Acquiring Multimodal Disaggregate Travel Behavior Data Using Smart Phones

by

Roshanak Taghipour Dizaji

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Civil Engineering

Waterloo, Ontario, Canada, 2012

AUTHOR’S DECLARATION

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

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

Abstract

Despite the significant advances that have been made in traffic sensor technologies, there are only a few systems that provide measurements at the trip level and fewer yet that can do so for all travel modes. On the other hand, traditional methods of collecting individual travel behavior (i.e. manual or web-based travel diaries) are resource intensive and prone to a wide range of errors. Moreover, although dedicated GPS loggers provide the ability to collect detailed travel behavior data with less effort, their use still faces several challenges including the need to distribute and retrieve the logger; the potential need to have the survey participants upload data from the logger to a server; and the need for survey participants to carry another device with them on all their trips.

The widespread adoption of smart phones provides an opportunity to acquire travel behavior data from individuals without the need for participants to record trips in a travel diary or to carry dedicated recording devices with them on their travels. The collected travel data can then be used by municipalities and regions for forecasting the travel demand or for analyzing the travel behavior of individuals. In the current research, a smart phone based travel behavior surveying system is designed, developed, and pilot tested. The custom software written for this study is capable of recording the travel characteristics of individuals over the course of any period of time (e.g. days or weeks) and across all travel modes. In this system, a custom application on the smart phone records the GPS data (using the onboard GPS unit) at a prescribed frequency and then automatically transmits the data to a dedicated server. In the server, the data are stored in a dedicated database to be then processed using trip characteristics inference algorithms.

The main challenge with the implemented system is the need to reduce the amount of energy consumed by the device to calculate and transmit the GPS fixes. In order to reduce the power consumption from the travel behavior data acquisition software, several techniques are proposed in the current study.

Finally, in order to evaluate the performance of the developed system, first the accuracy of the position information obtained from the data acquisition software is analyzed, and then the impact of the proposed methods for reducing the battery consumption is examined.

As a conclusion, the results of implemented system shows that collecting individual travel behavior data through the use of GPS enabled smart phones is technically feasible and would address most of the limitations associated with other survey techniques. According to the results, the accuracy of the GPS positions and speed collected through the implemented system is comparable to GPS loggers. Moreover, proposed battery reduction techniques are able to reduce the battery consumption rate from 13.3% per hour to 5.75% per hour (i.e. 57% reduction) when the trip maker is non-stationary and from 5.75% per hour to 1.41% per hour (i.e. 75.5% reduction) when the trip maker is stationary.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Professor Bruce Hellinga for giving me the opportunity to carry out this research, for all his guidance and support during my study, and for his enthusiastic personality which has been such a great encouragement through different stages of this research.

I sincerely appreciate my thesis reviewers, Professor Jeff Casello and Professor Wayne Brodland for all their guidance and suggestions on this research.

I am grateful to all my fellow graduate students in our research group specifically Reza Noroozi, Amir Ghods, Akram Nour, Ehsan Bagheri, and Amir Zarinbal for their support and advice.

I would also like to express my sincere thanks to my dear friends Pedram Izadpanah, Nassim Tolami, Camelia Atefi and Sahar Tolami for all their support, kindness and great friendship.

I wish to deeply thank my parents for their unconditional love, unlimited patience and support through all stages of my life.

I would like to thank my sister, Baharak, and my brother, Amin, for being part of my life and for their love and kindness.

My warmest thanks and love goes to my dear husband, Soroush, for his love, patience, and understanding through my study. I am grateful to him for his tremendous help and guidance throughout this study and for being my main motivation for starting and finishing my Master's program.

Finally, I would like to express my deepest love to my baby, Yaas, who was struggling to be part of this world while I was writing this thesis.

Dedication

This thesis is dedicated to my home country Iran.

Table of Contents

List of Figures

List of Tables

List of Symbols

θ_{cr} = Heading/angle (in degrees) associated with last critical point

θ_i = Heading/angle (in degrees) at point i relative to the North

θ_{th} = Angle threshold (in degrees) which is compared to $\Delta_{\theta, i}$ in order to identify the critical points on non-straight lines

$\Delta_{\theta,i}$ = Change in direction (in degrees) at point i which is calculated as the difference of θ_i and θ_{cr}

Δ_t = Time elapsed from the last captured GPS point

C_{MA} = Maximum age (in seconds) allowed for the last captured location update to be reused instead of capturing a new location update

C_{TO} = Timeout threshold (in seconds) for capturing a GPS point when the point is not available at the scheduled interval

E = Total error associated with a segment of a trip trajectory (m)

e_i = Error at location point i which is calculated as minimum distance between this point and chord $L_{j,k}$ $(j \leq i \leq k)$

G_{th} = Gap threshold which is the time gap (in seconds) to wait after the last attempted GPS point and before restarting GPS satellite connection

i = Location reference index

I_c = Capture interval at which GPS data is captured from GPS satellites

$I_{c,n}$ = Capture interval at state n of state machine

$I_{c,max}$ = Maximum capture interval which is used at the last state of state machine

$I_{c,min}$ = Minimum capture interval which is used at the first state of state machine

$I_{c,pre}$ = Capture interval at previous state of state machine from which a jump to the first state is made

I_t = Transmission Interval at which GPS data is transmitted to server

$L_{j,k}$ = Chord from location point j to location point k

N = Number of location points contained in a segment of a trip trajectory

N_{th}	=	Threshold for the number of GPS points captured at a state which meet a certain criteria
$S_{th,s}$	=	Stationary speed threshold which is used to distinguish between stationary mode and non–stationary mode
$S_{th,w}$	=	Walking speed threshold which is used to distinguish between walking mode and vehicle modes
t_{cr}	=	Timestamp at last critical point
t_i	=	Timestamp at location point i
t_{th}	=	Time threshold to detect critical points on straight line and at stationary mode
$t_{th,w}$	=	t_{th} for walking mode

Chapter 1
Introduction

1.1 Background

The constant growth of urban communities necessitates the development of transportation facilities which are able to manage the growth in travel demands at an appropriate level of service. Network development policies generally provide long term solutions to manage the travel demand, by development of network infrastructure. For example, building new transportation infrastructure (e.g. roads, public transit systems, etc.) or expanding capacity of existing infrastructure is expected to reduce the congestion and the delay experienced by travelers and provide increased mobility and mode choice.

Forecasting future travel demands is one of the major pre-requirements in development of network infrastructure. Activity-based models and four-step travel demand models are two common models applied for prediction of future demands. Development of these models requires an in-depth knowledge of existing travel demands at the individual level (i.e. origin, destination, trip purpose, travel mode of trip makers). These travel data of individuals are finally used to predict the traffic on road segments. In order to understand existing transportation demands and to evaluate and predict the impact of proposed infrastructure improvements and/or policies, it is necessary to collect travel data for a sample of individual travelers.

Beside the long-lasting but costly solutions provided by network development policies, management strategies are also introduced to handle the travel demand without the need to increase the capacity. Transportation Demand Management (TDM) strategies aims to control the demand by increasing the efficiency of existing infrastructure. These management strategies mostly provide short-term but cost-effective solutions. Individualized Marketing (IM) is one of the TDM strategies, which aims to manage the demand by modifying the travel behavior of trip makers through the following steps:

- Identifying travel requirements and preferences of individuals;
- Providing customized travel options based on individuals historical travel patterns;
- Providing self-assessment tool re “optimality” of travel choices;

– Before/After impact analysis.

In order to analyze the current travel pattern of individuals (e.g. Origin, destination, trip purpose, travel mode, activity locations, route choice), travel data of individuals are required.

As a result, collecting travel data from a sample of individual travelers is one of the major requirements for estimating future transportation demands. The following section describes different travel data collection techniques as well as the advantages and disadvantages related to each technique.

1.1.1 Travel data collection techniques

Traditional surveys

Traditional transportation survey techniques for acquiring travel behavior typically have consisted of manual or web-based trip diaries or telephone interviews in which survey respondents self-report their trips and the trip characteristics (e.g. origin, destination, departure time, mode(s) used, travel time, etc.) after trips were taken (Ortuzar and Willumsen 1994).

The fact regarding conventional surveys is that people usually forget the details of their trips (e.g. start/end times, durations, locations, and distances, mode) or they do not have enough time or incentive to complete all the details. Previous studies (Stopher and Collins 2005, Murakami and Wagner 1999, and Murakami et al. 1997) have reported that the data collected through these traditional techniques suffer from poor accuracy. Participants tend to under-report short activity stops, short trips and non-home-based trips. Moreover, auto users tend to underestimate travel time and public transportation users tend to overestimate travel time.

In addition, traditional travel surveys, require significant cost and effort on the part of the survey administers and the survey participants. Doherty et al. 2001 reported that survey participation rates substantially decrease as the effort required to complete the survey increases. The length of study therefore cannot exceed a few days due to the high cost and effort.

As a result of these issues, development of survey techniques which can automatically collect the travel data and extract the trip characteristics are of great interest to transportation professionals.

Traffic sensor technologies

Despite the significant advances that have been made in traffic sensor technologies, there are only a few systems that provide measurements at the trip level and fewer yet that can do so for all travel modes. As an example, loop detectors and Bluetooth detectors do not provide measurements at the trip level. Loop detectors measure spot speed, volume and occupancy, and Bluetooth measure travel time of a sample of vehicles for a section of a facility.

On the other hand, in-vehicle GPS navigation systems do not capture all travel modes such as walking and biking, while the travel data from these modes are essential for building a precise transportation demand model. Non-auto trips tend to also be under represented in other survey methods; therefore, these modes would be of particular interest in travel data collection. Moreover, the data collected through in-vehicle GPS represent the travel behavior of a single vehicle rather than an individual user. Despite the difficulties in acquiring these data, disaggregate travel demand models require the travel behavior data at individual levels.

Stand-alone GPS loggers

GPS loggers are able to collect individual travel data at trip level and also with much less effort compared to traditional surveys. Earlier studies used hand-held dedicated GPS units (e.g. Casas and Arce 1999, Wolf et al. 2001, and Casello et al. 2011); however, the use of dedicated GPS loggers also contains several challenges, namely (i) the distribution and retrieval of the units to/from the survey participants; (ii) the complexity of having survey participant transmit the data from the logger (usually by downloading to a computer connected to the internet) when on-board memory capacity is insufficient for multi-day surveys; and (iii) ensuring the survey participants carry the logger with them for all trips and turn on/off the logger prior to/after each trip; (iv) the time required to achieve a first GPS position (cold-start) may be more than 5 minutes, which results in missing the first part of the trip; (vi) non- automatic data transmission to server which constrains the real time data collection.

Smart phones

The widespread adoption of smart phones equipped with GPS sensors provides an opportunity to overcome most of the limitations associated with traditional surveys and GPS loggers. Using GPS-enabled smart phones, travel behavior data from individuals can be acquired without the need for participants to record trips in a travel diary or to carry dedicated recording devices with them on their travels.

A custom software application can be designed and developed which is capable of recording the travel characteristics (i.e. position, time, speed) of individuals over the course of any period of time (e.g. days or weeks) and across all travel modes. As opposed to traditional travel surveys, which require significant cost and effort on the part of the survey administers and the survey participants, the custom software application can be downloaded directly to the smart-phone by the survey participant. Then the application records the position of the smart phone (using the onboard GPS unit) at a prescribed frequency. These position data are automatically and wirelessly sent to a dedicated server without any survey participant intervention.

Moreover, survey participants are much more likely to carry their smart phone with them for all trips because they typically do so already. Also, most GPS equipped smart phones can obtain position information using assisted GPS, in which the GPS satellite positions is provided wirelessly to the smart-phone over the cellular network, greatly reducing the time required to obtain the first GPS position.

Different types of data (e.g. longitude, latitude, speed, azimuth, satellite information, and accelerometer) can be provided using the GPS-enabled smart phones which provide useful means for data analysis and trip characteristic inference algorithms. Since the GPS data are provided at a prescribed frequency, the accuracy of the recorded travel data is significantly improved in comparison to the data captured from traditional surveys. Furthermore, as opposed to GPS loggers, the frequent and automatic transmission of data to server provides the possibility of travel data collection for both real time and offline applications and without the limitation for on-board memory.

Although the data collection using GPS devices are exposed to different sources of errors such as satellite signal loss, previous studies (e.g. Casas and Arce 1999; Wolf et al. 2001) show that GPS devices still provide highly accurate travel data compared to self-reporting traditional surveys.

To summarize, GPS-enabled smart phones provide reliable and easy-to-access travel data and are potential alternatives to manual surveys. Therefore designing and developing a custom travel survey system to automate the data collection process and to make use of this valuable information is of great benefit to travel data collection techniques.

Figure 1 illustrates the framework of an automatic smart phone based travel surveying system. Smart phones can provide us with the first three steps of this framework, which are the foci of this research and will be discussed in the next sections:

- GPS Data Collection;
- Data Transmission to Server;
- Data Processing and Storing into Database.

Through the last step, Data Analysis and Trip Characteristic Inference, the position data are transformed to trip characteristics, using trip inference algorithms. Different algorithms have been discussed in the literature to transform the position data to travel characteristics, such as trip origin/destination (Doherty et al. 2001, Stopher and Collins 2005), Trip purpose (Wolf et al. 2001) and Travel mode (Chung and Shalaby 2005, Reddy et al. 2010)

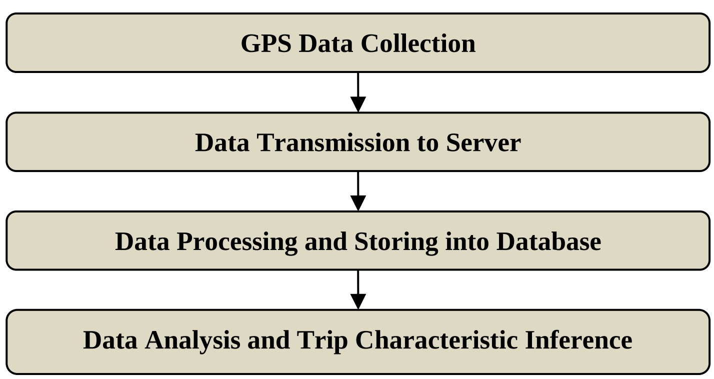
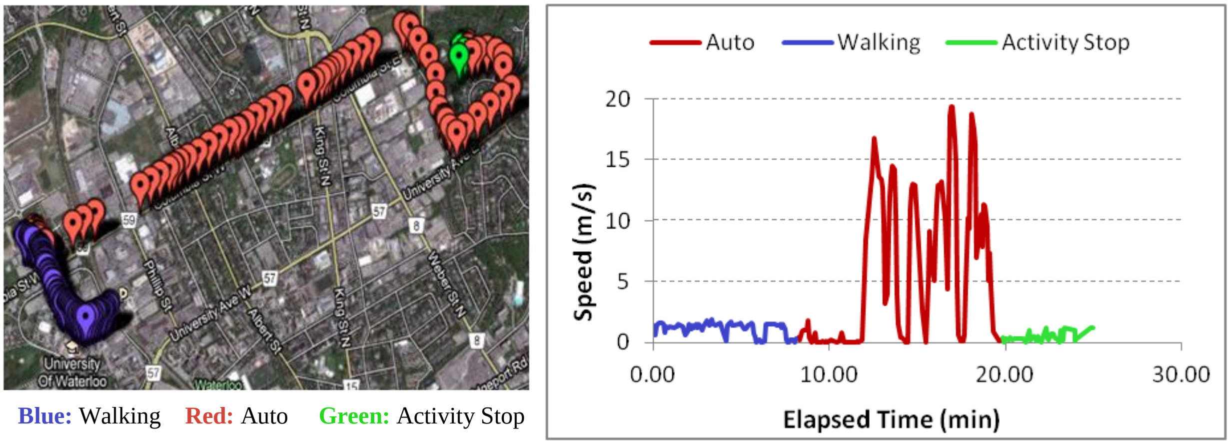


Figure 1: Smart-phone Based Automatic Travel Surveying System Framework

As a simple example, speed reported by the GPS device is one of the basic characteristics used to detect the travel mode. Figure 2a plots the GPS data captured for different travel modes using the travel surveying system developed for this research, and Figure 2b illustrates the speed changes of each mode recorded by the GPS device. In this example identifying the travel mode according to speed changes is not difficult. However, due to the complexity of the travel patterns in many cases, utilization of different information is required for identifying the trip characteristics. For instance, to improve the performance of inference models additional data such as transit stop/route data, land use/parcel data, bike/pedestrian routes, and road network data can be used.



Blue: Walking **Red:** Auto **Green:** Activity Stop

(Source: Google map)

a) **b)**

Figure 2: Travel mode inference using speed

1.2 Problem Definition

The focus of this research is on the design, development and evaluation of the first three stages of the travel surveying system illustrated in Figure 1. The components involved in the process of data collection, data transmission and data storing, in a smart-phone based travel surveying system can be seen in Figure 3.

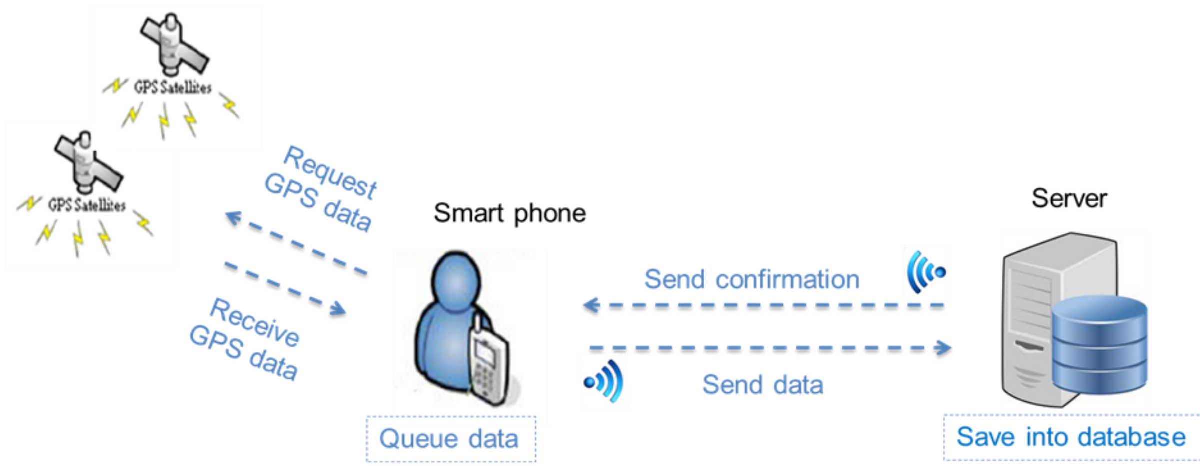


Figure 3: The process of data collection, data transmission and data storing - in a smart-phone based travel surveying system

The design and development of the illustrated system faces several challenges which are described as follows:

Selecting appropriate mode for self-locating using GPS data

Different approaches can be utilized for identifying the position of the smart-phone (e.g. using GPS satellite data or using cell tower triangulation). Most approaches make use of the GPS satellite data; however, even within this approach a number of different modes of operation are possible (e.g. Assisted and Autonomous) which are all discussed in Methodology chapter. Selecting an appropriate mode is important since each mode has different characteristics in terms of the accuracy of the estimated position, energy consumption rate, length of time required to obtain a position estimate, and the cost to the user in terms of data usages and other charges.

Choosing appropriate frequency for capturing GPS data

Determining the frequency at which the GPS data is acquired is a critical issue in data capturing process. Selecting a low frequency for data acquisition results in inaccurate travel data collection. On the other hand, choosing a high frequency provides more accurate travel data but causes significant amount of battery consumption. The goal is to design an application which is able to record travel data accurately while simultaneously minimizing battery power consumption by the application. In the Methodology chapter, a method is proposed which creates a balance between the frequency of data acquisition and power consumption rate.

Selecting proper network transport method for data transmission to server

Data transmission to server can be performed through different communication techniques. Each technique has specific characteristics in terms of the security with which data are transmitted through the network; the accessibility of the network through users' smart phones, service providers, and data plans; the cost imposed to the application developer and the users, and some other features. Therefore, choosing the appropriate method depends on the needs of the application to be developed. Details regarding each transmission method are discussed in the Methodology chapter.

Selecting appropriate frequency for data transmission to server

Determining the proper frequency at which the data are transmitted to the server is also of great importance to automatic data collection using smart phones. While choosing high frequencies is appropriate for real-time applications, considerable amount of battery is consumed by the device for frequent data transmission. In contrast, lower frequencies decrease the battery consumption rate, while they are suitable mostly for offline applications. Moreover, very low frequencies also result in high battery consumption since much data is queued into the device to be transmitted to the server altogether. The impact of data transmission on battery consumption is investigated in Results chapter.

Determining the information to be captured and transmitted to server

In addition to basic information (longitude, latitude, speed) different types of data could be provided by the on-board GPS of the smart phones, such as satellite information, azimuth and the position accuracy reported by the device. Furthermore, some additional data can be directly retrieved from the device, for instance, battery level and accelerometer recordings. This information could be useful in analyzing the performance of the device application and also in increasing the performance of trip characteristics inference algorithms. However, capturing and transmitting of this information to a server might be resource intensive regarding device battery, network bandwidth and storage space. Therefore, it is important to identify the impact of data on the performance of the system and to determine the helpful information, which is discussed through different experiments in Results chapter.

Managing the power consumption rate by the application

The use of smart phones for acquiring travel behavior information significantly influences power consumption of the battery. The objective in acquiring travel behavior information is to obtain relatively frequent location, speed, and heading information (say every 5 seconds). The relatively high frequency of data acquisition is necessary to accurately determine travel behavior characteristics, many of which need to be inferred, such as mode used, location of activity stops, route taken, etc. However, power consumption is correlated with the data acquisition and transmission frequency and therefore there exists a conflict between minimizing power consumption while simultaneously maximizing data acquisition frequency. Different methods and features are investigated in this research to manage the battery consumption rate while considering the accuracy level of trip data, which are discussed in the following sections.

1.3 Objectives

The main goal of this research is to design, develop and evaluate an automatic smart phone-based travel surveying system for recording individuals travel behavior data.

In order to achieve this goal, the following objectives are laid out for this thesis:

- Design and development of an application for Blackberry GPS-enabled smart phones which automatically and frequently collects the travel data and transfers the data to a dedicated server.
- Design and development of an application for the server side which continuously listens for a connection from device side and receives the transferred data by the device.
- Design and development of a database into which the data received by the server is stored
- Evaluation of the accuracy of position information obtained from device.
- Investigation of the impact of the application on device battery performance and development and evaluation of methods to optimize the battery consumption.

1.4 Thesis Structure

The remainder of this thesis is organized as follows:

- Chapter 2: Literature Review - describes the relevant literature regarding GPS position update methods.
- Chapter 3: Methodology - provides the frame work of the entire system and describes the algorithms and methods associated with each system component.
- Chapter 4: Results and Discussion - provides a preliminary investigation on the reliability of the developed system in recording the travel data, examines the performance of the proposed methods, and evaluates the impact of the developed application on device battery performance.
- Chapter 5: Conclusions and Recommendations

Chapter 2
Literature Review

The main challenge with smart phone based travel behavior surveying systems is the significant amount of energy consumed by the device to frequently calculate and transmit the GPS positions. An evaluation of the impact of GPS functionality on cell phone power consumption can be seen in the research work conducted by Aguilar 2008. In order to manage the energy consumed by the GPS fixes, different approaches have been studied, either from the perspective of cellular phone design (Ballantyne et al. 2006) or GPS position update methods (Wolfson et al. 1999, Leonhardi et al. 2000, Leonhardi and Rothermel 2002, Treu et al. 2008, and Barbeau et al. 2008).

This chapter provides a summary of the previous work on position update methods. These methods are in fact the techniques utilized by researchers for calculating the GPS updates and transmitting them to servers. The chapter starts by providing a summary of the basic update methods. Then, it follows by describing the optimized update techniques. Finally, a discussion is presented on these techniques.

