the destination is unknown. The traffic network is a grid-based urban network. For going from the origin, O to destination, D, there are multiple routes (as shown in Fig. 64). It is imperative to identify the route taken by the vehicle as the route information can provide insights for vehicle counts in different routes. The smartphone collected inertial sensor data and OBD speed information have been collected for that particular vehicle during its trip from the origin to destination. The trajectory identification method has 3 steps: maneuver identification, distance estimation, and trajectory identification.

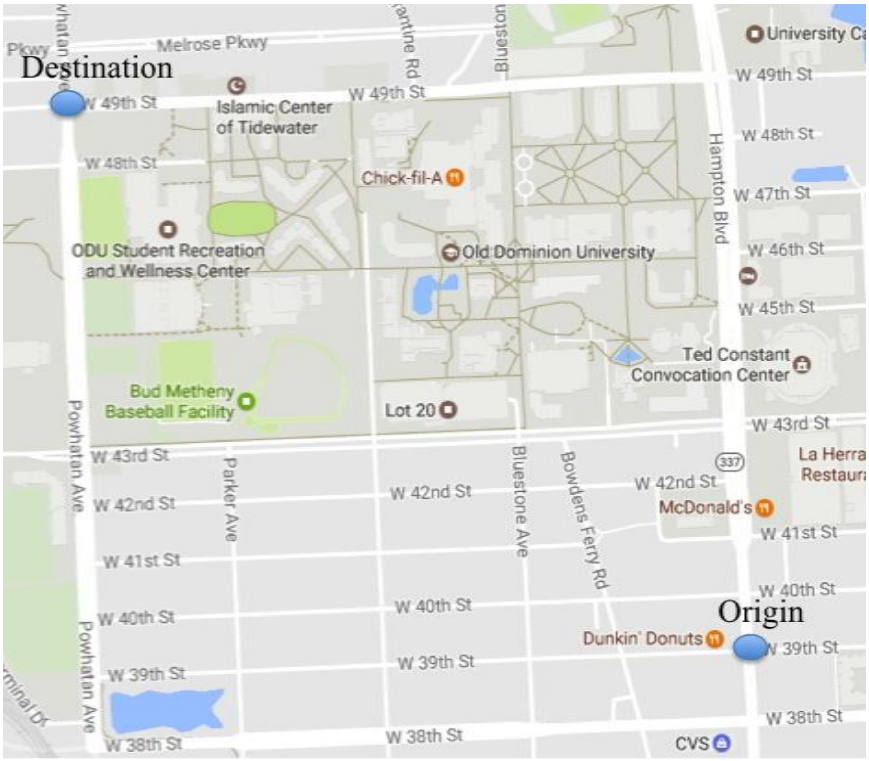

Fig. 63. Map showing the origin and destination of the vehicle considered.

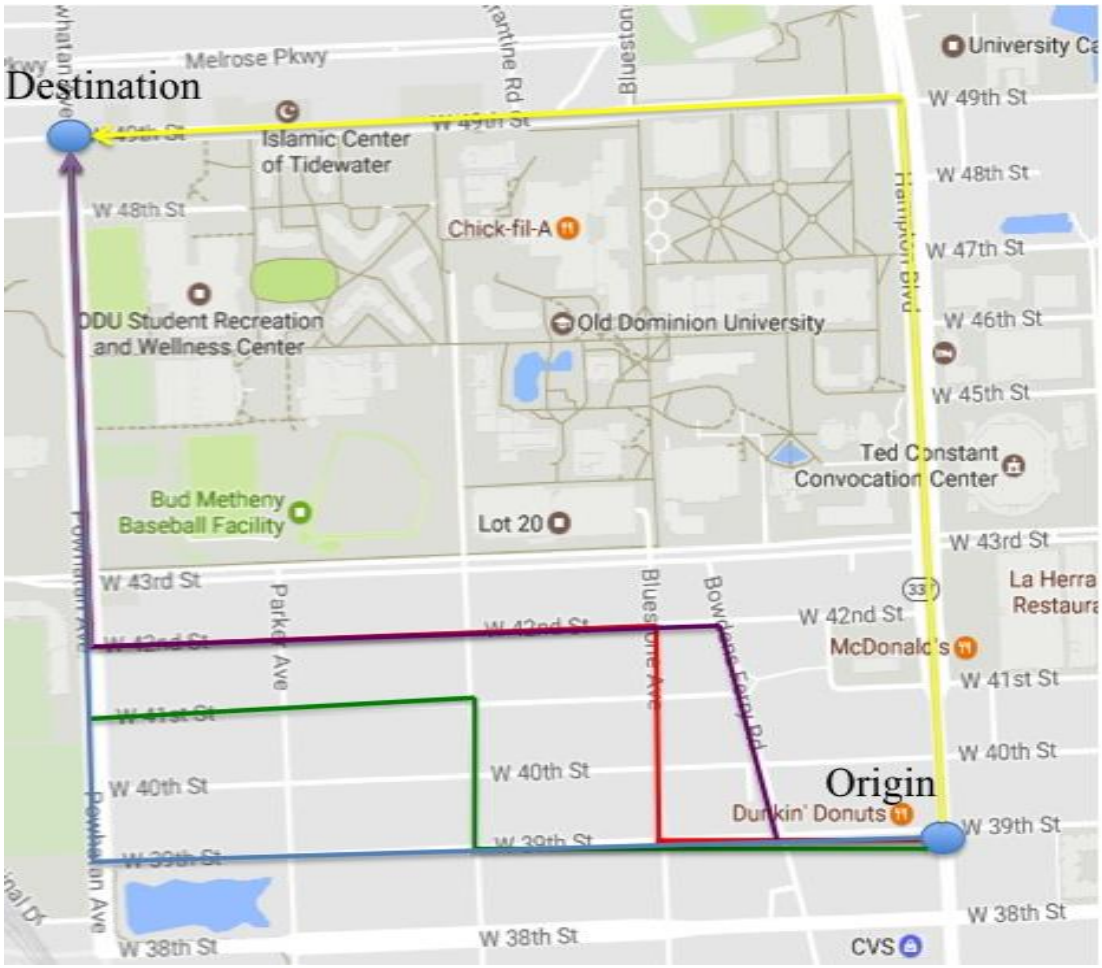

Fig. 64. Few of the possible routes from origin to destination.