2.1 Basic Position Update Methods

Leonhardi and Rothermel 2002 classified the position update methods into three main categories of Querying, Reporting and Combined, which are briefly described here:

2.1.1 Querying

In this method, the location updates are calculated and transmitted to a server based on the server's request and depending on the accuracy required by server. This method avoids unnecessary data transmission to server because the position data are transmitted only on-demand. The request by server could be made for occasional location information or based on a predefined frequency. However, this method is mostly efficient when frequent location updates are not required; otherwise, for each location update two connections need to be made between the device and the server: one from server side to request a new position update and the other one from device side to transmit the new location to server.

2.1.2 Reporting

In this method, the decision on when to transmit the position update to the server is made by the device side. Reporting method provides a more efficient way for transmitting frequent location updates to server compared to the Querying method. To be more precise, for each transmission from device side to server side there is no need for an extra query to be made by server side. However, the frequency at which location updates are transmitted might not necessarily provide the accuracy level required by the server. This method could be further categorized as (1) Simple, (2) Time-Based, (3) Distance-Based and (4) Dead-Reckoning subcategories. All the sub categories except for Simple use a fixed rate for calculation of position updates which might not necessarily be the same as position transmission rate.

In the Simple case, the location update is calculated and transmitted to server whenever a new one is available by mobile sensor (e.g. GPS). Therefore the number of data calculations and transmissions to server is determined based on the sensitivity of the mobile sensor, which might provide a fairly large number of updates. In general, utilization of the Simple case might result in transmission of too many unnecessary data points to the server.

In the Time-Based case, the position data are transmitted to the server at a prescribed time interval. The issue with the Time-Based method is that information of little or no value is sent to the server when the subject is stationary or is moving with a low speed.

In the Distance-Based method, a distance threshold is utilized for transmitting the location information to server. Whenever the distance between the current position and the last transmitted position exceeds the distance threshold, the location information is transmitted to server. As a result, less data are transmitted when the subject is stationary or is moving with low speed, and more data is transmitted when the subject is travelling with high speed.

Dead-Reckoning is similar to the Distance-Based method, but with some modifications. In the Dead-Reckoning method, the server utilizes the last transmitted location information such as position, speed and direction to predict the current location. The same calculations are also made in device side. Only if the distance between the estimated location and actual location exceeds a distance threshold is the actual position transmitted to server. This method is specifically beneficial when the moving path of the subject is known beforehand which results in better estimation of the current location. Dead-Reckoning approaches have been utilized in different research such as Wolfson et al. 1999, Leonhardi et al. 2000, and Treu et al. 2008, through which new features have been proposed to optimize the functionality of this method.

2.1.3 Combined

To overcome the issues of the discussed methods, Leonhardi and Rothermel 2002 integrated the features of the Querying and Reporting methods by proposing a new method named Combined method. In this method, a Distance-Based Reporting protocol is used to transmit the position update to the server when the distance exceeds the distance threshold. On the other hand, the server side checks the accuracy of the reported location information and if the required accuracy is not met a request is made by the server for a new location update. In this method, the value of the distance threshold is dynamically adjusted based on the movement characteristics of the subject which is tracked by the device and the server.

2.2 Optimized Position Update Methods

Barbeau et al. 2008 conducted an assessment on different position update methods. They stated that Querying method (also called Polling method) is only beneficial when just a few location updates are required; for example, when the present location of the user is being plotted on the map. On the other hand, the Time-Based Reporting method, in which the data are periodically transmitted to server, is not efficient when high frequencies are used. The reason is that much of the transmitted data might be needless and resource intensive. As for low frequencies, it might not provide the required accuracy for tracking the user path. Moreover, the problem with the Distance-Based method is that it tends to transmit unneeded data when there is no change in the direction of user path, e.g. on a straight line. They also argued that although the Dead-Reckoning method reduces the number of messages sent to the server, its functionality is based on a series of calculations made on both device side and server side to estimate the current location, which might considerably increase the resource consumption. To resolve the mentioned issues, Barbeau et al. have proposed two algorithms, namely (1) Critical Points Algorithm; and (2) Location Aware State Machine Algorithm. The Critical Points algorithm integrates different characteristics of the previously described update methods for transmitting the data into server. Location Aware State Machine Algorithm provides a dynamic solution for managing the frequency at which the location updates are calculated. Location Aware State Machine Algorithm attempts to resolve the problem of the Time-Based Reporting algorithm in which low frequencies result in inaccurate data collection and high frequencies result in significant resource consumption. The following section describes the two optimization algorithms proposed by Barbeau et al.

2.2.1 Critical Point Algorithm

Critical Point (CP) algorithm aims to decrease the data transmissions to server by ignoring non-critical points and only transmitting the critical points to server. As a result, the resources consumed by the device and the network (i.e. battery and bandwidth) are reduced. Non-critical points are the data points which do not provide extra information regarding the user path. For example, the repetitive data points at stationary mode or the data points within a straight line are non-critical points. On the other hand, the points at the corners of a path or at the locations where direction of the subject is changed are considered as critical points. Changes in direction are computed using azimuth data, which is illustrated in Figure 4. At each stage the three most recently captured points are used to calculate the change in direction. In other words, the azimuth (angle) between the first and second point and the first and third point are calculated. If the calculated value exceeds a predefined threshold the second point is marked as a critical point and transmitted to the server. Change in direction is only applicable when the subject is moving. In order to determine if the subject is moving or not, a speed threshold might be used.

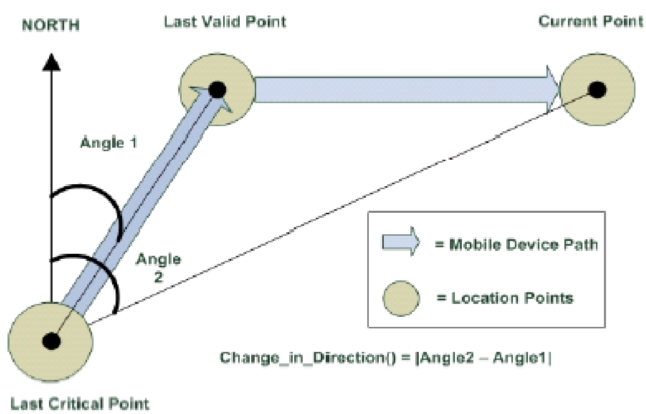
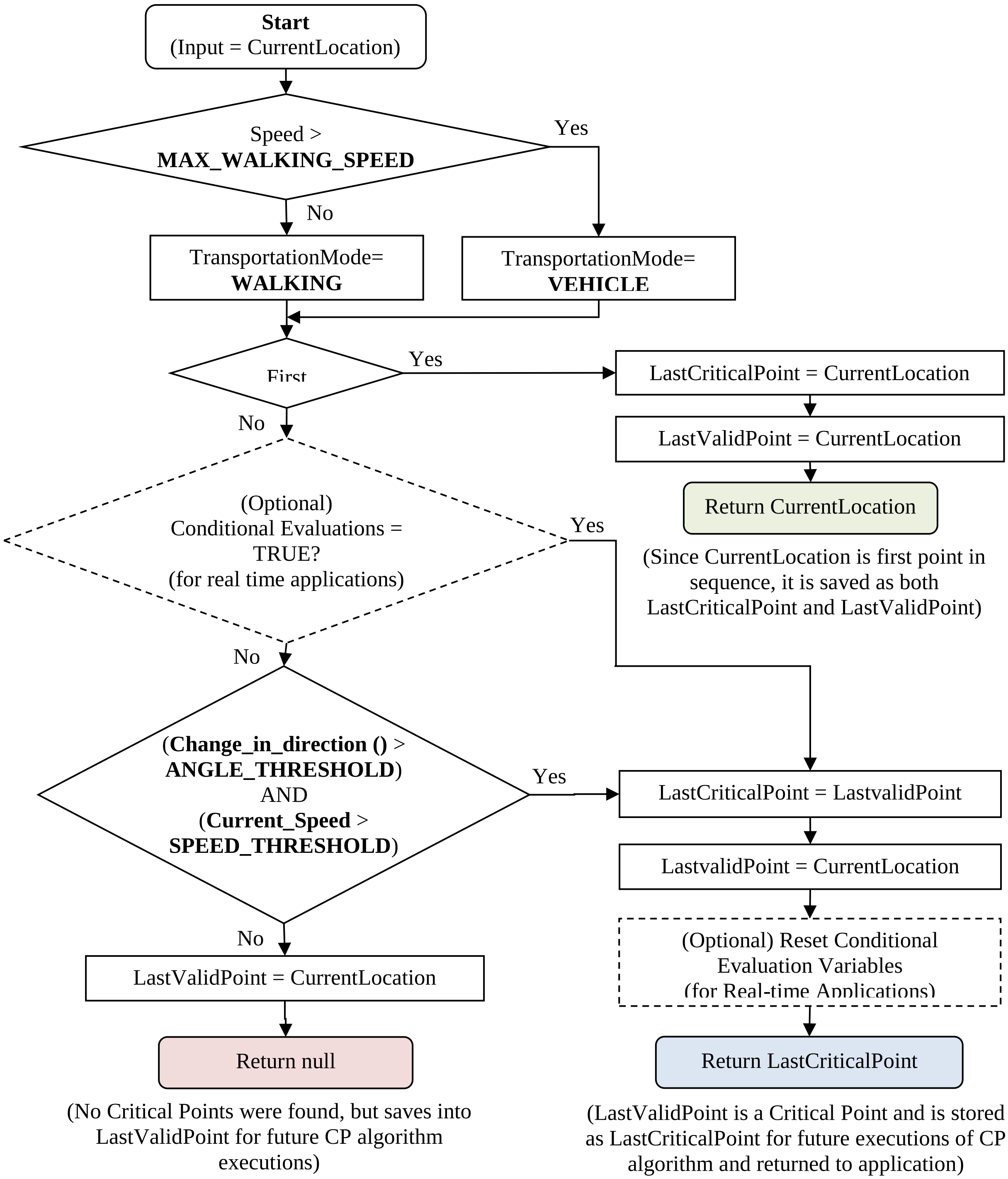


Figure 4: Calculating changes in direction (Barbeau et al. 2008)

It is also beneficial to utilize different azimuth values based on the travel mode. For example, for walking mode larger values of azimuth and for vehicular modes smaller values of azimuth are appropriate. To detect the travel mode, other speed thresholds might be considered. For instance, the speeds less than 5 m/s might be considered as walking and the speeds equal or more than 5 m/s might be considered as vehicle trip.

Figure 5 is the flow chart illustrating how the CP algorithm is executed. The first point calculated by the Location-Based Services (LBS) application is the first input to this algorithm and by default is considered a critical point. Whenever a new location point is provided by the application it is entered as the new input to CP algorithm (e.g. every 4 seconds). Since the first point is marked as a critical point it is transmitted to server. The algorithm also keeps a temporary copy of this point for further use. When receiving the second point, the critical point algorithm still is not able to determine if the point is critical. Therefore, the output of the CP algorithm at this stage is null and nothing is transmitted to server. However, information regarding the second point is saved into the algorithm for further analysis. When receiving the third point the data related to this point is first saved into the algorithm for assessment. Then, the change in direction is calculated using the difference in azimuth between the first and second point and the first and third point. If the change in direction is larger than the predefined angle threshold, then the speed value for the third point is assessed. If the speed is also larger than the speed threshold, it indicates that the subject is moving. Then the output of the CP algorithm would be the second point showing that this point is a critical point. As a result, the second point is transmitted to server. If the change in direction or the speed is less than the related threshold value, no critical point is produced as the output of CP algorithm and no transmission is made to server. When each new point is received the information for that point is temporarily saved into the CP algorithm for further calculations. Then the change in direction is calculated using the last three points. If the change in direction and also the speed of the last point surpass the threshold values, then the second last point is marked as critical point by CP algorithm and is transferred to server. Otherwise no point is generated as the output of the CP algorithm and no data is transmitted to server. This process is repeated until reaching the last location update provided by the application.



Change_in_direction () = Angle calculation (varies depending on the current speed)
ANGLE_THRESHOLD = Minimum change in azimuth when a point is marked critical
SPEED_THRESHOLD = Minimum speed when device is considered moving

Figure 5: The critical point algorithm (Barbeau et al. 2008)

In order to determine the critical points, additional criteria (conditional evaluations) such as time threshold, distance threshold and location probe might also be considered. Using time threshold, a point is marked as critical point if the time elapsed from the last critical point exceeds the threshold. This guarantees that the server is receiving a point after a certain time if no change in direction is made during that time; for example, when the user is moving on a straight line or not moving at all. Distance threshold is similar to the time threshold. The only difference is that a point is marked as critical point if the distance from the last critical point exceeds the threshold. This is applicable for the case when the user is traveling on a straight line. Time threshold and distance threshold can respectively be considered as the Time-Based and the Distance-Based protocols described as part of Reporting position update method in previous section. As for Location Probe, a point is marked as critical point when it is queried by the device. This can be considered as the Querying method discussed in previous section.

Barbeau at al. employed a Sanyo SCP-7050 mobile phone to evaluate the performance of the proposed CP algorithm. Since in their proposed method one transmission is made to server by detecting each single critical point, they investigated the impact of transmission interval on device battery life. Figure 6 illustrates the results of this experiment for transmission interval values of 15, 30, and 60 seconds. The results show that increasing the interval between transmissions to server increases the battery life considerably. The impact of CP algorithm on user path can also be seen in Figure 7. This figure shows that the critical points are detected when a change in direction happens, and non-critical points are discarded along the straight lines. Table 1 also presents the results from running the CP algorithm for 8 sample trips. This table shows the reduction in number of transmitted points as the result of running CP algorithm, which is on average more than 80%. The table also demonstrates the impact of CP algorithm on bytes savings and financial savings for the tested trips.

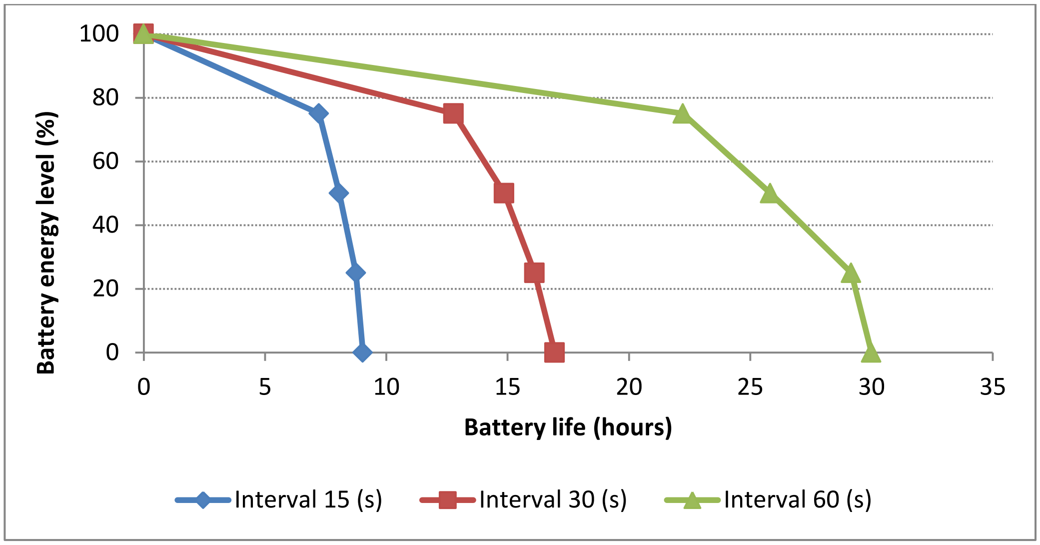


Figure 6: Impact of Wireless Transmissions Interval on Battery Life (Barbeau et al. 2008)

(a) (b)

Figure 7: Trip example showing the reduction in points – (a) all the points, (b) critical points (Barbeau et al. 2008)

Table 1: Savings of Data Charges Using the Critical Point Algorithm (Barbeau et al. 2008)

Trip	Total Number of Points	Number of Critical Points	% Fixes Transmitted	Bytes Saved	Financial Savings
1	73	26	35.61	5593	$0.17
2	363	56	15.42	36533	$1.10
3	489	65	13.29	50456	$1.50
4	208	73	35.09	16065	$0.48
5	357	62	17.37	35105	$1.05
6	2330	159	6.8	257159	$7.71
7	1022	139	13.6	105077	$3.15
8	811	137	16.89	80206	$2.40

2.2.2 State Machine Algorithm

As mentioned previously, using small time intervals for requesting GPS positions results in significant amount of battery consumption. To resolve this problem, State Machine algorithm tends to reduce the number of position updates by managing the frequency at which the positions are acquired. In other words, instead of using a fixed time interval, State Machine algorithm utilizes dynamic time intervals to acquire the GPS points. Since collection of accurate trip data needs short time intervals such as 4 or 5 seconds, large time intervals could be used when the user is stationary or is indoors. Figure 8 shows how the State Machine algorithm adjusts the position calculation rate based on the status of the user. State Machine algorithm starts at state 0 where the GPS points are acquired with small time intervals such as 4 seconds. If for a certain period of time the user is stationary (speed is less than a predefined threshold) or no valid GPS data is received due to weak GPS signals, State Machine algorithm moves toward the next states by making a step by step decrease in the frequency of data acquisition. The frequency is decreased until state machine reaches to the last state (state n), at which time interval becomes as large as a predefined threshold value such as 8 minutes. On the other hand, if a valid location point is captured, the time interval is gradually decreased until State Machine reaches to the first state (state 0). If the speed value for the first valid point exceeds the predefined stationary speed threshold (showing that the user is moving), then the State Machine directly snaps back to state 0.

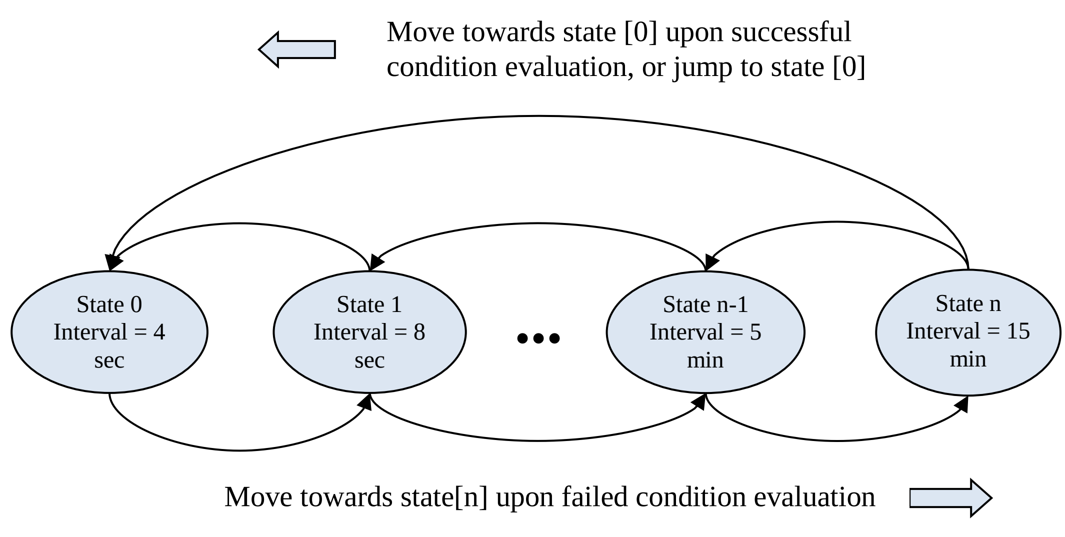


Figure 8: The location-aware state machine (Barbeau et al. 2008)

Barbeau at al. employed a Sanyo SCP-7050 mobile phone to analyze the impact of position request time interval on battery life. The results of this experiment are illustrated in Figure 9 which show that the battery life is considerably increased when time intervals of 300 seconds and more are used for calculation of GPS points. As an example, increasing the time interval from 60 seconds to 300 seconds results in the increase of battery life from a value less than 14 hours to more than 33 hours.

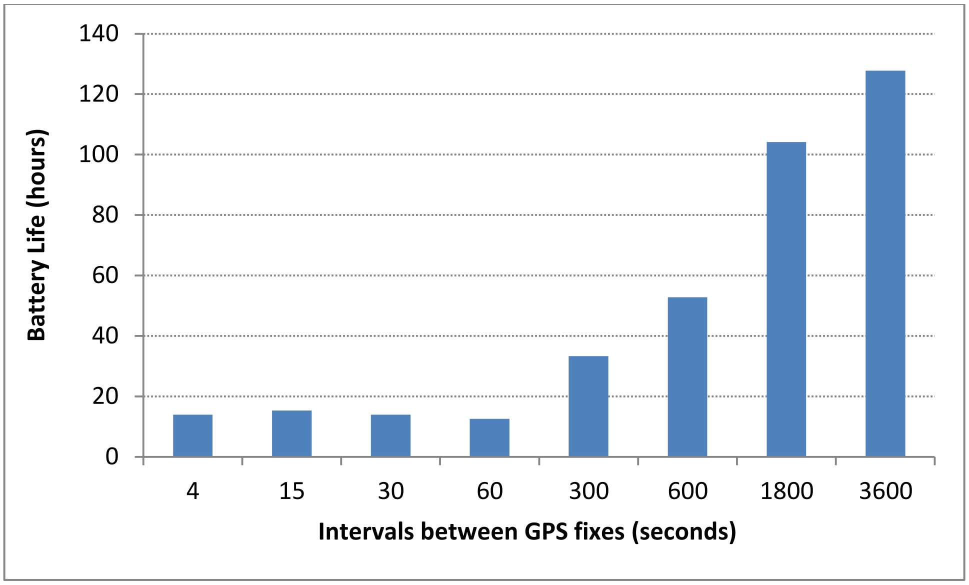


Figure 9: Impact of GPS Position Request Interval on Battery life (Barbeau et al. 2008)

An LBS application can benefit from CP algorithm and State Machine at the same time. Considering that normal operations of the mobile phones such as phone calls, screen display and messaging consume part of the device energy, using the proposed algorithms concurrently can maximize the energy savings.

2.3 Discussion

Although the proposed algorithms by Barbeau et al. 2008 show encouraging performance and the results demonstrated are informative, the direct impact of applying the State Machine algorithm and the CP algorithm on battery consumption rate has not been evaluated. As an example, no results have been presented to show how dynamic changes of position calculation frequency made by the State Machine algorithm affect the battery life. Furthermore, both the CP algorithm and the State Machine algorithm contain parameters that impact the performance of the algorithm. In the research by Barbeau et al. no analysis has been reported that describes the determination of the optimum parameter values in terms of minimizing the battery consumption and maximizing the position (i.e. trajectory) accuracy.

The CP algorithm and the State Machine algorithm proposed by Barbeau at al. have been used in different research works such as the work by Gonzalez et al. 2008 and Taylor and Labrador 2011. Gonzalez et al. used neural networks to analyze the impact of applying the CP algorithm on the accuracy of travel mode detection. Taylor and Labrador attempted to improve the performance of the State Machine algorithm by filtering the noisy GPS points using Modified Kalman Filters. Nevertheless, there does not appear to be any research yet conducted examining the trade-off between parameter values and battery performance for the CP and the State Machine algorithms.

Moreover, the implementation of the proposed algorithms by Barbeau et al. encounters some issues which are not addressed in the conducted research. The following is a brief description of these issues.

As mentioned previously, the CP algorithm considers the last 3 captured points at each stage to calculate the difference in direction. The problem with this approach is that when the path on which the subject is moving is a segment of a circle with large diameter, the change in direction never exceeds the predefined threshold and no critical point is detected due to angle change. Figure 10 is a hypothetical circle trajectory which includes eight GPS points captured at equal time intervals and $\Delta\theta$ is the calculated change in direction for the first three points. Assuming that $\Delta\theta$ is less than the predefined threshold, the change in direction never exceeds the threshold for each three consecutive points and as a result no critical point is recorded.

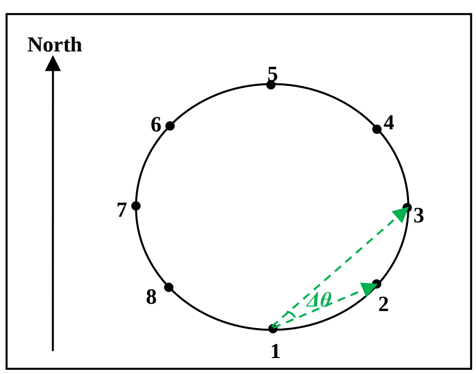


Figure 10: The issue related to calculation of change in direction in the CP algorithm proposed by Barbeau et al. 2008

Regarding the State Machine algorithm, two issues can be pointed out: first, at each state of the state machine if a GPS point with speed greater than the predefined stationary speed threshold is captured, it might be an indication that the user is moving. Therefore, in this case, it is expected that the state machine snaps back to the first state to capture the next points with maximum frequency. However, in the algorithm proposed by Barbeau et al., a jump to the first state is possible only from the last state and not from middle states. If the state machine is in the middle states and a valid point with speed higher than the predefined threshold is obtained, the movement to the first state will be performed gradually not immediately which prevents accurate data collection. Second, when the state number is greater than 0 (e.g. last state) acquisition of a point with speed higher than the predefined threshold might be related to GPS error. In other words, sometimes the reported speed is higher than predefined threshold while user is stationary. This might happen when the GPS signals are weak. Therefore, in this case the state machine should identify if the high speed is related to errors or to the movement of the user. If the high speed is related to an error, then jumping back to the first state and starting the whole process from the beginning would be inefficient. Nonetheless, in the algorithm proposed by Barbeau et al. no analysis is conducted to identify errors when positions with high speeds are reported.

Therefore, as a part of current research the two algorithms are enhanced to address the mentioned issues, and the optimum values of the parameters for the algorithms are determined in terms of minimizing the battery consumption and maximizing the position (i.e. trajectory) accuracy.

Chapter 3
Methodology

In the current research a travel behavior surveying system has been designed and developed for GPS-enabled Blackberry smart phones. The implemented system includes the following parts:

- An application for device side which runs on GPS-enabled Blackberry smart-phones and automatically and frequently collects travel data and transfers the data to a dedicated server.
- An application for server side which continuously listens for a connection from device side and receives the transferred data.
- A SQL Server database into which the data received by the server is stored.

This chapter starts by illustrating the framework of the entire system and then describes the algorithms and methods that are associated with each system component.

3.1 System Framework

Figure 11 illustrates the implemented system framework which includes two major components of device side and server side. This section briefly introduces the modules within these two components.

Device side

The operations within the device side application can be explained through the following modules:

Startup Module

Using this module, the log-in operation into the system is performed and the settings within the application are adjusted and Data Acquisition Module is started.

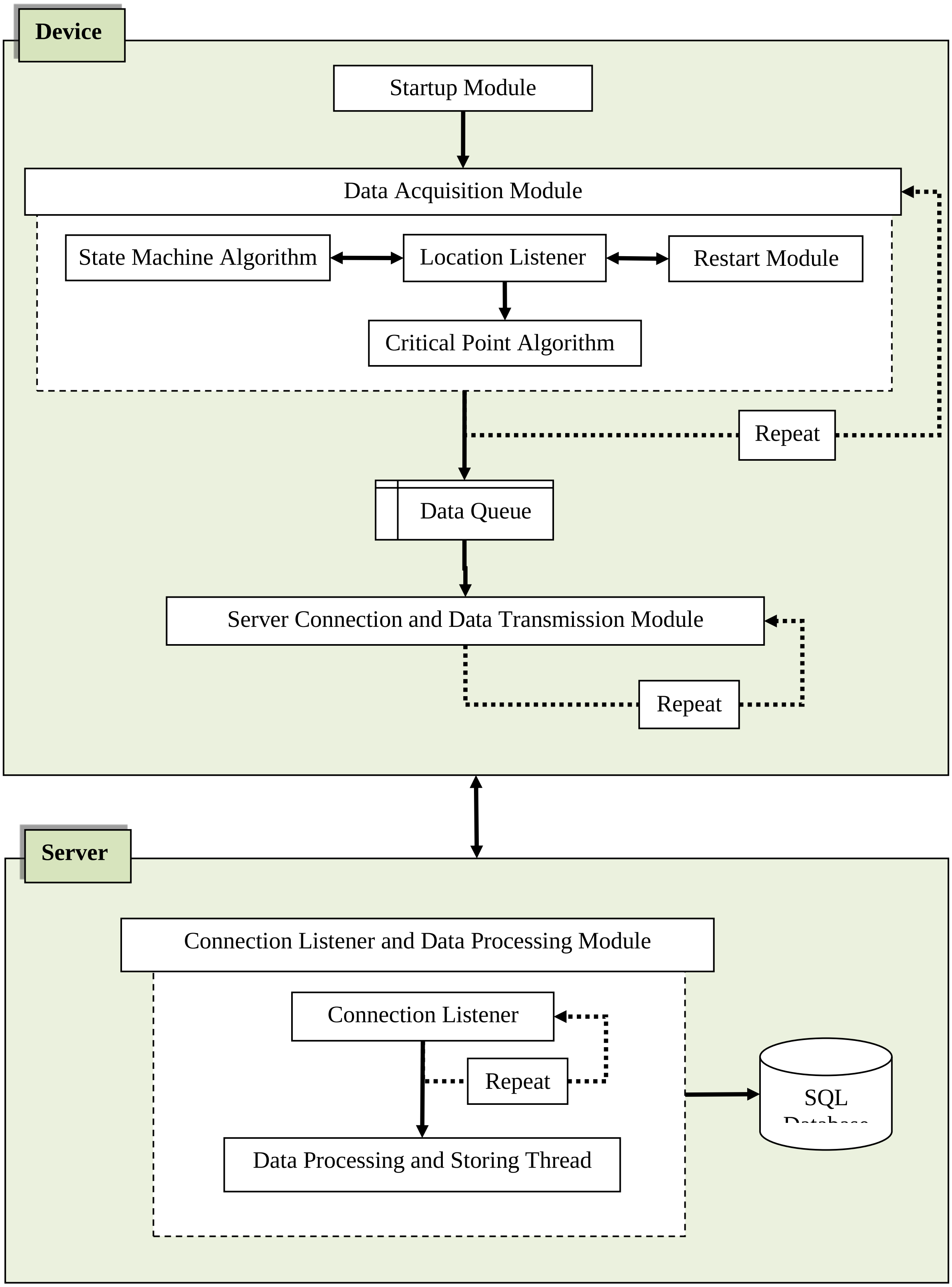


Figure 11: System Framework

Data Acquisition Module

The procedure within this module is described by the means of the following sub modules:

- **Location Listener**: This module communicates with GPS satellites and listens for a new location data at a prescribed frequency. Some complementary information is also retrieved through this module.
- **Critical Point (CP) Algorithm**: This algorithm identifies the critical data points of the user travel data.
- **State Machine (SM) Algorithm**: This algorithm is in connection with Location Listener and determines the frequency at which the data should be captured.
- **Restart Module**: This module restarts the Location listener when no GPS point is available for a certain time.

Data Queue

The data produced by Data Acquisition Module is queued into a Data Queue to be transmitted to server.

Server Connection and Data Transmission Module

This module reads the data from Data Queue at a prescribed frequency. Then it communicates with server side and transports the data over the network.

Server side

The server side contains the following components:

Connection Listener and Data Processing Module

This module is responsible for the operation of the application on server side; the module can be divided into the following sub modules:

- **Connection Listener**: This module listens for requested connections from device side and receives the transmitted information from device.
- **Data Processing and Storing Thread**: This module processes the received data and then stores the information into a database.

SQL Database

The application on server side is in connection with a SQL database, into which the processed data is stored. This database includes different tables which categorize the travel data received from users.

The details associated with each system component are discussed in the following sections. First, the methods for establishing communication between device and GPS satellites, and also between device and server are elaborated. Then the algorithms and methods associated with device and server components are explained.

3.2 Communication Methods

This section first describes the GPS modes utilized by Blackberry smart phones to communicate with GPS satellites and to obtain the GPS positions. Then the network transport methods for transmitting the data between a Blackberry smart phone and a server are discussed.

3.2.1 Blackberry GPS Modes

According to *BlackBerry Java Development Environment* 2009 and *BlackBerry Java Application* 2010, several modes might be employed by Blackberry smart phones to calculate a GPS position. These modes are categorized into the main GPS modes and some complementary modes. Each of these modes is suitable for a specific kind of application since they suggest different characteristics in terms of accuracy of position, startup performance, network cost, etc. This section gives a brief description on characteristics of each mode.

Main GPS Modes

Assisted

Assisted GPS (AGPS) mode obtains the GPS positions using the satellites signals and wireless network. Based on *BlackBerry Java Development Environment* 2009, the GPS positions acquired by this mode are accurate. When the satellites radio signals are weak, the wireless network acts as a supplement to obtain the GPS information.

In fact for this mode, AGPS servers are utilized to provide the orbital information of the satellites. AGPS servers contain databases which keep the orbital information downloaded from GPS satellites. This information is accessible to AGPS-enabled Blackberry devices through mobile network radio bearers (e.g. GSM, CDMA or Wi-Fi). The devices can acquire the orbital information from AGPS servers in a shorter time than acquiring them directly from satellites, because mobile network radio bearers usually operate on a high rate of data (“Assisted GPS," *Wikipedia*). Consequently, the time-to-first-fix (TTFF) is reduced and the startup performance is increased. Usually, the data for the first fix are provided within 30 seconds. As a result of quick response time for AGPS mode, the battery consumption of the device is also reduced. In addition to providing a short response time, the information provided by AGPS server enhances the connection between the GPS receiver and the GPS satellites when the signals are very low (e.g. inside the buildings).

Communication to AGPS servers might impose some charges on mobile users since the communication is made through a data connection such as internet. However, it is usually considered as a part of data plan provided by cell phone carriers.

Since the Assisted GPS mode operates on a wireless network, the coverage of the wireless network is required for this mode. Also this mode should be supported by the cell phone carrier and the device itself.

Autonomous

Autonomous mode obtains the GPS positions only using the GPS satellites and without employing the wireless network. Since the radio signals from satellites are the only resource utilized for Autonomous or stand-alone GPS to acquire positions, this mode cannot obtain the GPS data when there is not a clear view of satellites, e.g. indoors or in areas with high obstacles. On the other hand, relying only on radio signals from satellites gives the benefit of acquiring the GPS positions without imposing a cost on users.

Based on *BlackBerry Java Development Environment* 2009, the GPS positions retrieved by this mode are accurate; however, the response time is longer than the other GPS modes. Acquiring the first fix might take lots of minutes since the GPS receiver needs to get synchronized with at least four satellites. In other words, to obtain the first fix, stand-alone GPS should receive clear signal continuously to download the orbital information like ephemeris[1] and almanac[2] which can take up to 12.5 minutes. If the communication fails while the orbital information is being collected, the whole process must be started from the beginning (*NavCen* GPS User 1996). However when there is a clear view of satellites, the time-to-first-fix (TTFF) usually does not exceed 2 minutes. In general, the long response time by the Autonomous mode results in high battery consumption by the device.

Cell Site

Cell site mode employs Geo location service or wireless network to obtain the GPS positions. In fact for this mode the GPS data are acquired from the cell site towers. Acquisition of GPS information using Cell Site mode takes less time than the assisted and autonomous modes which results in less battery consumption by the device. However, the information provided by this mode suffers from poor accuracy. Moreover, speed and other tracking information cannot be reported by this mode.

Cell site mode employs the wireless network to obtain the first fix. Therefore, wireless network coverage is required for this mode for acquiring the data. Furthermore, cell site mode should be supported by the cell phone carrier and also by the mobile device.

Table 2 provides a summary of the features of the main GPS modes provided by Blackberry.

Table 2: Comparison of main GPS modes (*BlackBerry Java Development Environment* 2009 and *BlackBerry Java Application* 2010)

GPS mode	Service	Accuracy	Battery Usage	Indoor/ Obstruction coverage	Response Time	Cost
Assisted	Satellite and Wireless Network	High	Medium	Yes	Fast	Yes
Autonomous	Satellite-only	High	High	No	Slow	No
Cell Site	Cell Towers and Wireless Network	Low	Low	Yes	Very Fast*	No

*Cell Site is faster than Assisted

[1] a table showing the positions of astronomical objects in the sky on a number of dates in a regular sequence.
[2] an annual publication containing astronomical information such as future positions of celestial objects and star magnitudes.

Given that the accuracy of the positions provided by Cell site mode is very low (approximate positions), this mode is not suitable for the for applications that require highly accurate position data.

Autonomous and Assisted modes could both be appropriate options for accurate data collection. However, if the time to first fix (TTFF) and battery consumption are important factors within the data collection process, and the device is intended to be used in urban areas where the signals are poor, Assisted GPS might be the preferred option.

Complementary GPS modes

Blackberry APIs offer different complementary modes which apply Qualcomm gpsOne[1] technology and work on CDMA[2] network. These technologies can enhance the performance of the data acquisition application by optimizing the speed, accuracy and device energy consumption. These APIs are only applicable for Blackberry devices with version 5 or higher. The following provides a summary of different complementary modes offered by Blackberry.

MS-based

This mode employs wireless network to obtain the satellite information for the first fix, which enhances the TTFF. Then, for the next GPS fixes, the mode is changed to autonomous.

MS-assisted

This mode utilizes wireless network to acquire the satellite information

Speed Optimal

This mode attempts to retrieve the GPS fix with the highest speed, while considering the settings placed by the application

[1] Qualcomm gpsOne integrates different technologies such as GPS, cellular methods, Wi-Fi, and sensors to acquire the GPS data. Consequently, the performance is improved in terms of accuracy, speed and power consumption (Moeglein, M., and N. Krasner. 1998).
[2] CDMA (Code division multiple access) utilizes a single channel for transmitting the information of multiple users at the same time, which are divided by an assigned code. As a result, the network bandwidth is shared by multiple users (Theodore S. Rapporteur 2002)

Accuracy Optimal

The focus of this mode is on providing the most accurate GPS position that is available at the time of request. As a result, the accurate position might be computed using local calculations or be retrieved through the wireless network.

Data Optimal

This mode attempts to provide the GPS data using slightest possible amount of traffic. Therefore the data could be achieved through local calculations or by utilizing the network information.

Bluetooth® enabled GPS

A Bluetooth® enabled GPS device can be paired and used with Blackberry cell phone to obtain the GPS position. In this case, the GPS data will be retrieved based on the configuration of the Bluetooth® enabled GPS device which cannot be adjusted through the BlackBerry device application.

Since the mentioned modes can only be utilized on devices with Qualcomm gpsOne technology and on CDMA network, their application is not feasible for many Blackberry devices. However, when all conditions for applying these modes are available, the application can benefit from the features of a specific mode depending on its requirements

3.2.2 BlackBerry Network Transports

According to *Network communication* 2010, *A14 Network Transports* 2009, different communication techniques are offered by Blackberry solution for transmitting the data from Java applications through the wireless network. The following is the list of transport methods provided by BlackBerry® Java® Development Environment which are also illustrated in Figure 12.

- MDS
- BIS-B
- WAP 1.x
- Direct TCP

– Wi-Fi

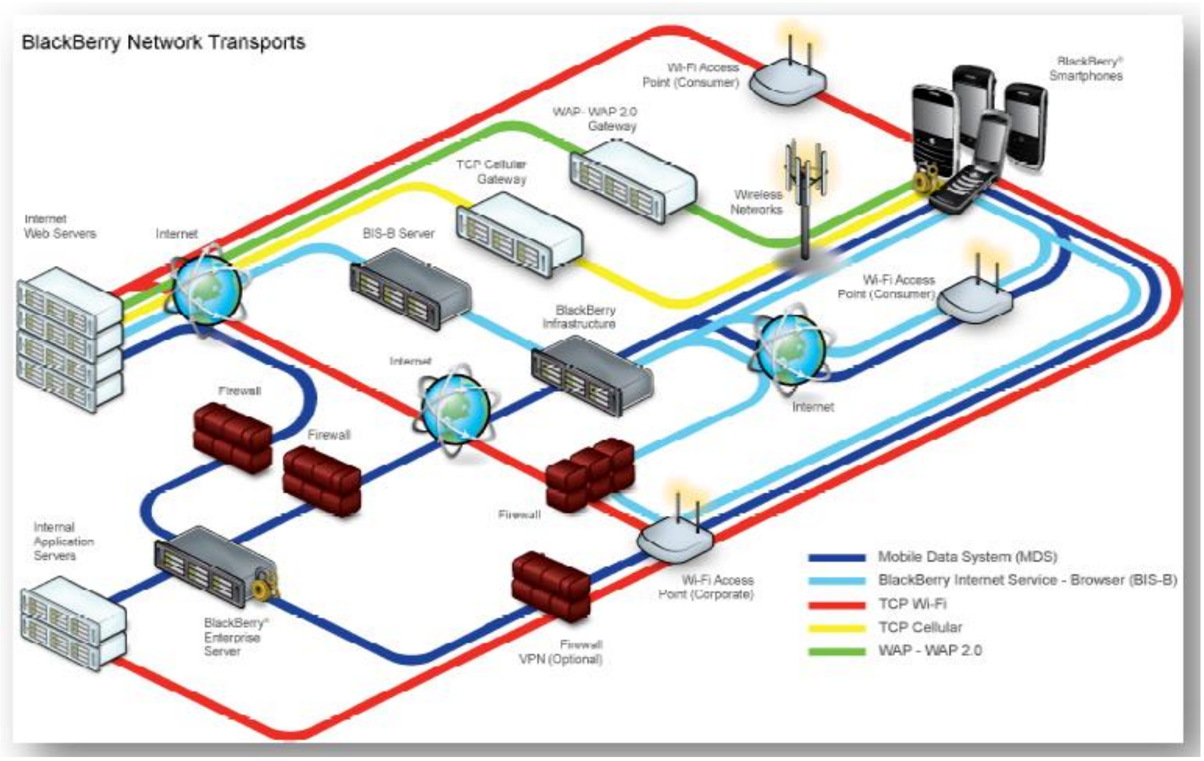


Figure 12: Blackberry Network Transports (*A14 Network Transports* 2009)

Due to the specific features offered by each communication technique, the following aspects might be explored before choosing a method.

– The category of Blackberry device users that use the application
– The required security level for transmitting the data
– The type of network utilized by the application
– The expected network traffic generated by the application
– The location of the server that data is transmitted to/from (Internet or Intranet)

The following section provides an overview on different Blackberry network transport methods based on *Network communication* 2010, *A14 Network Transports* 2009:

MDS (Mobile Data System)

As an introduction to MDS transport, first BlackBerry Enterprise Server (BES) is briefly described here. BES is a Blackberry middleware software package provided by Research In

Motion (RIM). This service is employed by organizations to supply a secure wireless gateway for their BlackBerry Smart phone members when connecting to the corporate application servers. It is installed behind an organization's firewall and utilizes MDS to allow third-party Java applications on Blackberry smart phones to create a reliable connection over the corporate firewall and access the corporate application servers or the servers on the internet.

MDS transport is only applicable for the members of the corporation that utilize BES. Through this transport method, data are transmitted between the Blackberry device and its associated BES. The data transmission is routed through either BlackBerry Infrastructure or Wi-Fi connection. The financial cost of data transmission through Wi-Fi is generally less than other transport methods. For this reason, the Wi-Fi transport is called least cost routing method.

For data transmission through BlackBerry Infrastructure, the Blackberry device utilizes connection technologies such as GPRS or 3G to connect to carrier's cell towers. Next, through the internet the data are first transmitted to Blackberry infrastructure and then to BES. The data received by BES can then be transferred to internal servers of company such as Application Server and E-mail server, or it might be transferred to Internet web servers.

Data transmission using MDS method benefits from high security features since it is encrypted through all the stages of transmission. Moreover, the data are compressed all the way.

BIS-B (BlackBerry® Internet Service - Browser)

BIS (BlackBerry Internet Service) is a service that allows small enterprises and individuals to communicate with Internet-based services such as public email or BlackBerry Messenger. BIS provides Java applications with a direct access to the internet by utilizing HTML web browser that is included in Smart phone devices. Using BIS-B transport method, data are transferred from Blackberry device to the internet by passing through BlackBerry® Infrastructure. For connection to BlackBerry® Infrastructure the same path used for MDS connection is applied. Then, the BlackBerry® Infrastructure directs the data to the requested Internet Web or other servers.

The data transmitted via this method are not encrypted; however, Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS) or Secure Sockets Layer (SSL) might be applied for a secure transmission.

One advantage of BIS-B is that the data transferred through the wireless network are compressed. Moreover, the data can automatically be transmitted through the least cost routing Wi-Fi connection, when available.

Third-party developers can apply this method for data transmission which is conditional on application approval and being a member of BlackBerry Alliance Program.

WAP 1.x (Wireless Application Protocol)

This method employs carrier's WAP gateway for transferring the data to the internet. The data is first transmitted to the cell towers and then to the WAP gateway and finally to Internet Web servers. WAP transport supports WAP 1.0, 1.2, and 2.0.

The transmission of data through WAP gateway is completely under the control of the service provider. Some service providers do not allow the data transmission through their WAP servers. Moreover, to utilize this transport method on a device, the WAP connection parameters should be set on the device by the user or by the wireless service provider. Although this transport method is supported by most of the mobile platforms, it might not be supported by all carriers and data plans.

TCP (Transmission Control Protocol) Cellular or Direct TCP

This method employs carrier's Internet gateway for transferring the data to the internet servers. The data transmission through this method is the most direct way of utilizing cellular radio. Direct TCP avoids the limitations of WAP transport by providing a direct connection to the Internet Web Servers through the cell towers.

TCP Cellular is supported by most of the mobile platforms, carriers, and data plans. In North America, service providers usually allow users to access their Internet gateway by adjusting the configuration of Blackberry devices for that. However, for using this transport the application might be examined for different service providers.

To use Direct TCP as the transport method, the access point name (APN) settings of the service provider should be set within the application or directly on the device under Options > TCP.

Wi-Fi

Using Wi-Fi transport, the data can be transferred to organizations or to the Internet Web Servers. The following connection routes might be used by a device to transfer the data to the network.

(1) By passing through consumer Wi-Fi access point, the data could be transferred directly to Internet Web Servers. Typically, this connection is free (does not incur data plan costs) and very fast since it does not pass through provider’s infrastructure and Blackberry infrastructure.

(2) By passing through corporate Wi-Fi access point and corporate firewall, the data are transferred to Internet Web Servers. This connection also avoids provider’s and Blackberry‘s infrastructure.

(3) By passing through corporate Wi-Fi access point and an optional virtual private network (VPN), the data can be transferred directly to corporate internal Application Servers, if permitted by administrators.

This transport method is applicable for many of the new Blackberry devices, but it is not supported by most of the older devices. Moreover, another limitation of this approach is that the cell phone must be connected via Wi-Fi to a network, and in most mobile environments Wi-Fi is not available.

Summary

For the purpose of acquiring travel behaviour data using smart phones, it is necessary to utilize an approach which is supported by most of the Blackberry platforms, service providers, data plans, networks and be available in most mobile environments. As explained above, WAP transport is not supported by all the service providers and Wi-Fi is not available in most mobile environments. MDS and BIS methods both have the advantage that the data transferred through the wireless network is compressed and optimized. Moreover, data transmission through MDS method benefits from high security features since it is encrypted through all the stages of transmission. However, MDS is only applicable for the members of the corporation that utilize BES. As for BIS-B method, third-party developers could apply this method for data transmission but it is conditional on application approval and being a member of BlackBerry Alliance Program. Data transmission through Direct TCP is the most direct way of utilizing cellular radio. This method is supported by most of the mobile platforms and service providers. However, the developed application should be tested through different service providers and the access point name (APN) settings of the service provider should be set by the user directly on the device under Options > TCP. The data transferred through this method is not compressed and optimized as is for the MDS and BIS methods. As a conclusion, the Direct TCP and BIS-B methods seem to be the most proper options for acquiring travel behaviour data. In the current research TCP method is utilized since it is supported by most of the service providers, data plans, and Blackberry platforms, it is available in most mobile environments, and it does not require a specific membership.