The maneuvers executed by the vehicle during the trip are identified by the gyroscope readings. The noise removal and orientation correction techniques were applied to the raw gyroscope readings. The transformed gyroscope along global Z axis is plotted and are shown in Fig. 65. As described in the maneuver identification chapter, the signature pattern for left turn and right turn is observed in the gyroscope plots for vehicle's making turning maneuvers. For this particular trip, in the example, 3 turning maneuvers were observed during the trip. Among them, 2 are right turns and 1 is a left turn maneuver.

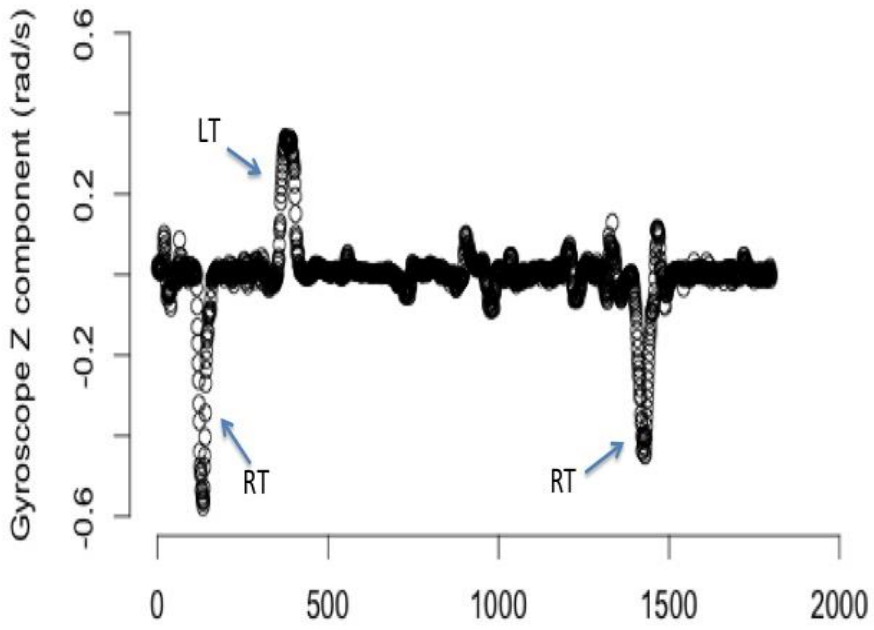

Fig. 65. Plot for gyroscope readings along the trip.

The sequences of the turning maneuvers are as follows:

Start…RT....LT….RT....Destination.

The sequences can be numbered as follows:

Start…RT_1…LT_1….RT_2...Destination

where, RT_1 and RT_2 indicate a first and ^{second} right turn, respectively.

Similarly, LT_1 indicates a first left turn.

The initial direction of the vehicle is identified employing the direction identification method described in Section 7. The initial direction is identified as West bound. Next, using the sequences of turns and identified initial direction of the vehicle, map matching is executed from the origin to the destination. Observing the map, it becomes clear that there are multiple routes through which if the vehicle travels, then the sequence of gyroscope patterns will be as follows:

Start… RT_1…LT_1….RT_2…Destination.

Fig. 66 shows three different routes from the origin to destination, which will have the above-mentioned sequence of gyroscope readings pattern. Thus, it is important to find a way to differentiate between the routes that will generate similar sequences. The OBD speed readings are employed for the purpose.

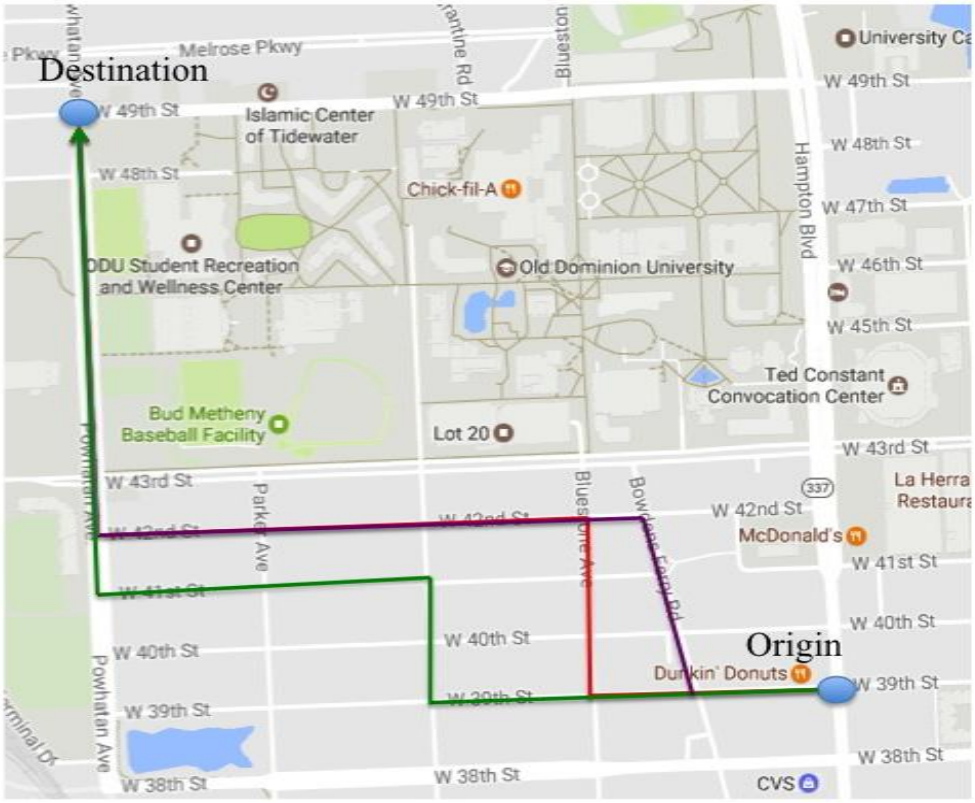

Fig. 66. Multiple routes that will show similar gyroscope signature.