3.3 Device Side

As explained earlier, the device side is responsible for acquiring the travel data and transmitting it to the server side. This section starts with a brief description regarding the user interface of the application[1] on device side. Then it follows with the details associated with the modules and algorithms within the application.

The user interface of the application on device side contains three screens which are described below. A snapshot of each screen can also be seen in Figure 13.

Login screen

Through this screen the login operation into the system is performed. The details regarding the login process are provided within the Startup Module in the following section.

Options screen

Through this screen the settings of the application can be adjusted. This screen is mainly implemented for system administrator to change the settings of the different algorithms used

[1] The application on device side is implemented in Java language (including 2413 lines). For this application Blackberry APIs are used, which are the extensions of JSR179 Application Programming Interfaces (API) 2008.

within the application. As a result of this, the performance of these algorithms can be analyzed according to the new settings.

a) Login screen b) Options screen c) Main screen

Figure 13: Device Application User Interface

Main screen

This screen displays the current status of the application which includes the information that is frequently captured by the application, the last time data were received by the server, identification of any errors that occur within the system, the current system time, etc. The screen is continuously updated as soon as receiving the new information. The Main screen also includes the following features:

Travel mode options – This feature was enabled for the collection of a test data set. Through this feature users can label their current travel mode (i.e. walking, biking, auto, activity stop, school bus, waiting for transit). Whenever the user travel mode is changed, the user selects the new travel mode, so the true mode is known within the test data set. The travel data will then be read by the application and sent to the server in addition to other information. Next, it will be stored into the database, representing the real travel mode of the user. Finally it will be utilized for comparison with the travel mode estimated by mode detection algorithms to measure the performance of these algorithms.

Speed Simulator – This feature is added to the application to be utilized by system administrator for test purposes. Through this feature, the system administrator is able to set a new speed value to be used instead of the real speed reported by GPS. This feature is helpful when the device is stationary, but non-stationary speed data is required to test the performance of an algorithm within the application, such as State Machine (SM) algorithm. This algorithm is sensitive to a predefined stationary speed threshold and it behaves differently according to the speeds below or above this threshold, which will be discussed in the following sections. As a result, through this feature the non-stationary speed could be simulated without a need for actual movement.

Finally through the Main screen and using "Start Capturing" menu item, the process of GPS data acquisition can be started.

More details regarding the features of each screen will be discussed in the next sections. The user manual of the device application can also be found in Appendix A.

3.3.1 Startup Module

Figure 14 illustrates the flowchart of the Startup Module. As explained earlier, this module is responsible for log-in operation into the system and also adjustment of the application settings before starting GPS data acquisition.

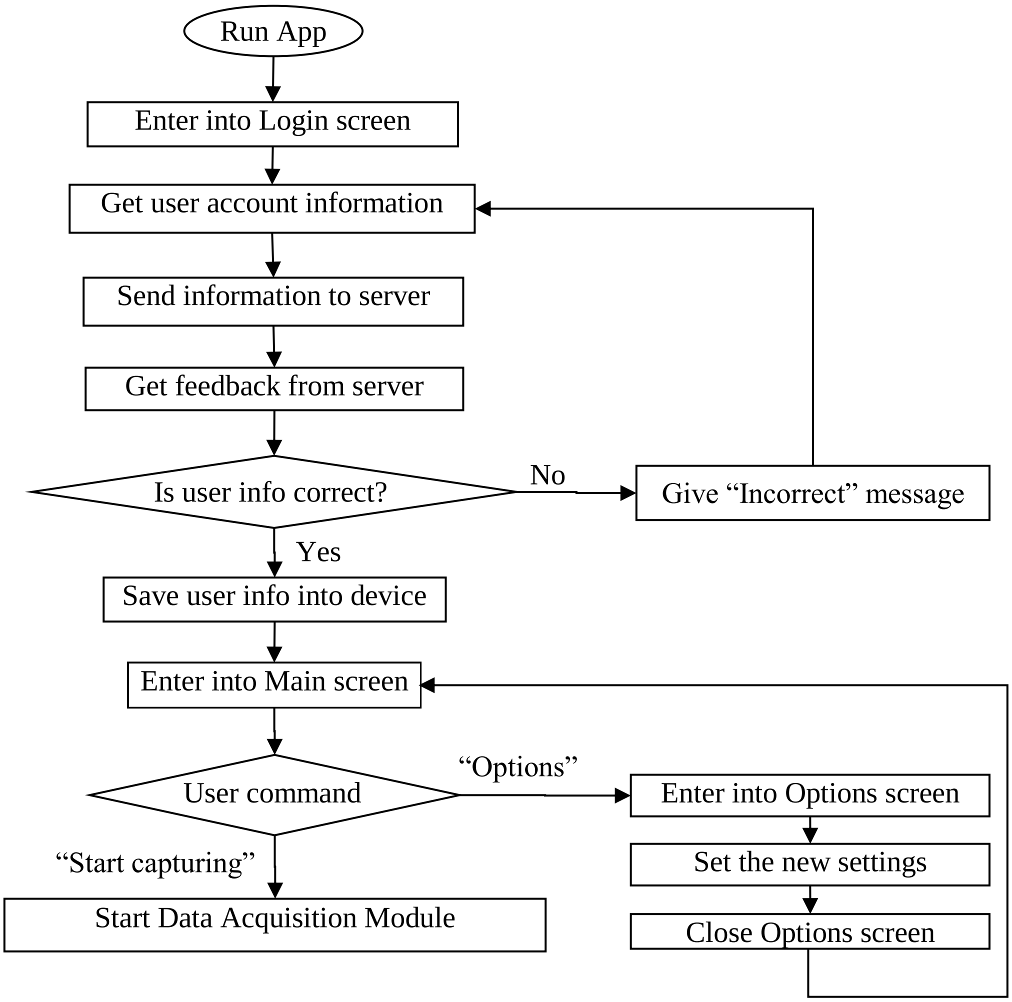


Figure 14: Login and Start Module Flowchart

Log-in process

In the implemented system users are distinguished by a unique user account (i.e. User ID and Password). The user account is used in server side to store and process the travel data of each user[1]. Identifying the users by their user account give them the ability to record their trips using any GPS-enabled Blackberry smart-phone. In other words, it eliminates the need to use a specific device or SIM card to record the travel data and users only need to download the customized application on their Blackberry device. Moreover, travelers’ privacy is protected as no personal information such as device PIN or telephone number is stored in the

[1] The users of the application must set up an account through a dedicated website, namely Intelligent Travel Analyzer, which is designed for the current travel surveying system to provide the users with the ability to track and analyze their travel patterns. Any modification for the user account must be performed through this website.

system. Therefore, in order to record the trip data by application, first the user account information is required. This information should be entered by the user through Login screen (Figure 13 a). The account information is then sent to the server for confirmation and if the information is correct, the user is entered into the main screen of the application. In this case, the user information is stored into the device memory, so that the user does not need to enter the information the next time s/he runs the application.

Setting adjustment and startup

Before starting the process of data capturing and transmission to server, the user can specify several system parameters. For example, the user can determine the frequency at which data are retrieved, the frequency at which data are transferred to the server, and the type of information to be transmitted to the server. Adjustment of these settings is performed through Options screen (Figure 13 b) which is accessible from the Main screen. The other features that can be adjusted through the Options screen will be discussed in the following sections. Once the settings of the application are adjusted, Data Acquisition Module can be started using “Start Capturing” menu item in Main screen. If the user does not change any of the settings, then the application will be started with the default settings.

3.3.2 Data Acquisition Module

This module is responsible for acquiring the travel data. Once it is started, Location Listener starts to communicate to the GPS satellites, SM algorithm starts managing the frequency at which Location Listener should capture the data, Restart Module starts calculating the time elapsed from the last captured GPS point, and CP algorithm waits for input data from Location Listener to identify critical points.

3.3.2.1 Location Listener

The role of Location Listener is to start a communication to GPS satellites and to listen for new GPS data at the prescribed frequency. In order to establish the communication with GPS satellites, different parameters are considered. Once the communication is established, different types of data might be captured through this module. The following section describes the required parameters and the captured data within this module.

Parameters

GPS mode

As explained in section 3.2, GPS mode identifies the mode by which the communication to the GPS satellites is created and GPS points are obtained. Selecting an appropriate mode is important since it affects the quality of travel data collection and battery power consumption. For the current application, by default Assisted GPS mode is utilized. The reason for selecting this method is that it reduces time-to-first-fix (TTFF) compared to Autonomous mode; it is much more accurate compared to Cell site mode; it is more sensitive in urban areas with obstructions compared to Autonomous mode; and it operates on most types of networks compared to complementary modes[1]. However, other GPS modes such as Autonomous could also be selected through "Options" screen. The impact of different GPS modes (i.e. Assisted, Autonomous and Cell site) on data accuracy and battery consumption will be analyzed later in the Results chapter.

Capture Interval (I_c)

Capture interval (I_c) identifies the frequency at which GPS data are requested by Location Listener. This parameter can significantly impact the accuracy of the data and the energy consumed by the device. While high frequencies can provide accurate travel data at the cost of significant battery consumption, low frequencies result in inaccurate data collection while saving the device energy. In order to analyze the impact of Capture Interval (I_c) on data accuracy and battery consumption, two options are considered in the current application: (i) capturing the GPS points at fixed time intervals; (ii) or at dynamic time intervals. Capturing the data at dynamic time intervals is a method developed to decrease the battery consumption while increasing the data accuracy. This method will be discussed later through State Machine (SM) algorithm. Using "Options" screen it can be determined either to use fixed

[1] Complementary GPS modes discussed in section 3.2 are not considered in this research. The reason is that although they provide optimum solutions for acquiring GPS data, they only operate on GPS smart phones with GPSOne technology and on CDMA network. Many new Blackberry devices currently support GPSOne technology; however, at the time the current research was conducted the Blackberry smart phones which operate on CDMA network were not available in Canada.

intervals by turning off the SM algorithm, or to use dynamic intervals by turning on the SM algorithm. The value of Capture Interval (I_c) also can be adjusted through this screen. As an example, when fixed time interval of 5 seconds is selected, Location Listener attempts to acquire the GPS points every 5 seconds. If dynamic capture intervals are used the changes in the value of I_c are displayed on Main screen. In this case, the minimum and maximum capture interval can be adjusted through "Options" screen; otherwise, default values will be used for minimum and maximum interval parameters which will be discussed later in SM algorithm section. The impact of capture interval (I_c) parameter on battery consumption will be analyzed later in the Results chapter.

Time-Out (C_{TO}), Max-Age (C_{MA})

Time-Out (C_{TO}) identifies a time out threshold (in seconds) for capturing the GPS points. In other words, when the GPS points are not available at the scheduled time intervals, Location Listener uses this threshold to provide the GPS points with some delay. For example, if I_c is 5 seconds and C_{TO} is 1 second, it means that new GPS point can be captured up to 6 seconds after capturing the previous GPS point. If new GPS point cannot be provided before the time out period ends, an invalid location point (i.e. a point with no valid data) will be passed into Location Listener (BlackBerry API 4.0.2).

Max-Age (C_{MA}) identifies the maximum age (in seconds) permitted for the recent location update to be used instead of a new location update. To be more precise, before capturing a new location update at the scheduled time interval, if a location update is available whose age is less than the max-age then that location point is reported instead of capturing a new location point. As an example, when C_{MA} is 10 seconds it means that if a location update is available within the last 10 seconds before the scheduled capture time, then that location is accepted as the new update instead of capturing a new one at the scheduled time (BlackBerry API 4.0.2).

The value of C_{TO} and C_{MA} parameters clearly should be less than I_c. It is also important to choose small values for these two parameters. The reason for this is that during the time specified by C_{TO} and C_{MA}, the device makes a continuous attempt to retrieve a GPS point. Therefore, if the values of these parameters are large and no GPS data are available (say because the device is inside a building and cannot receive GPS signals), a significant amount of battery is consumed. In the implemented application in the current research, the values of these parameters are dynamically set to half of the capture interval (I_c /2). Moreover, to avoid large values for these parameters when the capture interval (I_c) is large (e.g. when SM algorithm is in last state (I_c=300s) which will be described later), the values of these parameters are not allowed to exceed a maximum of 10 seconds.

Data

As described above, Location Listener makes a request for new GPS data based on the predefined Capture Interval (I_c). In addition to GPS information obtained from satellites by each request, the current application is designed to simultaneously obtain some information from the device and the user through the current module. This additional information is used for analyzing and increasing the performance of the designed travel surveying system. The complete set of data captured per each request by the current module is as follows:

Basic GPS information

The basic GPS information captured from satellites include longitude, latitude, speed (m/s), and timestamp (milliseconds). This information will be utilized in Results chapter through different experiments to analyze the accuracy of the Blackberry smart-phone in collecting the travel data and also to analyze the performance of algorithms within the device application.

Complementary GPS information

In addition to the basic GPS information, the following complementary GPS data are also acquired from GPS satellites:

- Number of Satellites: This data indicates the number of satellites used to determine the current GPS fix.
- Accuracy: This data indicates the accuracy of the current GPS point in meters. The relationship of this reported “accuracy” and the real accuracy of the position will be analyzed in Results chapter.
- Azimuth: These data indicates the heading (in degrees) of the current GPS point which is measured as the angle relative to North (θ). In this study, this parameter is used to

determine the critical GPS points which will be discussed in CP algorithm section. Then it will be used for analyzing the performance of this algorithm in Results chapter.

Computed Measures

Using the basic information obtained from GPS satellites, the following measures are computed by the application:

- Acceleration: This is the acceleration at current GPS point computed relative to the previous GPS point (m/s^2).
- Distance: This is the straight-line distance from the previous GPS point to the current GPS point (meter).

This information is utilized on the server side for extracting the trip characteristics by trip inference algorithms.

Device information

The application obtains the following data directly from the device:

- Battery Level: It shows the remaining battery level of the device in percent, which is retrieved at the moment the current GPS point is captured. This parameter is used in Results chapter to identify the performance of the application in terms of battery consumption.
- Accelerometer: It shows the accelerometer data of the device, which is retrieved at the moment the current GPS point is captured. Accelerometer data include three elements of x,y,z. Each element indicates the G-force value along the related axis (Figure 15) and is measured relative to ground. As an example, when the device is laid on a flat surface, the value along z axis (perpendicular to the ground) is equal to G-force or $9.81m/s^2$ (which is returned as 1000) and the values along x and y axes are 0. Accelerometer data may be useful for identifying the travel modes by mode detection algorithms; however, it is not supported by all Blackberry devices. Moreover, in order to obtain the accelerometer data, the accelerometer sensor must be turned on through the application to report the changes in movement. To obtain frequent accelerometer data, the sensor should be on continuously, which consumes a signification amount of battery (BlackBerry API 4.7.0). The impact of accelerometer on battery consumption is analyzed in the Results chapter.

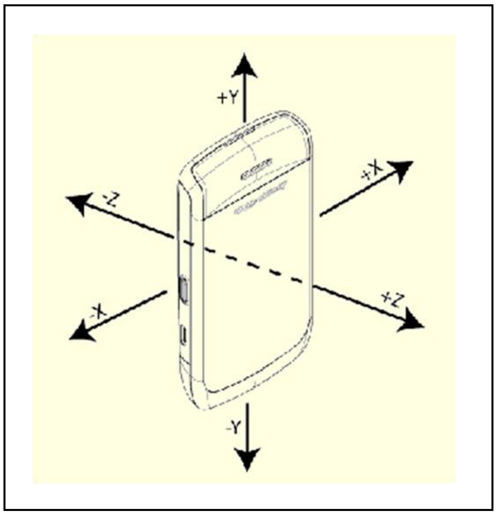


Figure 15: X, Y and Z axes along which xyz elements of accelerometer data is calculated (BlackBerry API 4.7.0)

User Input

The following is the information entered by the user to be utilized for test purposes as explained in section 3.2.

- Travel mode: It shows the current travel mode selected by user through the main screen. These data are read from the screen exactly at the moment the current GPS point is captured and it represents the real travel mode of the user at the current point. It is then utilized for identifying the performance of trip detection algorithms in server side.
- Simulated Speed: It shows the speed configured by the system administrator through the main screen. When this feature is activated by the administrator, its data is read from the screen exactly at the moment the current GPS point is captured and it is used as the speed of the current point. In this case the real speed reported by GPS is ignored. The feature is utilized in Results chapter for analyzing the performance of SM algorithm.

All the data obtained at each time interval are appended to each other to form the data of a single point. Then the data of the point is displayed on the main screen and at the same time is passed to CP algorithm to be identified either as critical point or non-critical point. This algorithm will be discussed in detail in the following sections.

3.3.2.2 State Machine (SM) Algorithm

As explained in previous sections, one of the main challenges with the smart phone based travel behavior surveying systems is the need to reduce the amount of energy consumed by the device to calculate and transmit the GPS fixes. SM algorithm is an optimization algorithm implemented in the current research to reduce the number of GPS position calculations and consequently to increase the battery life. This algorithm is adopted from the proposed algorithm by Barbeau et al. 2008. However, as discussed in the Literature Review chapter, the algorithm proposed by Barbeau et al. fails to address some problems which are improved within the SM algorithm developed in this section.

The SM algorithm aims to adjust the frequency at which GPS fixes are obtained based on the travel pattern of the user. By increasing the capture interval (I_c) between the GPS points, the device battery would then be saved. Since the required capture interval (I_c) to precisely record the user path is 4 or 5 seconds (Barbeau et al. 2008), larger intervals can be applied when the user is stationary and there is a clear view of satellites, or when GPS points are temporarily unavailable due to weak signals (e.g. in urban areas with tall buildings).

The idea is to start data capturing with a frequent time interval (e.g. every 5 sec) and if no valid points are obtained or if the user is stationary for a designated period of time, the time interval between position fixes is slowly increased until it reaches to a maximum time interval. On the other hand, if a valid point is obtained and the user is not stationary any more the time interval between position fixes is decreased and data will be captured at the most frequent rate.

Procedure

Figure 16 and Figure 17 are respectively the graph and the flowchart illustrating the procedure of the proposed SM algorithm in this research which is described below.

Each state in the SM algorithm is representative of a specific capture interval (I_c), and $I_{c\ n}$ is defined as the capture interval (I_c) at state n. The application starts at State 0 where $I_{c,n}$ is equal to the minimum capture interval ($I_{c,min}$) and GPS points are retrieved at their most frequent rate. If no valid point is obtained or if the user is stationary, i.e. speed < stationary speed threshold ($S_{th,\ s}$), for a designated period of time then the state is changed to State 1 where the GPS points are retrieved with capture interval of $I_{c,1}$.

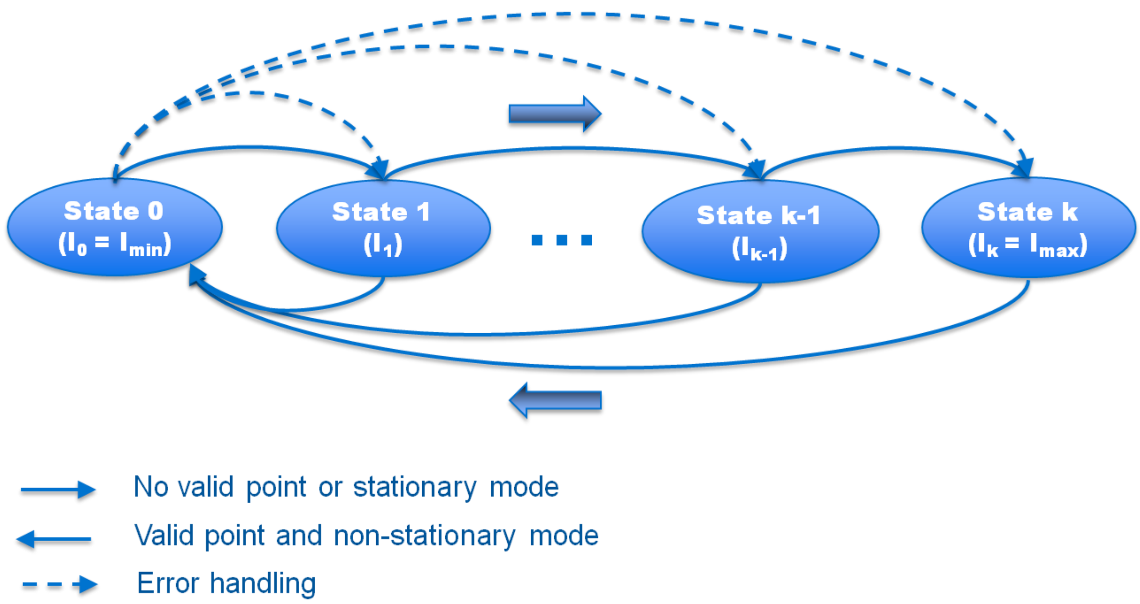


Figure 16: State Machine Algorithm

In order to determine the amount of time to be spent at each state, a counter is used. This counter counts the number of invalid GPS points (i.e. the points with no valid data) or the points with the speed less than $S_{th,s}$ at each state. If the counter reaches to a predefined state threshold (N_{th}) then a forward movement to the next state is made and the counter is reset.

Again, if no valid point is obtained or if the user is deemed to be stationary and the counter reaches to the state threshold (N_{th}), then the state machine moves forward to the next state with capture interval of $I_{c},_{n+1.}$ This process continues until the state machine reaches to its last state (State k) at which a maximum capture interval ($I_{c,max}$) is used, or until a valid point with speed higher than stationary speed threshold ($S_{th,s}$) is obtained.

If a valid point with speed higher than $S_{th\ s}$ is obtained at any state, it is very probable that the user has started to move. In this case, the state is immediately changed to the first state (State 0) where GPS points are retrieved at their most frequent rate[1]. In this way, the first issue of Barbeau et al.'s SM algorithm discussed in section 2.3 is corrected.

[1] Note that in the State Machine algorithm proposed by Barbeau et al. 2008, snapping back to the first state is performed only from the last state not from the middle states.

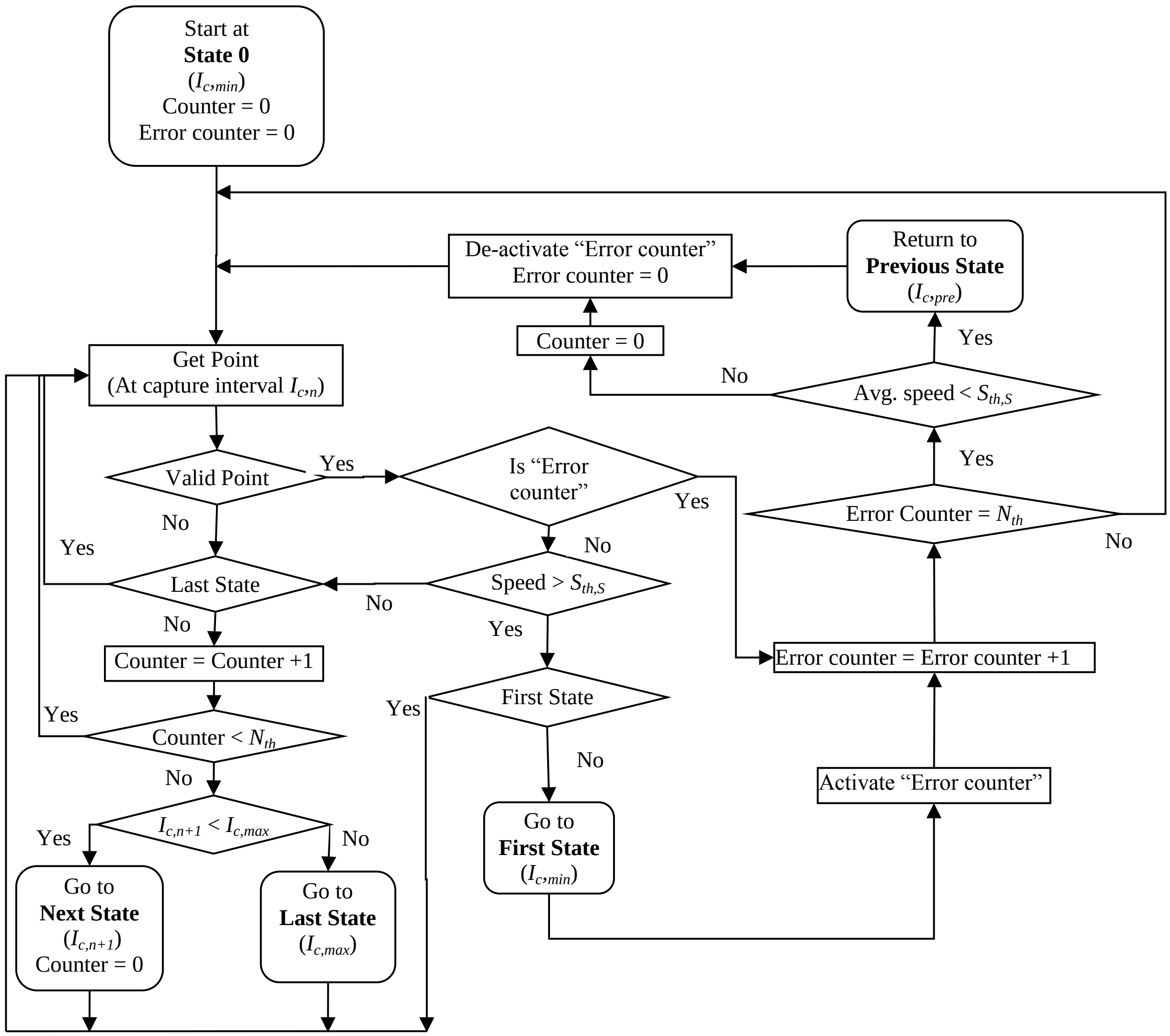


Figure 17: Flowchart of State Machine Algorithm

On the other hand, due to the errors associated with GPS, sometimes the speeds reported at stationary mode are higher than the defined $S_{th,s}$ which result in wrong decision regarding the status of the user. If this case is not recognized by the state machine, then the state is switched to the first state while the user is still stationary. As a result, the whole period of time elapsed to move forward in the state machine is wasted and the whole process should be started from the beginning. Specifically when there is a few numbers of satellites in view, the errors happen frequently, resulting in ineffective performance of SM algorithm.

Therefore, a mechanism is required to be developed for SM algorithm to handle the error occurrence. As explained earlier the SM algorithm proposed by Barbeau et al. 2008 does not handle the error occurrence. Following that, the SM algorithm implemented in this research did not initially consider the error cases. However, the poor performance of the algorithm was then identified through an experiment which will be discussed in Results chapter. Consequently, a mechanism has been developed through the current SM algorithm to resolve the error issue, which is explained here.

Whenever there is a backward movement to the first state, due to receiving a valid point with speed greater than $S_{th,s}$, the capture interval of the previous state ($I_{c,pre}$) is temporarily saved into the application. Then the new GPS points begin to be captured at the first state with highest frequency. At the same time, an error counter starts to count the captured points at this state until it reaches to N_{th} threshold. Then the average speed for the captured points is calculated. If this average value is higher than $S_{th,s}$, it is highly probable that the person has moved. Therefore, switching to state 0 was the right decision. In this case, the regular process of SM algorithm at this state is continued, and the value of $I_{c,pre}$ is ignored. If the average speed is less than $S_{th,s}$, it is identified that an error has happened in reporting the speed. In this case, by using the value of $I_{c,pre}$, the state machine directly returns back to the previous state at which the error happened and it follows its previous procedure. The advantage of this approach is that no time is wasted in the middle states in order to reach to the previous state.

As mentioned earlier, through the "Options" screen, the user can turn on the SM algorithm to retrieve the GPS points in a non-fixed interval, or can turn off the SM algorithm to retrieve the GPS points in a fixed interval. This feature is useful to analyze the performance of SM algorithm and to compare it with the fixed interval data capturing case.

The challenge with the SM algorithm is that at each state, specifically in the final states where the capture interval (I_n) is large, if the user starts to move sometime between capturing the GPS points, then a part of trip data might be missed. For example, if state machine is in the last state and $I_{c,max}$ = 5 minutes, and the user starts to move 1 minute after capturing the last GPS point, then 4 minutes of the travel data will not be captured. One possibility to resolve this issue might be using data from the accelerometer to determine if the users have begun to move. However, in the next sections it will be discussed that using accelerometer increases the battery consumption.

Parameters

This section provides a summary description of the parameters applied in the SM algorithm as well as the default values used for these parameters.

– $I_{c,min}$: minimum capture interval = 5 sec

This parameter identifies the most frequent rate for capturing the GPS points which is used in the first state (State 0). Based on literature, accurate route detection needs minimum interval of 4 to 5 seconds (Barbeau et al. 2008). Therefore, 5 seconds is used as the default value for this parameter. However, this value can be adjusted through "Options" screen

– $I_{c,max}$: maximum capture interval = 5 min

This parameter identifies the least frequent rate for capturing the GPS points which is used in the last state (State k). Large values for $I_{c,max}$ might result in large data loss when moving from the last state to the first state (i.e. when person starts moving after being stationary for a long period of time). Therefore, 5 minutes is used as the default value for this parameter. However, this value can be adjusted through "Options" screen

– $I_{c,n+1}$: capture interval at state n+1 = $I_{c,n} \times 2$

$I_{c,n}$ is the capture interval at state n. Excluding the last state, it is assumed that the capture interval at state n+1 be equal to twice the capture interval at state n. The reason for this is that small increase in capture interval results in large number of states. As a result, it will take a long time to reach to last state and energy saving will not be efficient. On the other hand, large increase in capture interval results in poor accuracy in user's path.

– $I_{c,pre}$: capture interval at previous state

This parameter identifies the capture interval at the last state from which the movement to the first state is made due to receiving a point with speed higher than $S_{th\ s}$. In this case, the value of $I_{c,pre}$ is temporarily saved into the application to be used for handling the speed errors. If it is identified that the movement to the first state has been due to error then $I_{c,pre}$ is used to return to the previous state.

– N_{th}: state counter threshold = 10 points

This parameter identifies a threshold for the number of continuous points captured in a state which are either invalid points (i.e. the points with no valid data) or are the points with speed less than $S_{th,s}$. Since capture interval at each state is different, the time taken to capture N_{th} points depends on the state and is not a static value. Choosing a large threshold value for N_{th} results in spending a long time at each state. Consequently, reaching to the last state takes too long and energy saving will not be efficient. Choosing a small threshold value for N_{th} results in data loss specifically when the person is temporarily stationary or is indoors. This parameter is also applicable for handing the speed errors in state machine. In this case, when state machine moves to the first state due to receiving a speed larger than $S_{th,s}$, all the captured points at this state are counted regardless of their speed value until N_{th} points are counted. Then, the average speed for the counted points is calculated. If the average speed is higher than $S_{th,s}$ then it is identified that the person has moved, otherwise it is identified that error has happened.

- $S_{th,s}$: stationary speed threshold = 0.75 m/s

  This parameter is used to distinguish the stationary and walking mode. Since the normal walking speed is 1.2 m/s, initially 0.5 m/s was considered as the threshold value to detect stationary state. According to different data samples captured for stationary mode it was identified that the GPS reported speed data contain some errors and that even when the person is stationary, the reported speed might exceed 0.5 m/s; however, it usually does not exceed 0.75 m/s. This will be discussed more through one of the experiments presented in the Results section. Therefore, 0.75 m/s is considered as the default value for this parameter. Different threshold values might be considered based on the sensitivity of the device. For example, in the conducted experiments, Blackberry Bold 9700 is generally more accurate than Blackberry Torch 9810 and its average stationary speed is generally lower.

3.3.2.3 Restart Module

When the user is in a location where no GPS point is available for a period of time, the communication with the satellites is automatically terminated by the device's built-in GPS to save the battery. In this case, the communication is not established again until it is internally restarted through the application. The APIs by Blackberry recommend not to restart the communication in less than 3 minutes, because it might take up to 3 minutes (i.e. TTFF) to retrieve the first fix after disconnection.

Therefore, a restart module is implemented in the current application to continuously check the time elapsed from the last captured GPS (Δ_t). If Δ_t exceeds a gap threshold (G_{th}), first the connection is restarted, and then an attempt is made to obtain a new GPS point. This attempt continues until the point is obtained or for a maximum of 3 minutes (TTFF). If a GPS point can be captured during these 3 minutes, then the regular process of data capturing will be followed to capture the GPS points at the defined capture interval (I_c). On the other hand, if no GPS point is available during these 3 minutes, Restart Module waits for the time gap of G_{th} and again restarts the communication and attempts to obtain a new GPS point for 3 minutes. This process continues the whole time that the application is running.

When the GPS signals are not available and the device attempts to obtain a GPS point continuously for the duration of 3 minutes (TTFF), significant amount of battery is consumed. Therefore it is important to choose a reasonable time gap (G_{th}) between these 3 minutes. By default 5 minutes is selected for G_{th}; however, the value of this parameter can be adjusted through the "options" screen. The reason for selecting 5 minutes for G_{th} is that choosing larger values for this parameter might result in data loss if GPS becomes available. On the other hand, selecting a small value for G_{th} results in relatively high battery consumption. Moreover, if the value of G_{th} is very small it might affect the performance of the SM algorithm which can be described as follows.

Initially, G_{th} was assumed to be zero and as a result TTFF was the only parameter considered for restarting the connection. In this condition, if no GPS point is captured within 3 minutes then the Restart Module assumes that GPS is not available, and therefore the connection is restarted after 3 minutes. This condition results in improper performance of the SM algorithm when the capture interval (I_c) is greater than 3 minutes. In this case, although the delay in obtaining a GPS point is due to the large capture interval, Restart Module restarts the connection after 3 minutes. As a result, the SM algorithm cannot reach to the states with I_c greater than 3 minutes and the data capturing starts from the first state of the state machine ($I_{c\,=}$ 5 seconds). This condition will be discussed more through one the experiments in the Results chapter.

3.3.2.4 Critical Point (CP) Algorithm

Critical Point (CP) algorithm is another optimization algorithm implemented in order to increase the battery life by reducing the number of points transmitted to server. This algorithm aims to identify the critical points on the user's trajectory and discard non critical points. Finally, only the points which are identified as critical points by this algorithm are transmitted to the server. In order to identify the critical points, the data of each single point obtained from Location Listener module is entered into this algorithm to be labeled either as a critical point or a non-critical point. The output of this algorithm is then entered into the Data Queue which is discussed in the next section. The CP algorithm implemented in this research is adopted from the algorithm proposed by Barbeau et al. 2008. However, some modifications[1] and improvements are made to the algorithm which are discussed here. This section starts by describing the criteria used by this algorithm to identify a point as critical point versus non critical point, and then proposes a method for measuring the error introduced on user's trajectory as the result of applying the CP algorithm.

Critical Points and Non Critical Points

Critical points are the points that are crucial in constructing the user's path (e.g., the points that are captured while the subject is moving and changing direction). In order to detect these points, the change in direction is computed each time a new GPS position is obtained. The speed of the point is also analyzed to identify if the subject is moving or is stationary. If the change in direction exceeds a predefined angle threshold (θ_{th}), and the speed is higher than a predefined stationary speed threshold ($S_{th,s}$), then the location is labeled as a critical point. When a GPS point is captured, the heading, measured as the angle relative to North (θ_i), is also reported (Figure 18). The change in heading ($\Delta_{\theta,\, i}$) is computed as the absolute difference

[1] In the CP algorithm proposed by Barbeau et al. 2008 each critical point is transmitted to the server as soon as it is identified by the algorithm. Therefore, each transmission contains a single point. In the method proposed in the current research, critical points are queued on the device and transmitted to the server at a fixed transmission interval. Although the purpose of both algorithms is to reduce the number of points transmitted to server, in Barbeau et al.'s method the number of transmissions to server is reduced, and in the current algorithm the number of points in each transmission is reduced.

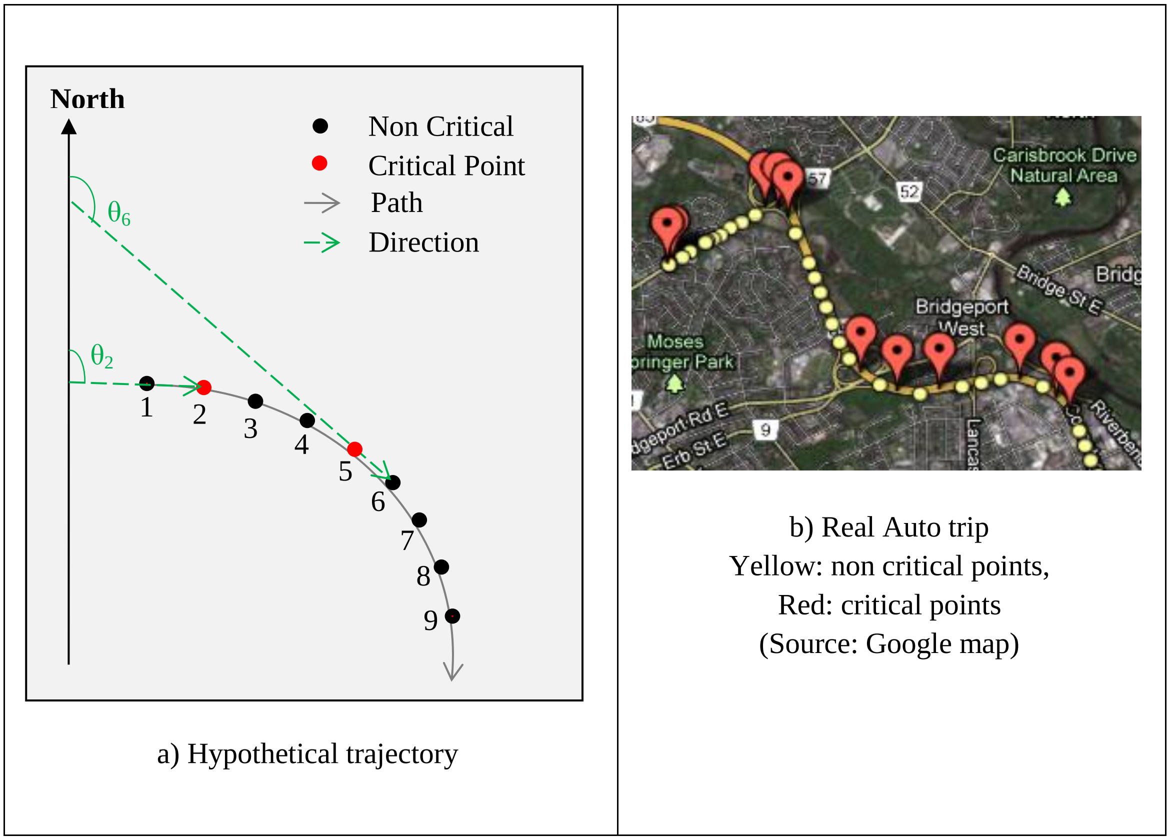
in the heading associated with the current GPS point and the heading associated with the previous critical point (θ_{cr}).

$$\Delta_{\theta,\,i} = |\theta_i - \theta_{cr}| \qquad \textbf{Equation 1}$$

If $\Delta_{\theta,\,i}$ exceeds an angle threshold (θ_{th}) and speed is greater than $S_{th,s}$, then the previous point GPS point (Point $_{i-1}$) is considered as a critical point and $\theta_{cr} = \theta_{i-1}$. The first GPS fix by default is also considered as a critical point by CP algorithm.

The hypothetical trajectory segment shown in Figure 18a, illustrates the application of the angle threshold in the CP algorithm. Assume the most recently obtained point is point 6. The heading associated with point 6 is identified by the dashed green line and is labeled as θ_6. The most recently labeled critical point is point 2, which has a heading of θ_2. The change in heading is determined as $\Delta_{\theta,\,6} = |\theta_6 - \theta_2|$ which is found to be larger than θ_{th} and therefore point 5 is labeled as a critical point. The application of the angle threshold to a real auto trip is also illustrated in Figure 18b. The value of this parameter, θ_{th}, can significantly affect the performance of the CP algorithm which will be analyzed in detail in Results chapter.

As explained earlier, in the CP algorithm proposed by Barbeau et al. (see section 2.2.1) the change in direction is always calculated using the last three captured points. In this way, if the user is traveling on a circular path with a large diameter, the change in direction might never exceed the θ_{th} (see section 2.3). However, as mentioned above, in the CP algorithm proposed in this research, the change in direction for each point is calculated relative to the last critical point. Therefore, even if the user is traveling on a circular path with a large diameter, the change in direction will exceed θ_{th} at some point.



a) Hypothetical trajectory

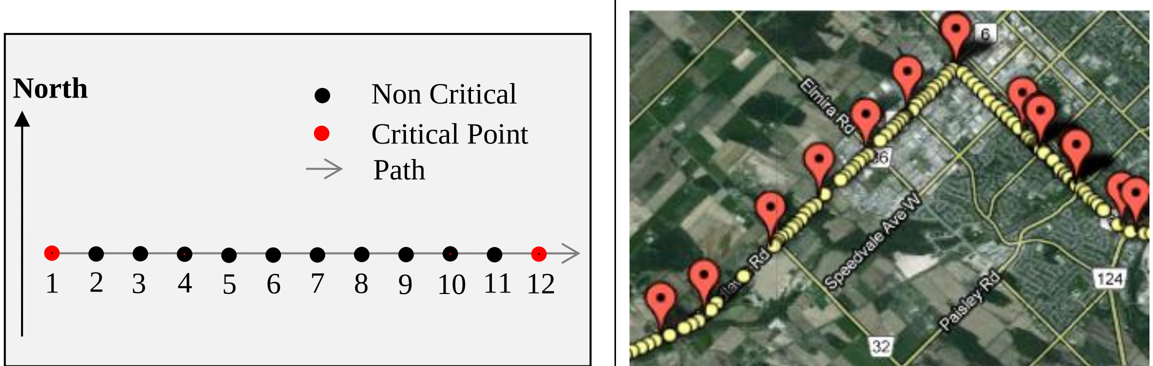
b) Real Auto trip
Yellow: non critical points,
Red: critical points
(Source: Google map)

Figure 18: Identifying critical points on curves and turning points based on angle-threshold

In contrast to critical points, non-critical points are the points that do not contribute to the user's path. For example, most of the points captured on a straight line or in stationary mode are not critical points and should be filtered by the CP algorithm. However, in order to avoid long periods in which no location data are transmitted to the server, a second criterion is introduced. If the time since the last location was labeled as a critical point ($t_i - t_{cr}$) exceeds a time threshold (t_{th}), then the current point is labeled as a critical point.

Selecting an appropriate time threshold is important in increasing the functionality of the CP algorithm. As the time threshold value increases, the number of points labeled as critical decreases and battery life is increased; however, fewer critical points may also negatively affect the ability to accurately represent the user's trip data. A detailed analysis of the impact of time threshold (t_{th}) parameter on the performance of the CP algorithm is presented in the Results chapter.

Figure 19 illustrates the influence of the time threshold within the CP algorithm. Figure 19a illustrates a hypothetical trajectory segment. Since the trajectory is straight, the application of the angle threshold does not result in the designation of any of the points as critical points. Assume point 1 has been labeled as a critical point. When point 12 is obtained, the time that has elapsed since the last critical point (point 1) was obtained exceeds the time threshold t_{th} and therefore point 12 is labeled as a critical point. Figure 19b illustrates the application of the angle threshold and time threshold to a real auto trip.



a) Hypothetical trajectory

b) Real Auto trip
Yellow: non critical points, Red: critical points (Source: Google map)

Figure 19: Identifying critical points on straight lines based on time-threshold

In the CP algorithm proposed in this study, θ_{th} is only applied to vehicular modes and for walking mode a time threshold ($t_{th,w}$) is considered to identify the critical points. The reason for this is that since GPS positions are obtained at a prescribed frequency, the accuracy of the reported heading is better when the traveler is moving at a higher speed (e.g., auto) than at a lower speed (e.g., walking). The time threshold for walking mode ($t_{th,w}$) is assumed to be equal to 15 seconds, since large values of $t_{th,w}$ results in missing many of the GPS points.

In order to distinguish between vehicular and non-vehicular modes, the speed value of GPS points is examined. By default the following speed threshold values are used to identify the travel mode of the subject:

– $S_{th,s}$: Stationary speed threshold = 0.75 m/s

– $S_{th,w}$: Walking/running speed threshold = 5 m/s

Then the travel mode is identified as follows:

– if speed $\leq S_{th,s}$ then mode is stationary

– if $S_{th,s} <$ speed $\leq S_{th,w}$ then mode is walking/running

– if $S_{th,w}$ < speed then mode is vehicle (biking, auto, transit)

The flow chart for the Critical Points algorithm can be seen in Figure 20, which illustrates how critical points are identified according to the criteria mentioned above.

In addition to the criteria described in this section, other factors are also considered by the current CP algorithm to identify the critical points. As an example, the first point captured after each switch between stationary and non-stationary modes is considered as a critical point. As a future consideration, the points at which significant changes in speed happen could also be labeled as critical points because these points may represent changes in mode (e.g. walking to auto).

Although the goal of implementing the CP algorithm is to transmit only the critical points to the server in order to decrease the battery consumption rate, a feature is provided through the Options screen which allows the user to only transfer critical points to server, or to transfer all the points (critical and non-critical) and to label the critical points. This feature is useful for different analysis purposes in server side. For example, the impact of CP algorithm on the performance of trip inference algorithms can be analyzed by comparing the output of inference algorithm when all the points are used as input versus the time when only the critical points are used as the input. Moreover, using this feature, the number of bytes saved through CP algorithm can be measured. For the first case when only the critical points are transmitted to server, only these points are passed into the Data Queue. For the other case, all the points are passed into the Data Queue.