The distance is estimated using OBD speed readings. The index of number of the highest/lowest gyroscope readings for each left turn or right turn signature pattern observed is stored in I.

$I=[I_{RT1}, I_{LT1}, I_{RT2}]$

The latitude longitude of the origin of the trip and destination of the trip is known.

The distance for each instance is measured as:

$$D_i = v_i * dt, \tag{65}$$

where i = index,

v_i= OBD speed at ith index,

dt = 0.1.

Thus, the total distance is measured from origin to the first right turn as following:

$$D_{RT1} = \sum_{I=0}^{I_{RT1}} D_I \ . \tag{66}$$

Similarly, the distance is measured from origin to the first RT, the two consecutive turning maneuvers, and the last turning maneuver to the destination point. The estimated distance by OBD speed and the actual distance measured using latitude and longitude is shown in Fig. 67. It is observed that the estimated distance may vary from 0 to 20 meter from actual distance. Thus, an error range equals actual distance, $D_{actual} \pm \varepsilon$ and is defined where ε is error valued as 20 meters. For trajectory identification via route matching, if a path is found to have an actual distance, D_{actual} and the estimated distance is within 20 meter of D_{actual}, then that path is selected to be the trajectory of the vehicle.

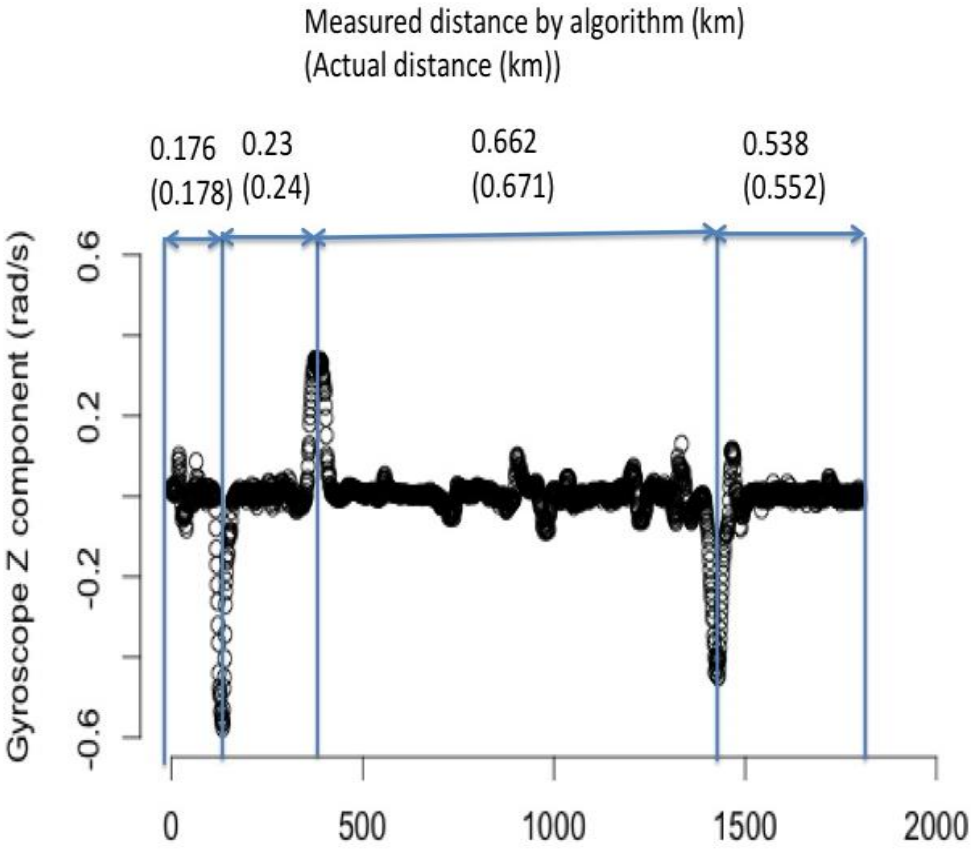

Fig. 67. Estimated distance and actual distance between turns.

Next, an algorithm is derived to identify trajectory of the vehicle via route matching. The algorithm is described through an example, where a node-node representation of the network a vehicle had traveled through is presented in Fig. 68. Each node represents an intersection. There are 12 nodes and the vehicle started from node s=1 and ended at node, t = 12. The vehicle originally took the path 1-2-3-7-10-11-12.