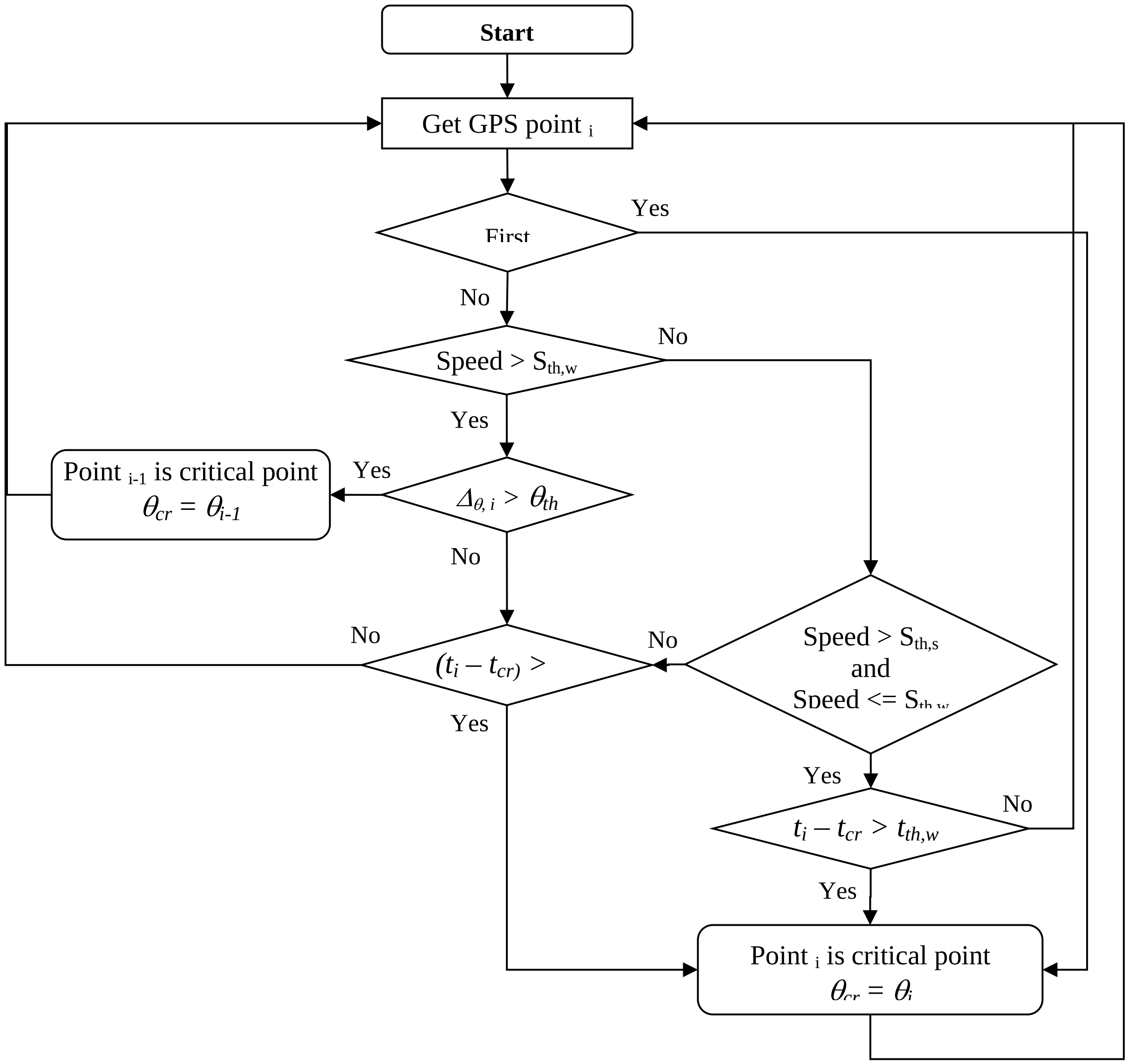


Figure 20: Flow chart of Critical Points Algorithm

Error Measurement

Since the CP algorithm discards some of the data points of the user's travel data, it is important to analyze the impact of this algorithm on the ability to reproduce the user's trajectory. This section proposes a method for measuring the error associated with reproducing the user's trajectory as a result of applying the CP algorithm. This error is quantified as illustrated in Figure 21. In this figure, the points represent all of the GPS location measurements obtained by the smart phone. The solid black line joins each consecutive location and represents the user's trajectory. The solid red points are locations labeled by the CP algorithm as critical points and are forwarded to the host server. The solid black points are non-critical points and therefore are not sent to the server. Consequently, recreating the trajectory based only on the critical points results in the dotted red line. The error in this recreated trajectory is computed as:

$$E = \sum_{i=1}^{N} e_i$$ **Equation 2**

Where:

E	=	total error associated with the trajectory (m)
i	=	location reference index
N	=	number of location points contained in the trajectory
e_i	=	minimum distance between location point i and chord $L_{j,k}$ ($j \le i \le k$)
$L_{j,k}$	=	chord from location j to location k

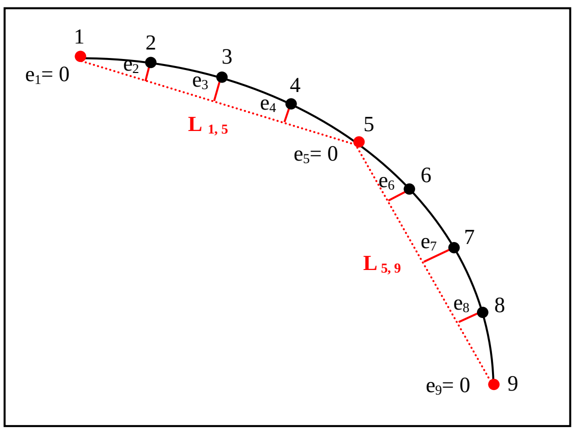


Figure 21: Calculation of the error in capturing the user's trajectory

In order to calculate individual e_i, the longitude and latitude data of the GPS point should first be converted to Cartesian ECEF[1] coordinates (XYZ). Then the shortest path of location point i to chord $L_{j,k}$ ($j \le i \le k$) should be computed.

The following formulae illustrate the conversion of longitude and latitude to xyz based on WGS84 model (Source: *The astronomical almanac* 2007).

Given latitude,

[1] Earth-Centered, Earth-Fixed (ECEF) is a Cartesian coordinate system which represents positions as X, Y, and Z coordinate ("ECEF," *Wikipedia*).

$$C = \frac{1}{\sqrt{\cos^2(latitude) + (1-f)^2 \sin^2(latitude)}}$$ **Equation 3**

$$S = (1-f)^2 C$$ **Equation 4**

Then ECEF coordinates are:

$$x = (aC + h)\cos(latitude)\cos(longitude)$$ **Equation 5**

$$y = (aC + h)\cos(latitude)\sin(longitude)$$ **Equation 6**

$$z = (aC + h)\sin(latitude)$$ **Equation 7**

Where, a is "equatorial earth radius" (6378137 meters), and f is "flattening parameter" (which identifies the shape of a polar cross-section ellipse). Since the reciprocal of f is almost an integer, commonly $1/f$ (298.257224) is presented, rather than f. h is "altitude above the reference ellipsoid", which might be assumed as zero when altitude data is not considered. Finally, latitude and longitude are geodetic latitude and longitude (radians).

The minimum distance between location point i and chord $L_{j,k}$ ($j \le i \le k$) is calculated based on the 3-Dimensional point-line distance formulation (Weisstein 2012) as illustrated in Figure 22 and Equation 8.

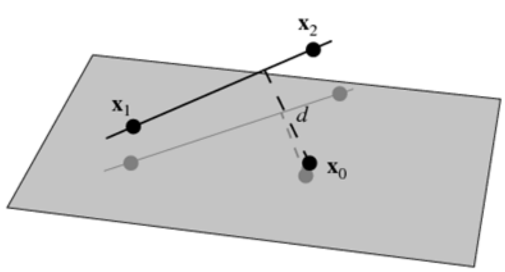


Figure 22: 3-Dimensional Point-Line Distance (Weisstein 2012)

$$d = \frac{|(X_2 - X_1) \times (X_1 - X_0)|}{|X_2 - X_1|}$$ **Equation 8**

Where

$X_0 = (x_0, y_0, z_0)$: the location of GPS point i,

$X_1 = (x_1, y_1, z_1)$: the critical point which represent j in chord $L_{j,k}$,

$X_2 = (x_2, y_2, z_2)$: the critical point which represents k in chord $L_{j,k}$,

and × denotes the cross product

3.3.3 Data Queue

Each data point produced as the output of the CP algorithm is queued into the Data Queue to be transferred to the server by Server Connection and Data Transmission Module. In other words, the data of the single points obtained from the previous steps are appended to each other at this stage and form a group of data (a set of GPS points). Finally, the data is transmitted to server as a group rather than a single point, because transmitting the single data points separately is too costly in terms of battery consumption and network overhead.

3.3.4 Server Connection and Data Transmission Module

The responsibility of this module is to create connections to the server and to transmit the data over the network. The following section describes the parameters that should be considered through this module in order to establish the connection and to transfer the data. Then a brief description is given on the type of data transferred to the server and the procedure followed when the data is transferred.

Parameters

Network Transport method

As explained in section 3.2.2 different methods can be used by a Blackberry device to connect to a server and to transmit the data over network. The method employed in the current research is Direct TCP. The reasons for selecting this method are: (1) data transmission through this method is the most direct way of utilizing cellular radio, (2) it is supported by most of the mobile platforms, (3) it employs carrier's Internet gateway for transferring the data and most service providers allow users to access their Internet gateway, (4) it does not need any membership or subscription to Blackberry services such as the ones needed by MDS and BIS-B methods, and (5) it avoids the limitations of WAP transport. However, since carrier's Internet gateway is employed for this method, the access point name (APN) settings of the service provider should be set within the application or directly on the device under Options > TCP.

Transmission Interval (I_t)

Transmission Interval (I_t) identifies the frequency at which data is transmitted to the server. At the defined transmission interval (I_t) the current module checks the Data Queue for any possible data. If there are any data in the queue, it requests a connection to the server side and transmits all the queued data. Selecting an appropriate value for this parameter is very important since it has a significant impact on device battery consumption. The impact of transmission interval (I_t) on battery consumption will later be analyzed in Results chapter. The value of this parameter could be adjusted through the Options screen.

Data

When the first data group (first set of GPS points at each application run) is transmitted to the server, the Server Connection and Data Transmission module appends some additional information to the data group which is briefly described below. This additional information is attached only to the first group of data since it is constant throughout the application run-time.

- Device model: This data shows the model of Blackberry device on which the application is being run. The device model is retrieved directly from the device by the application. This data will then be used in server side to compare the performance of different devices.
- GPS mode: This data identifies the GPS mode being used to obtain the position data. It will then be used in server side to compare the impact of different GPS modes on the performance of application.
- Software version: This data shows the current version of the application, which will then be used to track the performance of different versions of the application.

Once the data are successfully transmitted to the server, the data are removed from Data Queue. On the other hand, if any problem occurs during connection or data transmission procedure, the data that have not been successfully transmitted will not be removed from Data Queue. Then, at the next transmission interval another attempt will be made to transmit the failed data in addition to any newly captured data. This procedure continues until the problem is solved.

When the first group of data is successfully transmitted to the server, a unique ID (TripRunID) is returned back from the server to the device. This ID is allocated by the server to the current application's run to identify all the back and forth transmissions between that run and the server. Each time the application is run, there is a number of specific characteristics such as the user running the application, the device model used for running the application, the date/time that the application is being run, and the data captured through the application, which are associated with the travel behavior data. The TripRunID is used as a representative of all these characteristics, and all the data stored in the server side are identified by this ID. This ID remains constant for the entire time that the application is running and it will be changed if the application is closed and started again.

Moreover, when each group of data is successfully transmitted to the server, a confirmation message is displayed on the Main screen which shows the time the data was received by the server.

Finally, whenever the user closes the application, all the data queued after the last transmission are immediately transmitted to the server.

3.4 Server Side

As explained earlier, the application on the server side is responsible for accepting the connection requests from the device side, receiving the data transmitted, and storing the data into a database. The following sections describe the details associated with the components within the server side.

3.4.1 Connection Listener and Data Processing Module

The whole operation within the application on the server side is carried out within this module, which can be described through the following sub modules:

3.4.1.1 Connection Listener

The role of this module is to continuously listen for a connection request from the device side. Once a connection request is made from a device, this module accepts the connection and receives the transmitted data. Then it creates a thread, namely Data Processing and Storing Thread, to process the received data. Once the Connection Listener module has passed the data to this thread for processing, it closes the connection and continues to listen for next connection request and data transmission. Therefore, by transmitting each group of data from device side a new connection is created between the device and the server, and a new thread is allocated by Connection Listener to process the transmitted data. Different threads can work simultaneously on the data received from different devices to accelerate the whole process. While the threads are processing the data, the Connection Listener is listening for a new connection request.

3.4.1.2 Data Processing and Storing Thread

When the thread receives the data, it starts to split the data into the single data components that have been appended to each other in the device side. As explained earlier, each time the application is started, first the log-in information of the device is sent to the server for confirmation and then the data captured through the application are transferred to the server in the form of data group. Therefore, the data received by the current thread could be log-in information or a group of GPS data. In the first case, the data is parsed into the UserId and Password. Then the thread makes a request to the database to search for the UserId and Password, and if it is found in the database and the user account is not disabled or expired, a confirmation message is sent back to the device; otherwise, a fail message will be returned. In the second case, the additional information (i.e. Device model, GPS mode and software version) is first extracted from the data if the data are related to the first group. Then, the group data is parsed into the individual location points. Finally, each single point is parsed into the single measures of the point, such as longitude, latitude, speed, battery level, etc. At the end, the thread creates a connection to the database and stores all the extracted data into the database. Moreover, if the data are related to the first group received from the device application, the related thread requests a new TripRunID from the database and allocates it to that application’s run. After that, all the data groups received from that application’s run are identified and stored by that TripRunID.

3.4.2 Database

This section briefly describes the main tables of the SQL database implemented for the current research in order to store the data received from the device. The structure of the database can be seen in Figure 23.

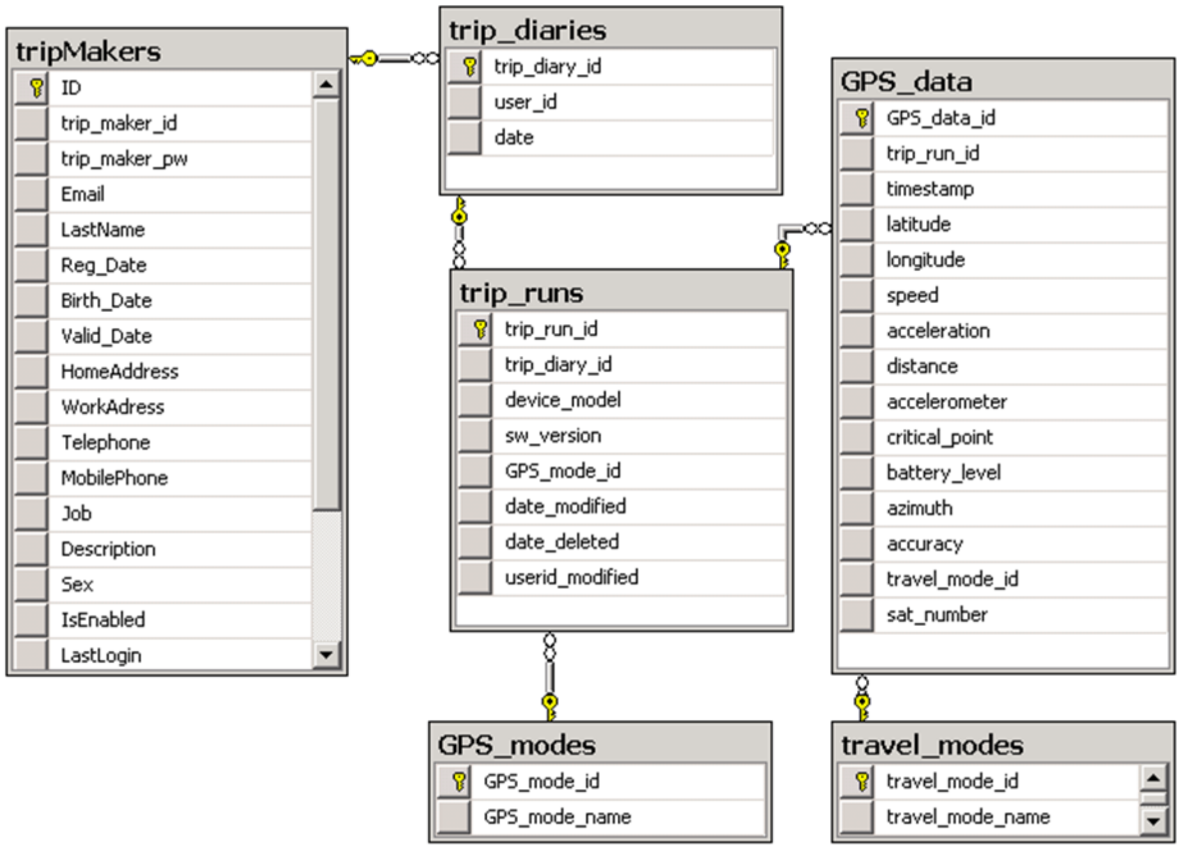


Figure 23: Database structure

The main tables within the database can be described as follows:

Trip Makers Table - This table stores all the information related to the trip maker (i.e. the user of the application). Only the UserId and Password are the required information for this table.The other information such as name, address, email are optional information that can be provided by the user when registering for an account. In this table each user is identified by a unique Id.

Trip Diaries Table – This table identifies the daily trips of the users. For all the trips made by a user in a specific day there would be a single record in this table which is identified by a unique id (trip_diary_id). Each user in Trip Maker Table can be related to different records in Trip Diaries Table, where each record represents the trips made in a single day.

Trip Runs Table - This table identifies different times that an application is run by a user during a single day. Therefore each trip_diary_id in Trip Diaries Table can be related to defferent records in this table, where each record represents an application run. In other words, for each single day a user could have different records in this table. Each record in this table is identified by a unique id (Trip_run_id). Trip_run_id is infact the TripRunID that was explained earlier in this chapter and which is used for back and forth transmissions between the device application and the server. Some additional information is also stored in this table such as the information related to each run of the application, i.e. sw_version, GPS_mode_id, device_model which are Software version, GPS mode, and Device model, respectively.

GPS Data Table – This table includes all the information related to each single point such as the GPS information captured from satelites, e.g. longitude, latitude and speed, or the information retrived by the application when capturing that point, e.g. battery level, accelerometer data, and travel mode. Therefore, for each single point a single record is allocated in this table. Consequently, each Trip_run_id in Trip Runs Table could be related to several records in this table.

Chapter 4
Results

This section first provides a preliminary investigation on the reliability of the developed system in recording the accurate travel data[1]. Next the performance of the proposed optimization algorithms (i.e. SM algorithm and CP algorithm) is assessed. Finally, an analysis is conducted on the battery consumption rate of the developed device application. For this purpose, different experiments have been conducted for each part using the developed application on the Blackberry device. Then the results of the experiments are discussed in the associated section.

4.1 Preliminary Investigation

4.1.1 Accuracy of position and speed reported by Blackberry smart-phone

The purpose of this experiment is to investigate the accuracy of the Blackberry smart phone in recording the position data and speed. For this purpose, the position accuracy and the speed reported by Blackberry smart-phone are compared to dedicated GPS logger in walking mode and stationary mode.

Walking Mode

For the walking mode, one Blackberry Bold 9700 (running the application described in the previous chapter) and two GPS loggers were employed simultaneously to capture GPS data. One of the loggers has been configured to capture the GPS positions at 1 second Capture interval (I_c) and the other logger was configured to capture the GPS positions at 5 seconds Capture interval (I_c). The Capture interval (I_c) for the Blackberry device was also set to 5 seconds. Then a trial walk was conducted for about half an hour in which Autonomous GPS mode was used to collect the GPS data.

The path of the recorded trial walk by the three devices can be seen in Figure 24. Figure 25a to Figure 25e also show the zoomed-in graphs of the trial walk. All the figures show that

[1] The appropriate position accuracy level in transportation analysis conducted in this research is considered to be the level at which activity locations and the alignment of trip makers paths are identifiable.

the Blackberry smart phone performed well in terms of recording the accurate positions, since the captured positions are very close to the true path (i.e. the associated walking path on MapPoint network) and no significant outliers are observed in the captured data. On the other hand, Figure 25a to Figure 25c illustrate the outliers present in the data collected via the GPS loggers. The reason for this might be loss of signal or long start up time. The loggers needed to be turned on some time prior to starting the data collection in order to connect to GPS satellites. In general, the figures show that the Blackberry device performed better than the GPS loggers. However, it is not possible to quantify the magnitude of the position errors as the true positions are not known.

Figure 26 illustrates also the speed comparison results for this experiment. The figure shows that the average speed of GPS positions captured by Blackberry device is 5.12 km/h which is between the average speed recorded by 5 seconds GPS logger (4.96 km/h), and 1 second GPS logger (5.43 km/h). Moreover, the 95% confidence limit for the mean speed of each device is presented in Table 3. This table shows that the confidence limit of 1 second logger does not have any overlap with the ones from the other two devices, and also the mean speed of 1 second logger is larger than the mean speed of both Blackberry device and 5 second logger. On the other hand, the confidence limit for the mean speed of Blackberry device and 5 second logger overlap in some part. Therefore, we cannot conclude that the mean speed of the two devices is different. In order to verify these results, a speed mean comparison test has also been carried out between the loggers and the Blackberry device (details are provided in Appendix D):

H_0: $\mu_1 = \mu_2$ H_1: $\mu_1 \neq \mu_2$

μ_1: Mean speed by Blackberry

μ_2: Mean speed by logger

Table 4 and Table 5 illustrate the results of these comparison tests. Based on these results, with 95% confidence we can conclude that the mean speed of 1 second logger and Blackberry are statistically different, because the null hypothesis is rejected. Also, with 95% confidence we accept that the mean speed of 5 second logger and Blackberry are not statistically different, because the null hypothesis was not rejected.

These results are somewhat surprising. The two GPS loggers used in the experiment are identical and the only difference is the frequency at which the GPS locations are obtained.

However, from the statistical analysis and the plots, it is apparent that the data collected at 1 second frequency are subject to larger errors (e.g. Figure 25c) than the data collected at 5 second frequency. The reason for these larger errors is not known.

Nevertheless, based on the visual check from the mapping and the speed comparison between the devices, it can be concluded that the accuracy of the Blackberry device in capturing the GPS positions appears to be sufficiently high for the purpose of acquiring travel behavior data and is comparable to data collected via a dedicated GPS logger.