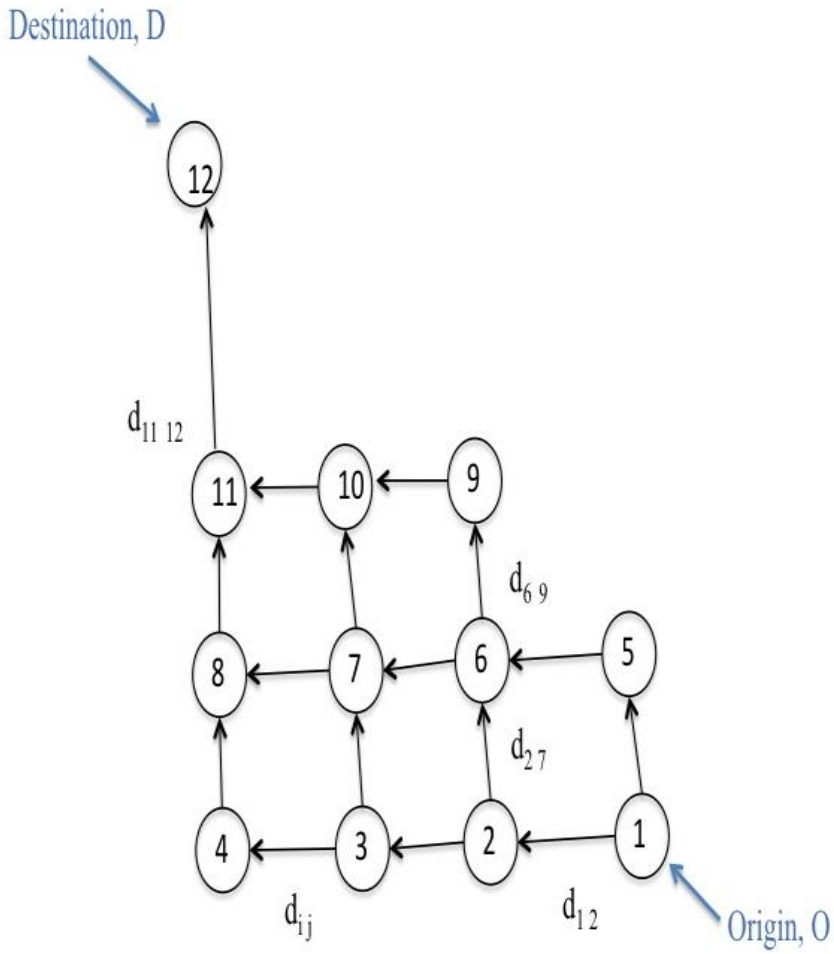

Fig. 68. Node-node representation of an example network.

The relative direction of each node with respect to its adjacent node is presented in a matrix form and is shown in Fig. 69. The distance between each adjacent node is presented in

another matrix and is shown in Fig. 70. Based on the initial direction that the vehicle enters a particular node and the maneuver executed at that node, the final direction the vehicle proceeded to can be identified. For example, if the vehicle entered the node with direction being West bound and made a right turn at that node, then the final direction of the vehicle will be North bound. Thus, the vehicle would proceed to the node that is at North bound of the initial node. This direction identification logic is presented in TABLE 31.

Node	Direction			
	N	S	E	W
1	5			2
2	6		1	3
3	7		2	4
4	8		3	
5		1		6
6	9	2	5	7
7	10	3	6	8
8	11	4	7	
9		6		10
10		7	9	11
11	12	8	10	
12		11		

W= West, E=East, N= North, S= South

Fig. 69. Matrix representing relative direction of node with its adjacent node.

Node	1	2	3	4	5	6	7	8	9	10	11	12
1		d_{1-2}			d_{1-5}							
2	d_{2-1}		d_{2-3}			d_{2-6}						
3		d_{3-2}		d_{3-4}			d_{3-7}					
4			d_{4-3}					d_{4-8}				
5	d_{5-1}					d_{5-6}						
6		d_{6-2}			d_{6-5}		d_{6-7}		d_{6-9}			
7			d_{7-3}			d_{7-6}		d_{7-8}		d_{7-10}		
8				d_{8-4}			d_{8-7}				d_{8-11}	
9						d_{9-6}				d_{9-10}		
10							d_{10-7}		d_{10-9}		d_{10-11}	
11								d_{11-8}		d_{11-10}		d_{11-12}
12											d_{12-11}	

Fig. 70. Matrix representing distance of node with its adjacent node.

TABLE 31 Logic for direction finding

Initial Direction	Turn at Intersection	Outbound Direction
West bound	Right turn	North bound
	Left turn	South bound
East bound	Right turn	South bound
	Left turn	North bound
North bound	Right turn	East bound
	Left turn	West bound
South bound	Right turn	West bound
	Left turn	East bound

For this example, the following information is known about the vehicle's trajectory:

Start node, s=1

Destination node, t =12

Initial direction, ID: West bound

Turn type, T =[T1, T2, T3]

T can take value 0 or 1, where 0 represents a right turn and 1 represents a left turn.

Distance, D=[D_{ST1}, D_{T1T2}, D_{T2T3}, D_{T3D}]

The algorithm for trajectory matching is as follows:

1.3 Go to source node and find which node is at initial direction of source node. For this example, the node West bound to start node 1 is node 2 (see the matrix shown in Fig. 69). Go to that node. Example: go to 2.

2.3 Check distance $d_{1\ 2}$

3.3 If $d_{1\text{-}2} - \varepsilon < D_{ST1} < d_{1\ 2} + \varepsilon$, then path 1-2. ε is set to be 20 meter empirically.

4.3 If not, then check which node is at ID of 2. In the example, node 3 is at a West bound direction of node 2. Thus go to node 3.

5.3 Check distance $d_{2\ 3}$

6.3 If $d_{1\ 2}+d_{2\ 3} - \varepsilon < D_{ST1} < d_{1\ 2}+d_{2\ 3} - \varepsilon$, then path 1-2-3. If not, then check which node is at ID of 3.

These steps will be repeated until the distance traveled from source to a node is found to be within the error margin. Once the path is found, then check the turn type at T. In this example, T1= right turn. Using the logic presented in TABLE 31, find the outbound direction of the vehicle. For this example, the outbound direction is North bound. Thus, go to the node that is northbound to the current node. In this example, the node is 7.