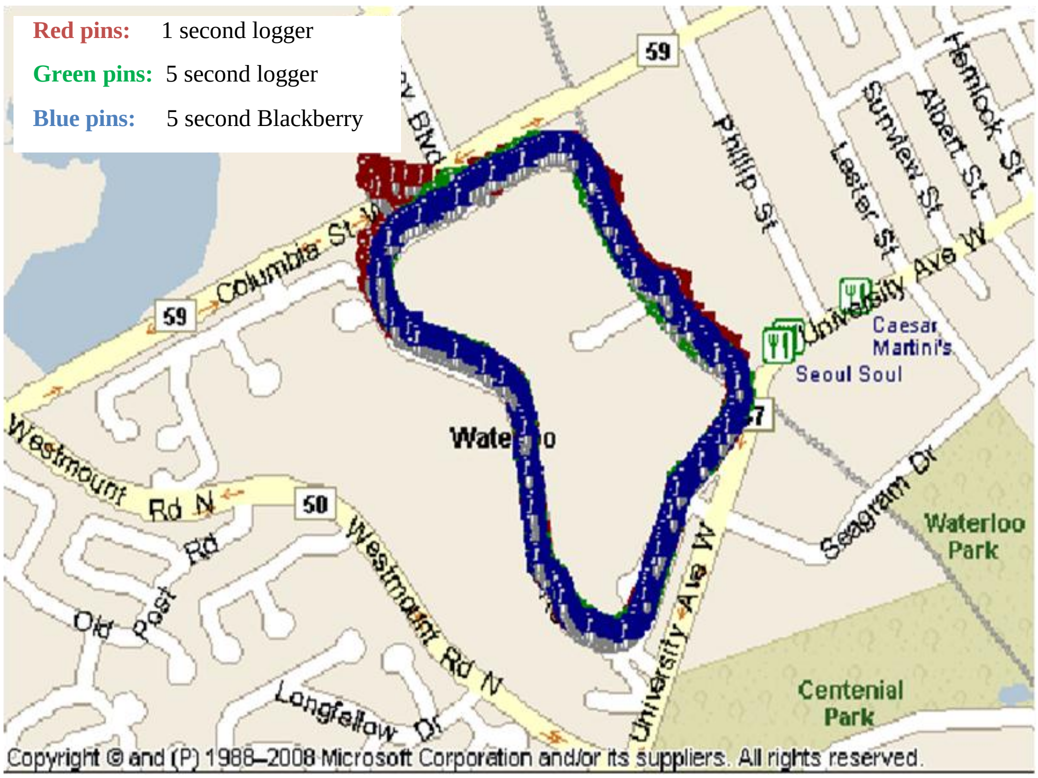
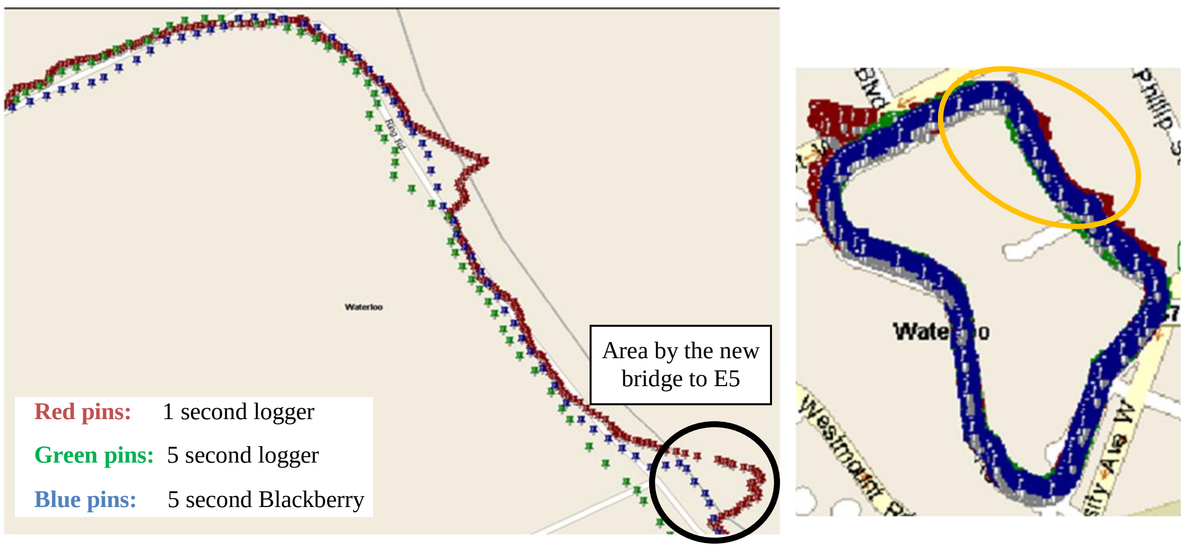


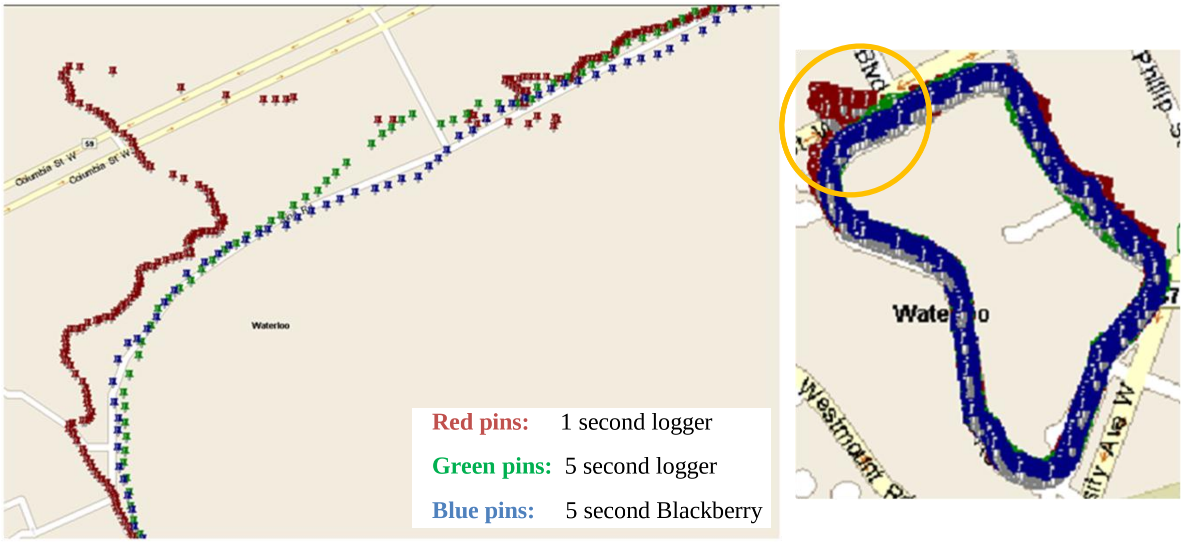
Figure 24: GPS positions collected from Blackberry smart-phone and GPS loggers, in a trial walk around UW Ring Rd. (Source: MapPoint)


Red pins: 1 second logger
Green pins: 5 second logger
Blue pins: 5 second Blackberry
Waterloo

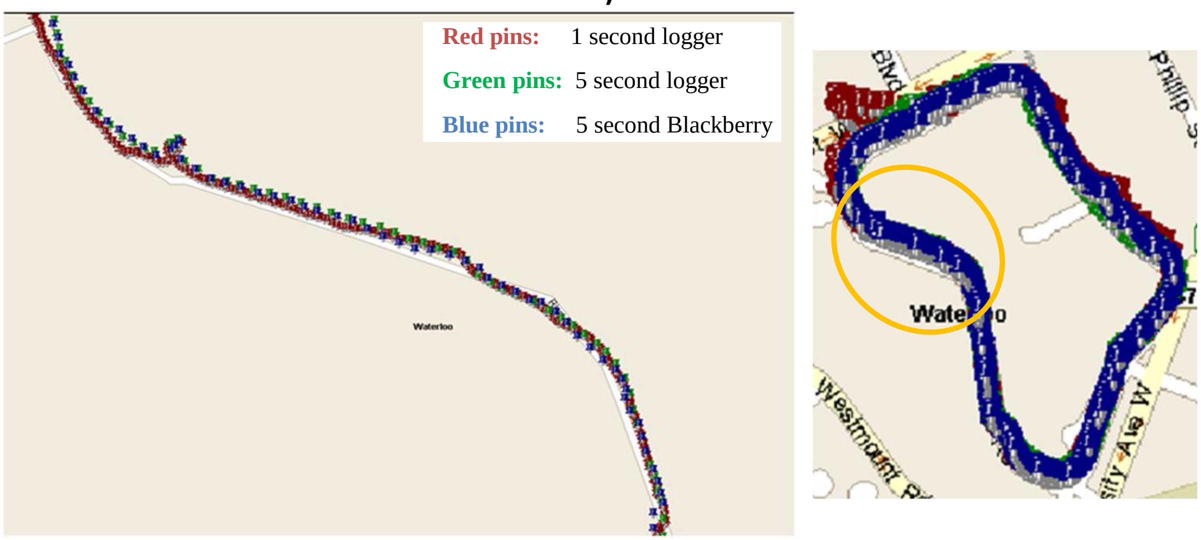
a)


Area by the new bridge to E5
Red pins: 1 second logger
Green pins: 5 second logger
Blue pins: 5 second Blackberry
Waterloo

b)


Red pins: 1 second logger
Green pins: 5 second logger
Blue pins: 5 second Blackberry
Columbia St W
Waterloo
Westmount
Phillip

c)


Red pins: 1 second logger
Green pins: 5 second logger
Blue pins: 5 second Blackberry
Waterloo
Westmount
Phillip

d)

e)

Figure 25: Zoomed-in maps comparing the GPS positions collected from Blackberry smart-phone and GPS loggers, in a trial walk around UW Ring Rd. (Source: MapPoint)

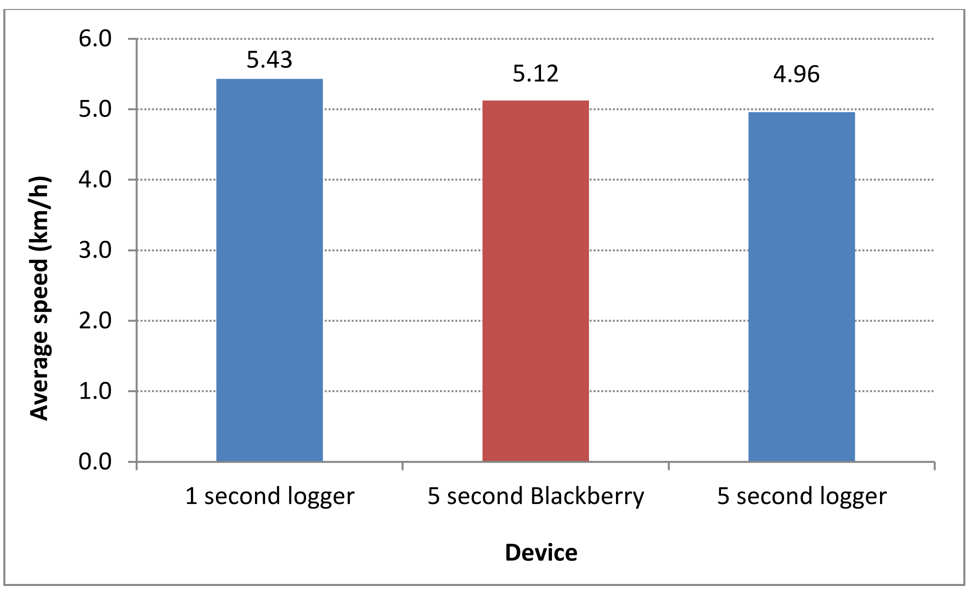


Figure 26: Comparison of average walking speed between Blackberry smart-phone and GPS loggers

Table 3: 95% confidence limits on the mean speeds of Blackberry smart-phone and GPS loggers

Speed (km/h)			
	1 second logger	5 second Blackberry	5 second logger
95% lower confidence limit	5.35	4.99	4.81
Sample Mean	5.43	5.12	4.96
95% upper confidence limit	5.51	5.24	5.11

Table 4: Comparison of walking speed recorded by 1 second logger and 5 second Blackberry

Device	Average Speed (km/h)	Standard deviation of Speed	No. of points
1 second logger	5.43	1.62	1587
5 second Blackberry	5.12	1.10	315
$\nu = 624.32$, $\|(t_o = 4.28)\| > (t_{0.025,\nu} = 1.96)$ -> Reject H_0			

Table 5: Comparison of walking speed recorded by 5 second logger and 5 second Blackberry

Device	Average Speed (km/h)	Standard deviation of Speed	No. of points
5 second logger	4.96	1.38	323
5 second Blackberry	5.12	1.10	315
$\nu = 611.61$, $\|(t_o = 1.60)\| < (t_{0.025,\nu} = 1.96)$ -> Fail to reject H_0			

Stationary Mode

For the stationary mode, a Blackberry Bold 9700 and a GPS logger were simultaneously used to capture the GPS data. For both devices the capture interval (I_c) was set to 5 seconds. Two locations were considered for the data collection: Location 1, which was on top of a hill with no obstructions where devices could get full GPS coverage, and Location 2 which was close to buildings that hindered satellites view (Figure 27). The experiment was conducted for 7 minutes for each location. However, in order to reduce the error caused by start-up, only 5 minutes of data (60 points) were analyzed. Autonomous GPS mode was used for the data collection.

The collected GPS positions were converted to UTM[1] coordinates (Northing and Easting) and the mean value of the UTM coordinates was calculated (i.e. mean position). Then, the

[1] Universal Transverse Mercator

distance (in meters) of each position to the mean position was calculated (i.e. individual position error). Finally, average position error was calculated.

Table 6 shows the error computed for each device and at each location. According to this table, at Location 1 where there is a clear view of satellites, the average error computed for the logger is 0.55 meter and the average error computed for the Blackberry device is 0.95 meter. In contrast to walking trip experiment, it seems that the logger performed better in comparison to the Blackberry smart-phone. At Location 2 where nearby buildings may occlude the satellite signal, both the Blackberry device and the logger exhibited a much larger mean position errors (5.80m, 6.26m respectively), but the Blackberry provided better position accuracy than the GPS logger. Figure 28 and Figure 29 also illustrate the variation in the GPS positions collected by each device at location 1 and location 2, respectively. In each figure the center position (0,0) represents the mean position of logger and Blackberry points, and the plotted points show the distance of logger and Blackberry positions from this mean position. Figure 28 shows that the Blackberry points and Logger points do not overlap and it appears that the mean positions of logger and Blackberry are different. According to the results presented in Figure 28 and Figure 29, the variation of the data at Location 2, where there are a limited number of satellites in view, is higher for each device in comparison to Location 1, where there is a clear view of satellites.

Figure 27: Location 1 and Location 2 on the map, where stationary GPS data collected by GPS logger and Blackberry (Source: Google map)

Table 6: Comparison of position accuracy between Blackberry smart-phone and GPS logger

Location	Device	Average Error (meter)
Location 1	Logger	0.55
Location 1	Blackberry	0.95
Location 2	Logger	6.26
Location 2	Blackberry	5.80

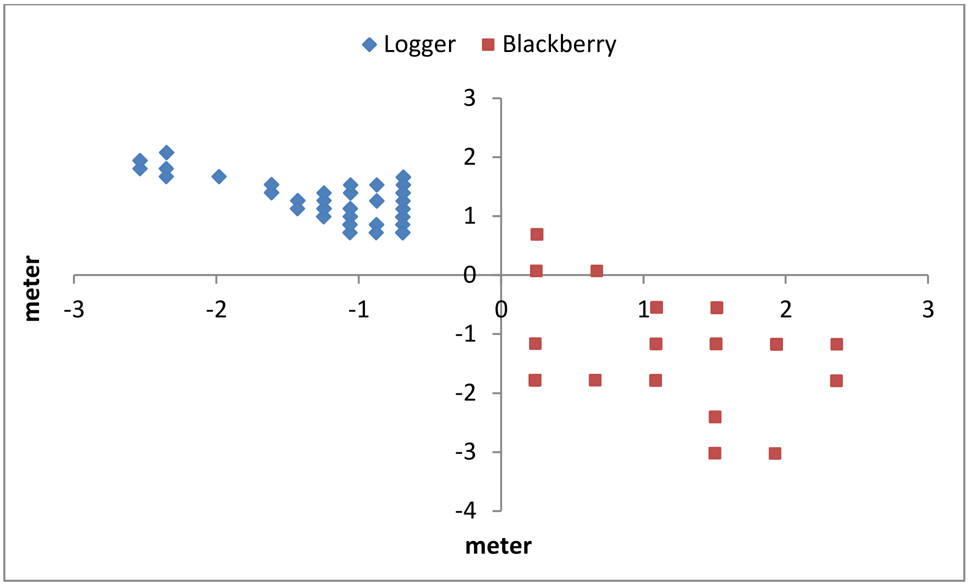


Figure 28: Variation in the positions (distance from mean position) collected by GPS logger and Blackberry at stationary mode in location 1 (clear satellite view)

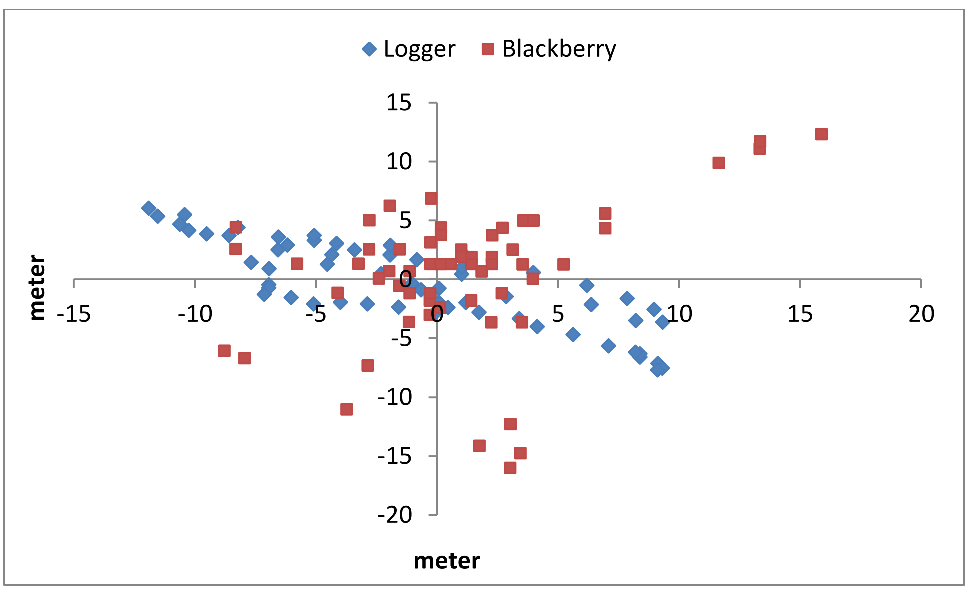


Figure 29: Variation in the positions (distance from mean position) collected by GPS logger and Blackberry at stationary mode in location 2 (weak satellite signals)

4.1.2 Accuracy of Blackberry smart-phone at different GPS modes

The purpose of this experiment is to compare the accuracy of the Blackberry smart phone for different GPS modes (i.e. Autonomous, Assisted and Cell site modes) to GPS logger in a walking trip.

For this experiment, one Blackberry Bold 9700 and one GPS logger have been simultaneously employed to capture the GPS data. Both devices were configured to capture the data at capture interval (I_c) of 5 seconds, and a separate walking test was conducted for each GPS mode for a duration of 5 minutes.

Figure 30, Figure 31, and Figure 32 respectively show the data captured at Autonomous mode, Assisted mode, and Cell site mode. The results from Autonomous and Assisted modes show that at both modes Blackberry device performed well, following a consistent path with no outliers. Moreover, the position data collected via both modes closely followed the data collected via the GPS logger data. The data collected via the Assisted mode appear to be more accurate than the data collected via the GPS logger; however, this can not be quantitatively demonstrated because the true positions are not known.

For Cell Site mode, the reported positions by Blackberry were bouncing back and forth between two points (Figure 32). The reason for this is that at cell site mode the positions are determined based on cell towers and as a result the reported positions accuracy is too low.

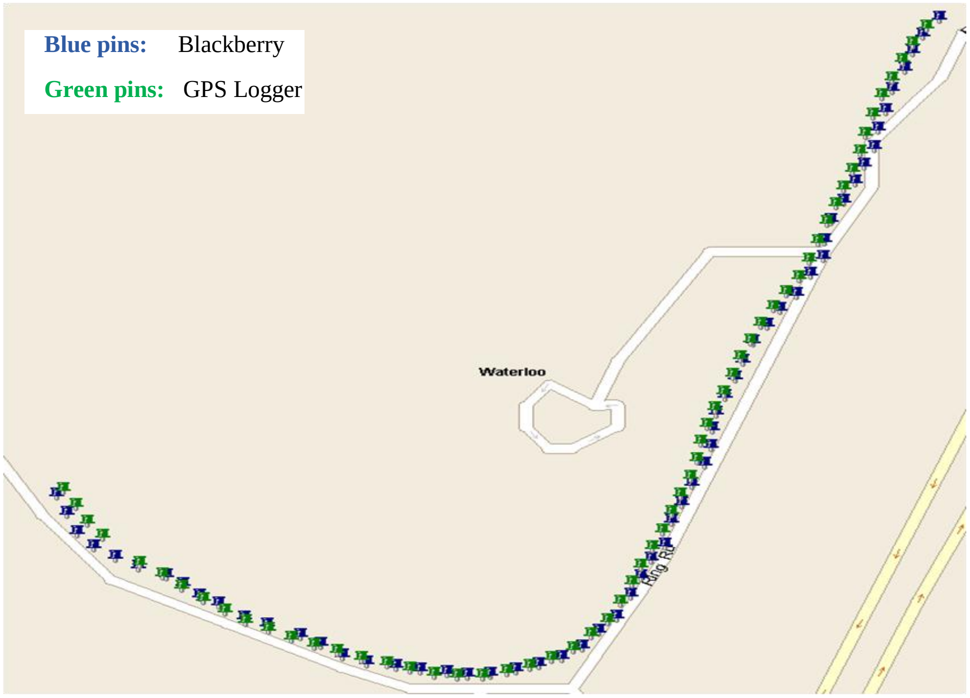


Figure 30: Comparison of GPS positions between Blackberry Autonomous mode and GPS logger, in a walking trip (Source: MapPoint)

Figure 31: Comparison of GPS positions between Blackberry Assisted mode and GPS logger, in a walking trip (Source: MapPoint)

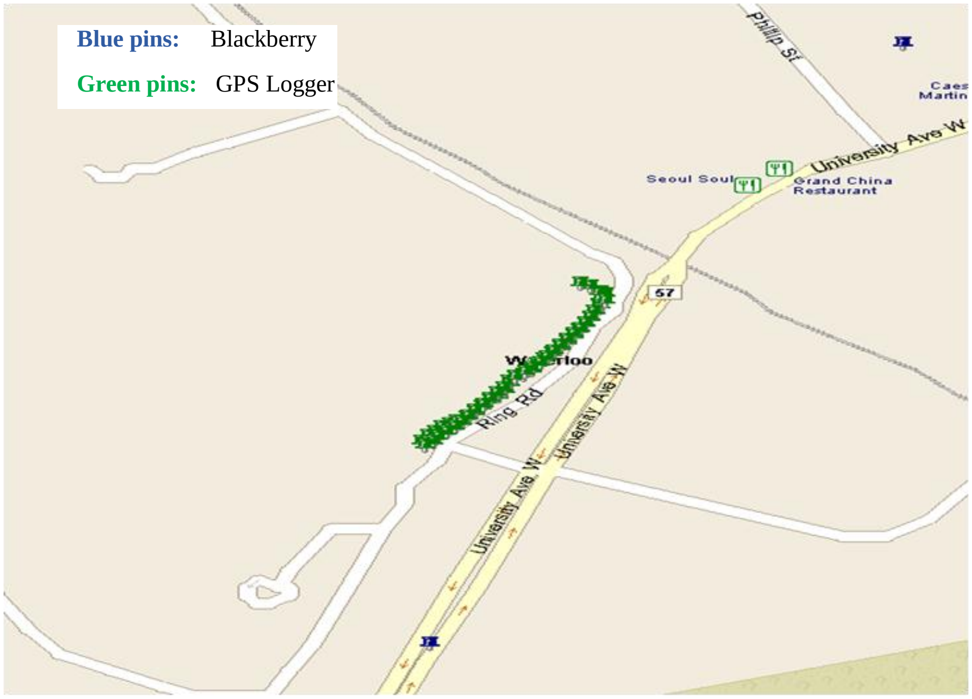


Figure 32: Comparison of GPS position accuracy between Blackberry cell site mode and GPS logger, in a walking trip (Source: MapPoint)

4.1.3 Comparison of battery consumption by different GPS modes

The purpose of this experiment is to compare the battery consumption of Autonomous, Assisted and cell site modes. For this purpose a Blackberry Bold 9700 is utilized, and the battery consumption was examined for 30 minutes for each GPS mode. To measure the battery consumption, the "Battery Level" data recorded by the application has been used. All the tests have been conducted at stationary mode and with capture interval (I_c) of 5 seconds.

The results from this experiment are illustrated in Figure 33. This figure shows the time taken to consume 1% of battery at each mode. According to the result, Autonomous mode has the highest rate of battery consumption at which 1% of battery is on average consumed in 3.72 minutes. Since capture interval (I_c) is a fix value equal to 5 seconds, the battery consumption rate is expected to be linear over time, which is also confirmed through an experiment in section4.3.2. Therefore, for Autonomous mode the battery is drained completely after 6.2 hours. On the other hand, Cell Site mode has the lowest rate of battery consumption with 1% usage on average at every 6 minutes. As a result, the battery lasts for 10 hours when Cell Site mode is used. The battery consumption by Assisted mode falls between the ones from the other two modes. In this case, 1% of battery is consumed on average at each 4.80 minutes and the battery is drained completely within 8 hours.

The results from this experiment confirm the comparison that has been made in Literature Review in terms of the battery consumption rate by each GPS mode. Although the Cell Site mode has the least battery consumption rate, it is not a proper mode for frequent data collection because, as shown in the previous section, the accuracy of positions is too low for collecting travel behavior data. On the other hand, Assisted mode, being in the second place in terms of battery consumption, seems to be a reasonable choice for frequent data collection.