7.3 Check distance $d_{3\ 7}$.

8.3 If $d_{1\ 2}+d_{2\ 3} + d_{3\ 7} - \varepsilon < D_{ST1} + D_{T1T2} < d_{1\ 2}+d_{2\ 3} + d_{3\ 7} + \varepsilon$, then path 1-2-3-7. If not, then check which node is at North bound of 7.

These steps will be repeated until the total path from source to destination node is found.

The algorithm can be presented in the following flow chart (Fig. 71).

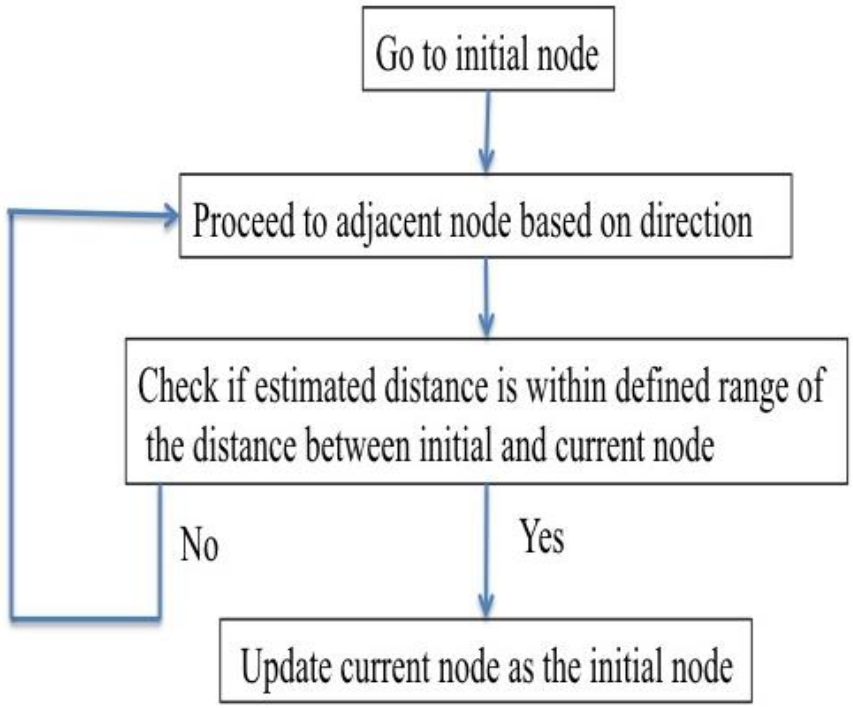

Fig. 71. Flow chart of trajectory identification algorithm.

8.2 Results and Discussion

This algorithm has been applied to a small data set containing 4 trips. The trips have the same origin and destination (shown in Fig. 63) but the route for each was different. Two different drivers drove for these trips. The algorithm was able to correctly identify vehicle's trajectory for each of the trip. The accuracy of the algorithm, however, also depends on the accuracy of the maneuver identification and the direction identification algorithm. If the maneuvers and initial direction of the trip are not accurately identified, then the identified trajectory of the vehicle will be inaccurate. For future work, a correctional method will be developed to correct the trajectory

in cases if the trajectory was identified incorrectly due to incorrect maneuver or direction identification.

CHAPTER 9

CONCLUSIONS

The summary, benefits, and potential applications, contributions and future works of this dissertation are presented in this chapter.

9.1 Summary

This dissertation presents methods to detect a vehicle's trajectory at an intersection by using smartphone-collected inertial sensor data and speed data collected from an On Board Diagnostic device. Additionally, a method for identifying a vehicle's trajectory at an urban network is also demonstrated. The main highlight of the work is that the trajectory identification is done without applying information obtained from GPS. As GPS has several limitations, such as, connectivity issues, high power consumption, low precision, vulnerability to spoofing attacks etc., the objective of this dissertation was to minimize GPS dependency and supplement information from other sensors to identify a vehicle's trajectory. Due to the inherent noise in the accelerometer data and the variation in phone's orientation, detecting vehicle trajectory was found to be nontrivial. Thus, data preprocessing was an important step before developing any algorithm for trajectory identification. In this dissertation, a two-step noise removal technique is demonstrated. In the first step, a threshold value on gravity acceleration data was applied to remove noise from raw data due to sudden movement of the phone. In the second step, a median filter is applied on raw data for data smoothening. In order to ensure that, the developed methodology is applicable to data collected from mobile devices regardless of their orientation inside the vehicle, an orientation correction technique is developed and applied on the cleaned data. Using the orientation angle estimation method developed by [39], the orientation angle of

the phone is determined. A quaternion transformation is then applied on the cleaned data to transform the data from a phone's local to a global coordinate system. Once these preprocessing steps are completed, then different methodologies for trajectory identification are explored.

The trajectory identification problem is approached in two steps. First, the maneuver made by the vehicle at the intersection is identified. Next, the direction of the trip through the intersection is identified. A vehicle can undergo three different maneuver types: left turn, right turn, and through at an intersection. Gyroscope and magnetometer readings from smartphones are employed to develop four different methodologies for maneuver identification. The developed methods are tested on trip data collected from 3 different intersections. Different parameters of gyroscope readings (mean, median, range, etc.) have been explored and range of gyroscope is found to be the most distinctive feature for maneuver identification. Among the different methods developed, a k-medoids clustering based on maximum and minimum of gyroscope global Z component readings was found to be most accurate for maneuver identification. For the Intersection 1, which had 274 trips, the algorithm had a 98% accuracy for identifying left turns, a 96% accuracy for identifying right turns, and a 100% accuracy for identifying through maneuvers. In Intersection 2, which had 414 trips, the algorithm had a 99% accuracy for identifying left turns, 100% accuracy for identifying right turns, and a 99% accuracy for identifying through maneuvers. For Intersection 3 with 68 trips, the algorithm had a 100% accuracy for identifying left turns, right turns, and through maneuvers, respectively. Additionally, other methodologies were tested for maneuver identification. An algorithm is developed based on the threshold value of gyroscope readings and sensitivity analysis of that value is done. A method using DTW distance between a signature pattern of each maneuver was also evaluated to see its effectiveness in maneuver identification. The magnetometer data were

also evaluated to assess their accuracy for maneuver identification. Among all the methods developed and tested, the k-medoids clustering algorithm based on maximum and minimum value of gyroscope readings yielded the highest accuracy for distinguishing different maneuver types.

Next, the direction of the trip through the intersection is identified. Two different methods were developed and tested for direction identification. Of the two methods, a 6 step algorithm developed employing accelerometer data, gyroscope data, and speed information from the OBD is found to be more effective to identify trip direction. For Intersection 1, among the 29 trips that met the consecutive accelerating or decelerating state and gyroscope threshold value requirements, 24 trips were identified correctly. The accuracy was 83%. Of the 235 turn trips, 178 trips met the consecutive accelerating or decelerating state and gyroscope threshold value requirement and of them the direction of 131 trips were correctly identified with accuracy being 74%.

An algorithm is described that can be used to identify a vehicle's trajectory in an urban grid network. First, a method is developed to estimate the distance traveled by the vehicle using OBD speed. Then, an algorithm is demonstrated to match the trajectory with different available routes from the origin to destination based on the estimated distance.

The results demonstrate that the algorithms are effective in identifying a vehicle's trajectory information at an intersection. Overall, this work demonstrates the feasibility of extracting trajectory information through other sensors and, thus, eliminates need for continuous GPS connectivity.

9.2 Benefits and Potential Applications of Vehicle's Trajectory Identification

The proposed methods of vehicle trajectory identification have the potential to be used in several fields. Some of the possible applications are defined in the following sections.

9.2.1 GPS Localization Correction

GPS has some significant issues, such as low precision of localization, vulnerability to spoofing attacks, etc. In urban cities with dense high-rise buildings and underpasses, the GPS signals can get obstructed, causing GPS outage for an extended period of time. In some situations, blockages of some satellites but not others occur resulting in large errors in positional coordinates. Thus, exact vehicle location information may not be available at all times. This study aims to fill this gap by using the smartphone inertial sensor data and speed data collected using OBD to identify the trajectory of the vehicle at an intersection. In addition, GPS is vulnerable to spoofing attacks [2]. This study should help to identify vehicle's trajectory in situations when GPS is spoofed. Using GPS along with the developed methods in this research will increase the localization accuracy of a vehicle.

9.2.2 Minimize GPS Dependency for Trajectory Information

The work done in this dissertation eliminates the need for continuous GPS connectivity for a vehicle's trajectory identification. High-power consumption while GPS is in use [3, 4] is a significant drawback of using GPS in mobile devices. Our work can act as a supplement to GPS and thus freeing from continuous GPS dependency.

9.2.3 Queue Length, Traffic Flow, Travel Time Estimation

This work can be used as an important basis for estimating traffic flow through the intersection. Using this work, vehicle trajectories for each movement can be determined. This information can be incorporated in methods developed by researchers to estimate traffic flow

[45] and queue lengths [46] at intersections. Using the trajectory information, travel time between different origins to destinations can also be estimated. Queue length, traffic flow, and travel time are all valuable information for intersection design, signal time optimization, signal coordination, and traffic management. Thus, this dissertation work can provide valuable insights for infrastructure developments and road network expansion requirements.

9.2.4 Input for Traffic Simulation Software

Using the developed methods for trajectory identification, travel time in route can be estimated. This information can be incorporated into traffic simulation software to evaluate traffic flow dynamics, traffic signal operations, what-if scenarios of different traffic control measures, and to visualize overall network performance.

9.3 Contributions

The research has the following contributions:

9.3.1 Developing a Maneuver Identification Method

An effective method is developed to differentiate three different maneuvers that a vehicle can execute in an intersection: left turn, right turn and through using smartphone's inertial sensor data.

9.3.2 Developing a Vehicle Direction Identification Method

An effective method is developed to identify vehicle's direction at an intersection using smartphone's inertial sensor data and speed data from OBD device, for example, identifying whether a vehicle is going towards North bound direction or South bound direction.

9.3.3 Developing an Algorithm for a Vehicle's Trajectory Identification

An algorithm is developed to estimate the distance traveled by the vehicle by using inertial sensor data and OBD speed data. An algorithm is demonstrated that can be utilized to identify the vehicle's trajectory via route matching with multiple routes from a known origin to a known destination.

9.3.4 Developing a Noise Removal and an Orientation Correction Technique

An appropriate noise removal technique is developed in this dissertation to filter out noises from smartphone collected raw data. Also an orientation correction technique is developed to find a method for trajectory identification that is applicable to all data collected from all smartphones regardless of their orientation.

9.4 Future Work

Future work will include quantifying positional error on GPS coordinates, frequency of GPS signal loss, and estimating the accuracy of trajectory identified by the developed algorithm over the accuracy of a trajectory identified by GPS on a larger dataset. Additionally, it is intended to refine the current algorithm such that it would be applicable to more complex intersection configurations beyond the typical four-legged intersection. Also, research is underway to develop an additional algorithm to identify a phone's movements due to user interaction and apply this to filter out data generated by any local movement. This will enable the current algorithm for maneuver and trip direction identification to be applicable to data collected by phones that were not in a stationary position during the data collection period. In addition, in situations where multiple phones are present inside the vehicle and collecting data simultaneously, it is intended to develop an approach to differentiate data collected by different

phones and identifies the location of phone inside the car based on data collected by a particular phone.

REFERENCES

[1] D. Schrank, B. Eisele, and T. Lomax, "TTI's 2012 Urban Mobility Report,"Texas A&M Transportation Institute," *The Texas A&M University System*, 2012.

[2] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful GPS spoofing attacks," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011, pp. 75-86.

[3] F. Simjee and P. H. Chou, "Accurate battery lifetime estimation using high-frequency power profile emulation," in *Proceedings of the 2005 international symposium on Low power electronics and design*, 2005, pp. 307-310.

[4] W. Ballantyne, G. Turetzky, G. Slimak, and J. Shewfelt, "Tracking & Wireless-powerdown-achieving low energy-per-fix in cell phones-Accelerating deployment of location-based services requires reducing the energy-per-fix (EPF) to a minimum, as," *GPS World*, vol. 17, pp. 24-32, 2006.

[5] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, *et al.*, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Communications Surveys & Tutorials*, vol. 13, pp. 584-616, 2011.

[6] P. Gasper, Z.Sazalay, and S. Aradi, "Highly Auntomated Vehicle Systems." 2014, Budapest: BME MOGI.

[7] Y. Liang, H. Liu, and D. Rajan, "Optimal placement and configuration of roadside units in vehicular networks," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, 2012, pp. 1-6.

[8] S. M. Turner, W. L. Eisele, R. J. Benz, and D. J. Holdener, "Travel time data collection handbook," 1998.

[9] Contus. (2015). *How to Measure Acceleration in Smatphones using Accelerometer*. Available: http://blog.contus.com/how-to-measure-acceleration-in-smartphones-using-accelerometer/

[10] Mozilla Developer Network. (2014). *Orientation and motion data explained.* Available:https://developer.mozilla.org/enUS/docs/Web/Guide/Events/Orientation_and_motion_data_explained.

[11] Hubpages. (2013). *Sensors in smartphone*. Available: http://hubpages.com/technology/Sensors-in-smartphone

[12] J. Fingas. (2012). *Engineer guy shows how a phone accelerometer works, knows what's up and sideways.* Available: http://www.engadget.com/2012/05/22/the-engineer-guy-shows-how-a-smartphone-accelerometer-works/

[13] N. Mohssen, R. Momtaz, H. Aly, and M. Youssef, "It's the human that matters: Accurate user orientation estimation for mobile computing applications," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2014, pp. 70-79.

[14] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin, "Sensing vehicle dynamics for determining driver phone use," in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, 2013, pp. 41-54.

[15] S. Nawaz and C. Mascolo, "Mining users' significant driving routes with low-power sensors," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, 2014, pp. 236-250.

[16] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *NSDI*, 2011.

[17] H. Aly and M. Youssef, "semMatch: Road semantics-based accurate map matching for challenging positioning data," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2015, p. 5.

[18] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, "Cooperative transit tracking using smart-phones," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, 2010, pp. 85-98.

[19] C. Y. Goh, J. Dauwels, N. Mitrovic, M. T. Asif, A. Oran, and P. Jaillet, "Online map-matching based on hidden markov model for real-time traffic sensing applications," in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, 2012, pp. 776-781.

[20] M. Bierlaire, J. Chen, and J. Newman, "A probabilistic map matching method for smartphone GPS data," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 78-98, 2013.

[21] J. Chen and M. Bierlaire, "Probabilistic multimodal map matching with rich smartphone data," *Journal of Intelligent Transportation Systems*, vol. 19, pp. 134-148, 2015.

[22] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data," in *Proceedings of the 31st International Conference on Very Large Data Bases*, 2005, pp. 853-864.

[23] J. Krumm, E. Horvitz, and J. Letchner, "Map matching with travel time constraints," SAE Technical Paper 0148-7191, 2007.

[24] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, "Did you see Bob?: Human localization using mobile phones," in *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, 2010, pp. 149-160.

[25] I. Constandache, R. R. Choudhury, and I. Rhee, "Compacc: Using mobile phone compasses and accelerometers for localization," in *IEEE Infocom*, 2010, pp. 1-9.

[26] M. Uddin, A. Gupta, K. Maly, T. Nadeem, S. Godambe, and A. Zaritsky, "SmartSpaghetti: Accurate and robust tracking of Human's location," in *Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on*, 2014, pp. 129-132.

[27] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive GPS-based positioning for smartphones," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, 2010, pp. 299-314.

[28] Y. Liu, M. Dashti, M. A. A. Rahman, and J. Zhang, "Indoor localization using smartphone inertial sensors," in *Positioning, Navigation and Communication (WPNC), 2014 11th Workshop on*, 2014, pp. 1-6.

[29] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, 2012, pp. 234-239.

[30] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, 2011, pp. 1609-1615.

[31] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, 2013, p. 13.

[32] H. Chu, V. Raman, J. Shen, A. Kansal, V. Bahl, and R. R. Choudhury, "I am a smartphone and I know my user is driving," in *Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on*, 2014, pp. 1-8.

[33] M. Müller, "Dynamic time warping," *Information Retrieval for Music and Motion*, pp. 69-84, 2007.

[34] W. H. Abdulla, D. Chow, and G. Sin, "Cross-words reference template for DTW-based speech recognition systems," in *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*, 2003, pp. 1576-1579.

[35] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 26, pp. 43-49, 1978.

[36] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing,* vol. 5, pp. 657-675, 2009.

[37] Stack Exchange. (2012). *Cross validated*. Available: https://stats.stackexchange.com/questions/94172/how-to-perform-k-medoids-when-having-the-distance-matrix

[38] M. Schirmer and H. Höpfner, "Smartphone hardware sensors," *Mobile Media Group Presentation.* Available online at https://www. uni-weimar. de/medien/wiki/images/Zeitmaschinensmartphonesensors. pdf (accessed 5 June 201 4), 2015.

[39] T. Ozyagcilar, "Implementing a tilt-compensated eCompass using accelerometer and magnetometer sensors," *Freescale Semiconductor, AN,* vol. 4248, 2012.

[40] T. Giorgino, "Computing and visualizing dynamic time warping alignments in R: The dtw package," *Journal of Statistical Software,* vol. 31, pp. 1-24, 2009.

[41] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik, "Cluster: Cluster analysis basics and extensions," *R Package Version,* vol. 1, p. 56, 2012.

[42] *Cluster analysis*. Available: https://www.stat.berkeley.edu/~s133/Cluster2a.html

[43] *Principal component analysis*. Available: http://webspace.ship.edu/pgmarr/Geo441/Lectures/Lec%2017%20-%20Principal%20Component%20Analysis.pdf

[44] B. Caffo. (2007). *Lecture 18*. Available: http://ocw.jhsph.edu/courses/MethodsInBiostatisticsII/PDFs/lecture18.pdf

[45] T. Seo and T. Kusakabe, "Probe vehicle-based traffic flow estimation method without fundamental diagram," *Transportation Research Procedia,* vol. 9, pp. 149-163, 2015.

[46] M. Cetin, "Estimating queue dynamics at signalized intersections from probe vehicle data: Methodology based on kinematic wave model," *Transportation Research Record: Journal of the Transportation Research Board,* pp. 164-172, 2012.

APPENDIX A: PREVIOUS METHODOLOGY

In this section, a method for maneuver using the orientation angle of phone and gyroscope readings is discussed. The objective is to find if the employment of these two parameters together can improve maneuver identification accuracy. This method was later abandoned as it was found that using the gyroscope reading and orientation angles alternately for maneuver identification yields better accuracy than using both simultaneously.

A1 Algorithm for Maneuver Identification using both the Gyroscope Reading and Orientation Angles Simultaneously

The range value of the Z component of transformed gyroscope readings and the difference in value of the orientation angle, ψ for 1st 3 seconds and the last 3 seconds of the trip are employed for the algorithm development. For any particular trip, i, the range of gyroscope reading, r is calculated as

$$\text{Range, r} = max(x_1, x_2, x_3, \dots\dots, x_n) - min(x_1, x_2, x_3, \dots\dots, x_n) \quad (67)$$

where,

x The Z component of gyroscope reading for a trip

The range of orientation angle, s, for that particular trip is calculated as

$$s = mean\{\psi_1, \psi_2, \psi_3, \dots\dots, \psi_{30}\} - mean(\psi_{n-30}, \psi_{n-29}, \psi_{n-28}, \dots\dots, \psi_n) \quad (68)$$

where,

ψ The orientation angle for a *trip* with 1 to n data points

For any particular trip, if r is greater than 0.29, and at the same time, s is greater than 60 degrees, the maneuver executed in the trip is identified as a turn, else it identified as a through maneuver.

For any particular trip with maneuver identified as a turn, if the absolute value of the maximum positive gyroscope reading is higher than the absolute value of the maximum negative gyroscope reading then the turn is identified as a left turn, else it is identified as a right turn. The results for this algorithm are discussed in the next section.

A2 Results

Using both the gyroscope reading and orientation angles simultaneously for maneuver identification yielded poor results for identifying turns. For Intersection 1, the algorithm had a 55% accuracy for identifying left turns, a 63% accuracy for identifying right turns, and a 100% accuracy for identifying through maneuvers (TABLE 32). For Intersection 2, the algorithm had a 51% accuracy for identifying left turns, a 58% accuracy for identifying right turns, and a 99% accuracy for identifying through maneuvers (TABLE 33). For Intersection 3, the algorithm had a 34% accuracy for identifying left turns, a 67% accuracy for identifying right turns, and a100% accuracy for identifying through maneuvers (TABLE 34).

The observation from the results are listed below:

- The algorithm using both orientation angles and gyroscope readings for maneuver identification showed less accuracy for turn maneuver identification
- The algorithm showed high accuracy in through maneuver identification.
- Overall, the accuracy of the algorithm is less than the algorithms that used only gyroscope readings to identify maneuver type.

TABLE 32 Results of maneuver identification using both gyroscope and orientation angles simultaneously for Intersection 1

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	79	0	66	55
Right Turn	0	56	34	63
Through	0	0	39	100

TABLE 33 Results of maneuver identification using both gyroscope and orientation angles simultaneously for Intersection 2

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	63	0	61	51
Right Turn	0	77	56	58
Through	1	0	156	99

TABLE 34 Results of maneuver identification using both gyroscope and orientation angles simultaneously for Intersection 3

	Identified			
	Left Turn	Right Turn	Through	%Accuracy
Left Turn	3	0	6	34
Right Turn	0	2	1	67
Through	0	0	56	100

VITA

Umama Ahmed was born in Sylhet, Bangladesh. She competed her B.Sc. in Civil Engineering from Bangladesh University of Engineering and Technology in 2008. She completed her M.S. in Transportation Engineering from Marquette University, Milwaukee, WI in 2010. She attended one semester in University of Illinois at Chicago as a Ph.D. student before transferring to Department of Modeling, Simulation and Visualization Engineering at Old Dominion University. She has served as the founding member and treasurer of Transportation Engineering Students Organization in 2015. She also served as the secretary of Bangladesh Students' Association at Old Dominion University in 2015. During her M.S. and Ph.D. studies, she published multiple journal articles and presented her papers at top-tier conferences. Her paper based on her M.S. thesis, was awarded "The Best Freeway Operations Paper for the Year 2013" by the Transportation Research Board's Freeway Operations committee. Her research interests include intelligent transportation systems, data analysis, machine learning techniques, and modeling and simulation of transportation systems. She is also interested in working with data collected from mobile devices and using these data to extract information related to human activity, behavior, traffic information, etc.