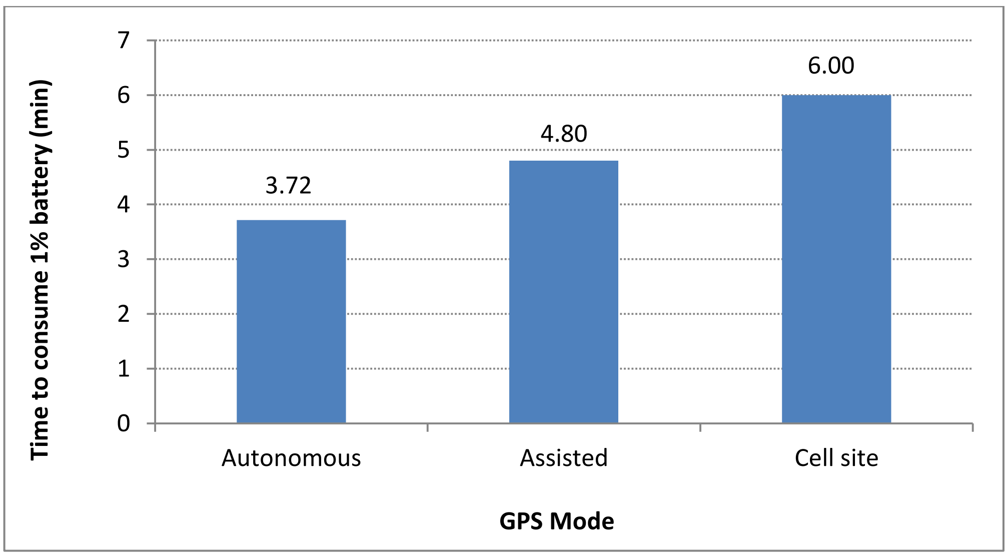


Figure 33: Comparison of battery consumption rate by different GPS modes

4.1.4 Comparison of accuracy between Blackberry smart-phones

In the current research, Blackberry Bold 9700 and Torch 9810 are employed to collect and analyze the travel data. Therefore, some experiments have been conducted to compare the position, speed and "accuracy" reported by these two devices, which are discussed in this section.

Visual comparison of positions

The purpose of this experiment is to visually compare the GPS positions recorded by Blackberry Bold 9700 and Torch 9810. For this purpose, an auto and a bike trip were conducted. The test duration for auto trip was 40 minutes and for bike trip was 20 minutes. For both tests, the capture interval (I_c) of 5 seconds and the GPS mode of Assisted was used to collect the travel data. The data was collected using a Blackberry Bold 9700 and a Blackberry Torch 9810 simultaneously for each test.

The GPS positions recorded in the auto trip are plotted in Figure 34, and the positions recorded in the bike trip are plotted in Figure 36. Figure 35 and Figure 37 also respectively illustrate the zoomed-in map for the auto trip and the bike trip. These figures show that the GPS positions recorded by the two devices closely follow each other. In other words, devices show similar results in capturing the data for each of the tested travel modes, and the position accuracy seems to be approximately the same.

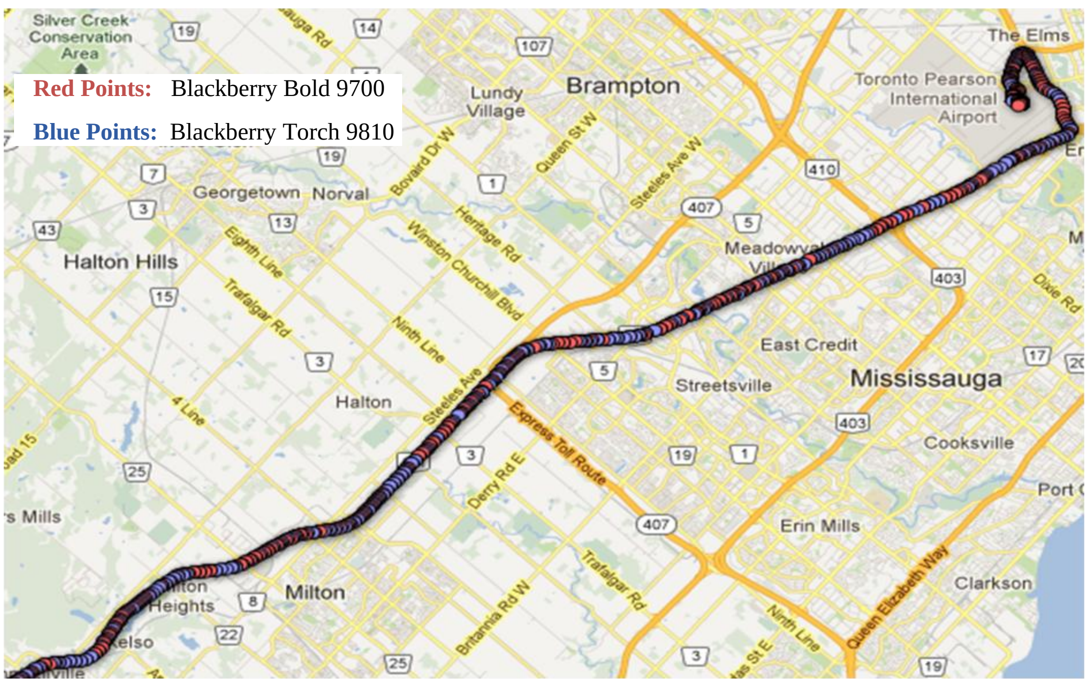


Figure 34: Auto trip from Toronto Pearson Airport to Waterloo, captured by Blackberry Bold 9700 and Torch 9810 (Source: Google map)

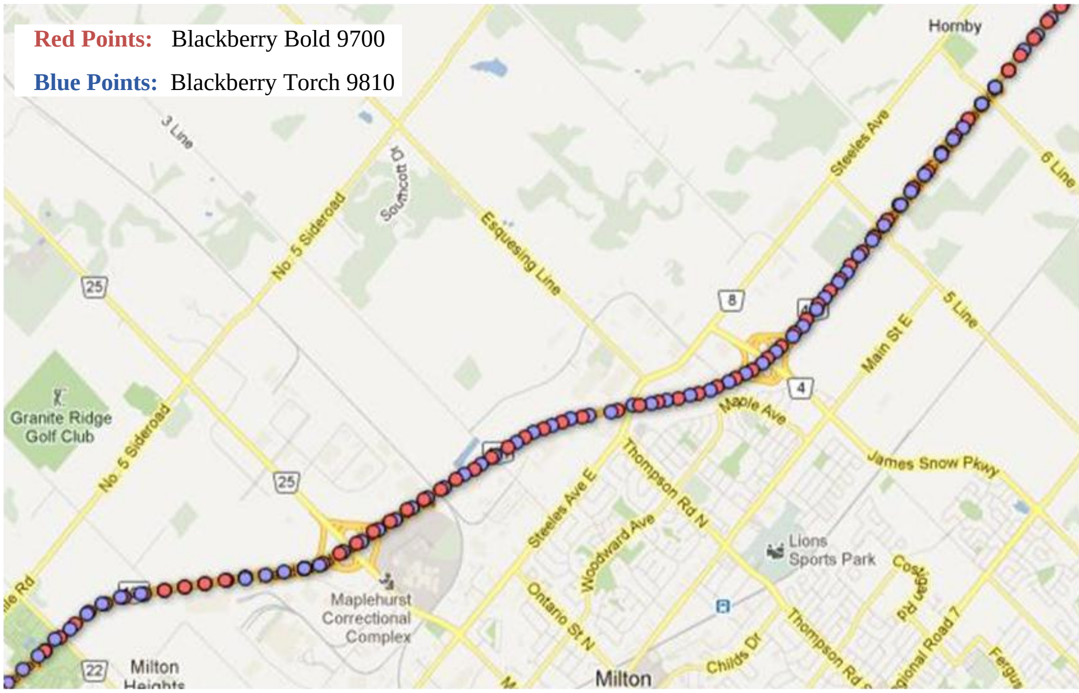


Figure 35: Zoomed-in map of auto trip from Toronto Pearson Airport to Waterloo, captured by Blackberry Bold 9700 and Torch 9810 (Source: Google map)

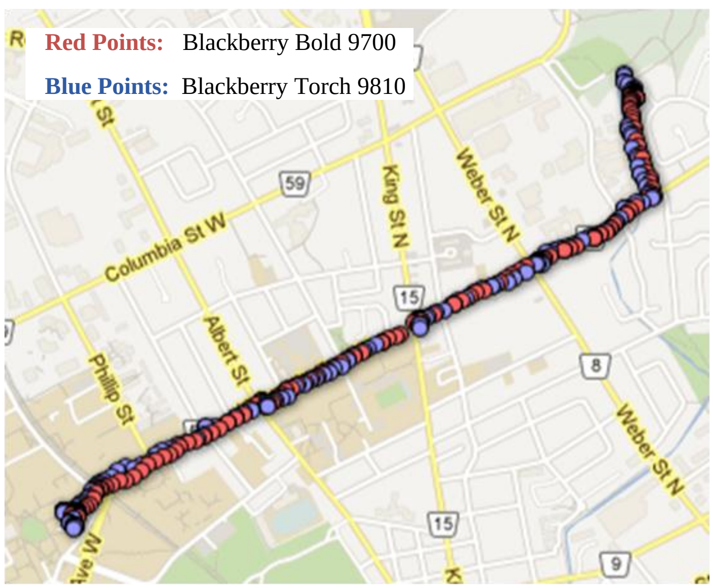


Figure 36: Bike trip from Hillside Trail to University of Waterloo, captured by Blackberry Bold 9700 and Torch 9810 (Source: Google map)

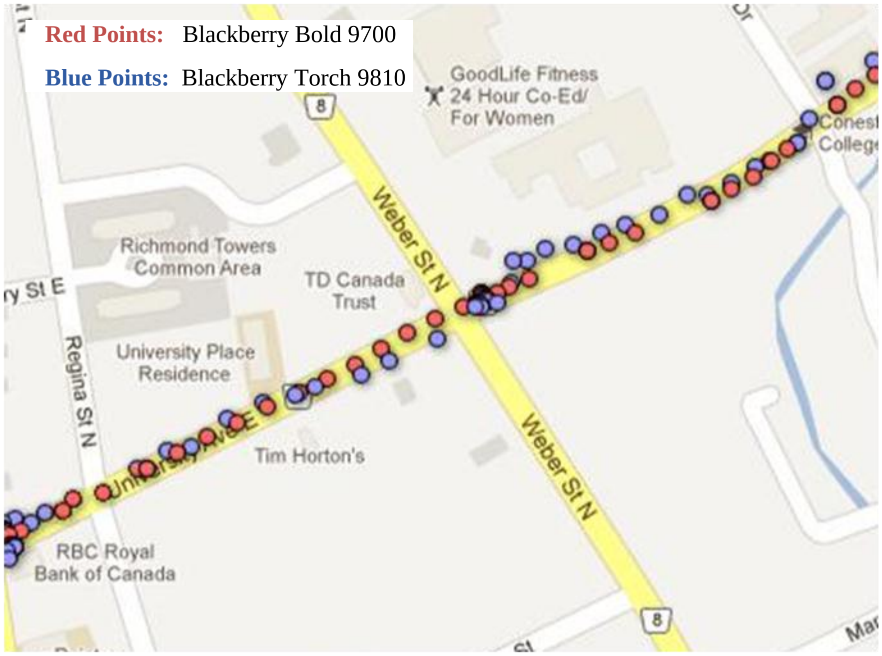


Figure 37: Zoomed-in map of bike trip from Hillside Trail to University of Waterloo, captured by Blackberry Bold 9700 and Torch 9810 (Source: Google map)

Comparison of the reported speed

In this experiment the speed of the GPS positions captured by Blackberry Bold 9700 and Blackberry Torch 9810 are compared. For this purpose, an auto trip, a bike trip, and a walking trip were conducted, each for the duration of about 20 minutes. For each test, the two devices were used simultaneously to capture the travel data. The devices were configured to capture the data at capture interval (I_c) of 5 seconds and using Assisted GPS mode.

Figure 38, Figure 39, and Figure 40 respectively compare the speed reported by the two devices for the auto trip, bike trip, and walking trip. The results from the auto trip show that the speeds reported by the two devices are following the same trend over time, and it does not reveal any large differences between the two devices in reporting the position speed. The results from the bike trip also show that the speeds reported by the two devices are very similar. However, in the first and last part of the bike trip where the speed is too low, the difference between the speeds of the two devices is more apparent than the middle part where the speed is high. For the walking trip, the speeds reported by the two devices seem to be different and the Torch device reveals a high fluctuation in the reported speed in comparison to the Bold device.

Moreover, the data captured from the two devices for all the modes show that the precision of the speed values reported by the Torch device is restricted to 0.25 m/s increments. As an example, for the walking mode the reported speeds are 0, 0.25, 0.5, 0.75, or next. This seems to be a restriction of the Torch device.

According to the results, this restriction of the Torch device in reporting the accurate speed and also the high fluctuation of the speed of the Torch device becomes apparent when the speed is low. Based on Figure 38 and Figure 39 these limitations of the Torch device in general seem not to have a substantial negative impact on trips with higher speeds such as auto and bike trips. In order to identify if the mean speeds reported by the two devices are statistically different or not, a mean comparison test also has been conducted for each mode (details are provided in Appendix D):

H_0: $\mu_1 = \mu_2$ H_1: $\mu_1 \neq \mu_2$

μ_1: Mean speed by Torch 9810

μ_2: Mean speed by Bold 9700

The results of the hypothesis tests for auto, bike and walk trips are summarized in Table 7, Table 8, and Table 9, respectively. According to the results, we fail to reject the null hypothesis for auto and bike trips. Therefore, with 95% confidence we accept that the mean speeds of the two devices are not statistically different for auto and bike trips. On the other hand, the results show that the null hypothesis is rejected for the walk trip. Therefore, we conclude that the mean speeds of the two devices are statistically different for walk trips.

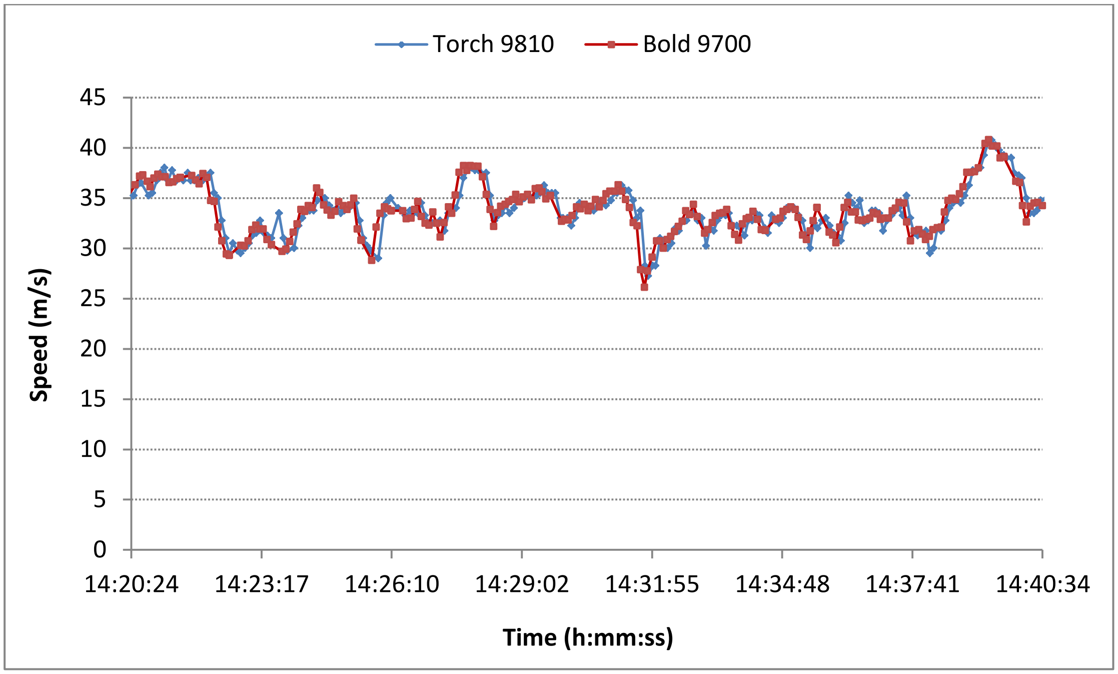


Figure 38: Comparison of speed reported by Bold 9700 and Torch 9810 in an auto trip

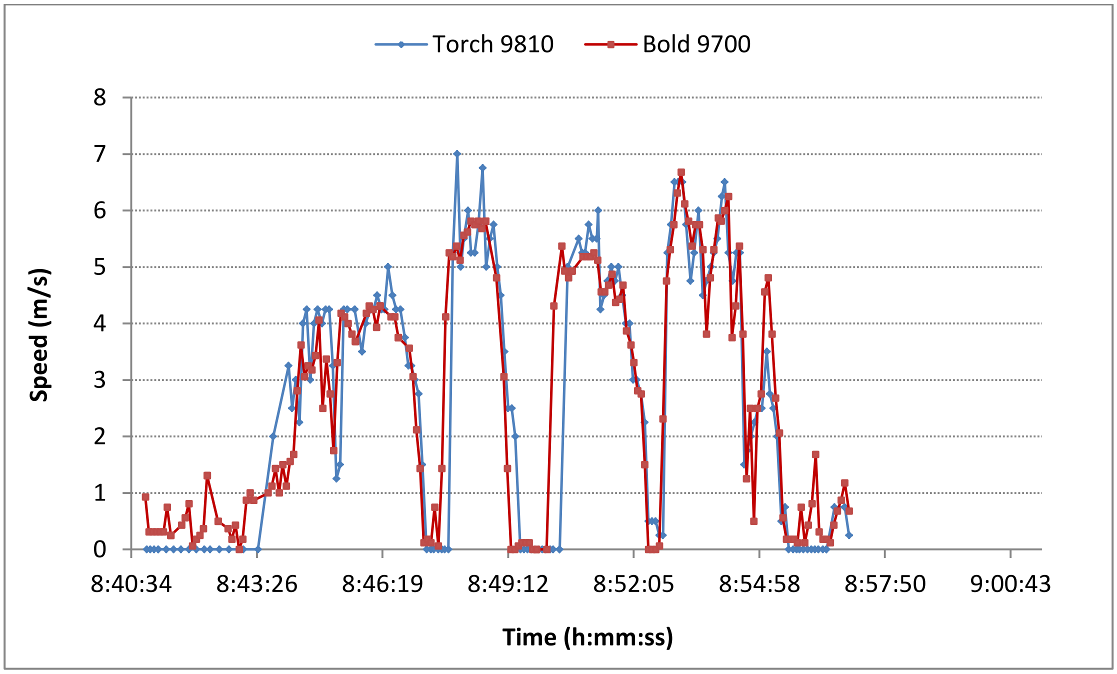


Figure 39: Comparison of speed reported by Bold 9700 and Torch 9810 in a bike trip

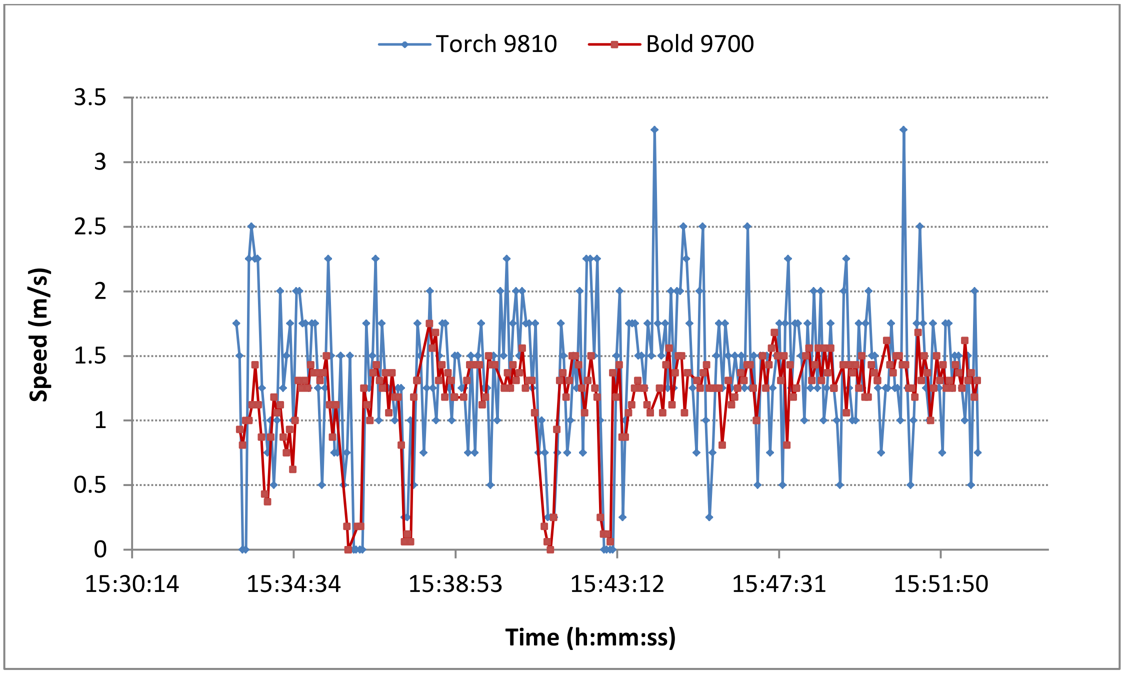


Figure 40: Comparison of speed reported by Bold 9700 and Torch 9810 in a walking trip

Table 7: Comparison of auto speed recorded by Blackberry Torch-9810 and Blackberry Bold 9700

Device	Average Speed (m/s)	Standard deviation of Speed	No. of points
Torch 9810	32.33	4.48	296
Bold 9700	32.45	4.34	297
$\nu = 590$, $\lvert(t_o = -0.31)\rvert < (t_{0.025,\nu} = 1.96)$ -> Fail to reject H_0			

Table 8: Comparison of bike speed recorded by Blackberry Torch-9810 and Blackberry Bold 9700

Device	Average Speed (m/s)	Standard deviation of Speed	No. of points
Torch 9810	2.81	2.27	167
Bold 9700	2.66	2.11	191
$\nu = 342$, $\lvert(t_o = 0.65)\rvert < (t_{0.025,\nu} = 1.96)$ -> Fail to reject H_0			

Table 9: Comparison of walking speed recorded by Blackberry Torch-9810 and Blackberry Bold 9700

Device	Average Speed (m/s)	Standard deviation of Speed	No. of points
Torch 9810	1.35	0.6	234
Bold 9700	1.18	0.36	234
$\nu = 385$, $\lvert(t_o = 3.69)\rvert > (t_{0.025,\nu} = 1.96)$ -> Reject H_0			

These results suggest that the accuracy of the Torch device in reporting lower speeds is not as good as the Bold device. The significant fluctuations in the reported speeds from the Torch device at lower speeds suggests that speed accuracy might be improved through the use of a moving average. Figure 41 illustrates the Mean Absolute Error Proportional (MAEP) versus the interval size for moving average speed for the walk trip. The MAEP is calculated based on the following equations:

$$MAEP = \frac{\dfrac{\sum_{i=1}^{n_m} |e_i|}{n_m}}{\dfrac{\sum_{i=1}^{k} |S_{i,B}|}{k}} * 100 \qquad \textbf{Equation 9}$$

$$|e_i| = \left| S_{i,B} - \overline{S}_{m,T} \right| \qquad \textbf{Equation 10}$$

where,

$S_{i,B}$ = Speed reported by Bold at location point i

$\overline{S}_{m,T}$ = Moving average on speed reported by Torch when interval size is m

e_i = Absolute error between reported speed by Bold and smoothed speed by Torch

n_m = Number of calculated e_i when the interval size is m

k = Number of location points reported by Bold

According to the results from Figure 41, by smoothing the Torch speed, the error approximately reduces to 15% when large interval size of 100 is used. Interval sizes of 3, 4, 5, and 10 reduce the error to 29.7%, 28.5%, 27.8%, and 26.7%, respectively. Interval size 10 does not seem to be a reasonable choice for smoothing the speed since it does not have a large contribution in decreasing the error comparing to the smaller interval sizes. Since the location data are obtained every 5 seconds, by using the interval size 3 to 5 the speed is smoothed over 15 to 25 seconds, respectively. In the current experiment, the average speed recorded by Bold for the walk trip is 1.18 m/s. Therefore by considering the interval size of 3 (MAEP = 29.7%), the average speed by Torch can range from 0.83 m/s to 1.53 m/s. By considering the interval size of 5 (MAEP = 27.8%), the average speed by Torch can range from 0.85 m/s to 1.5 m/s. The results from interval size 3 and 5 are very similar. According to these results interval size 3 appears to be a reasonable option for smoothing the speed since the computed average speed range (0.83 m/s to 1.53 m/s) still can be considered acceptable for walking mode.

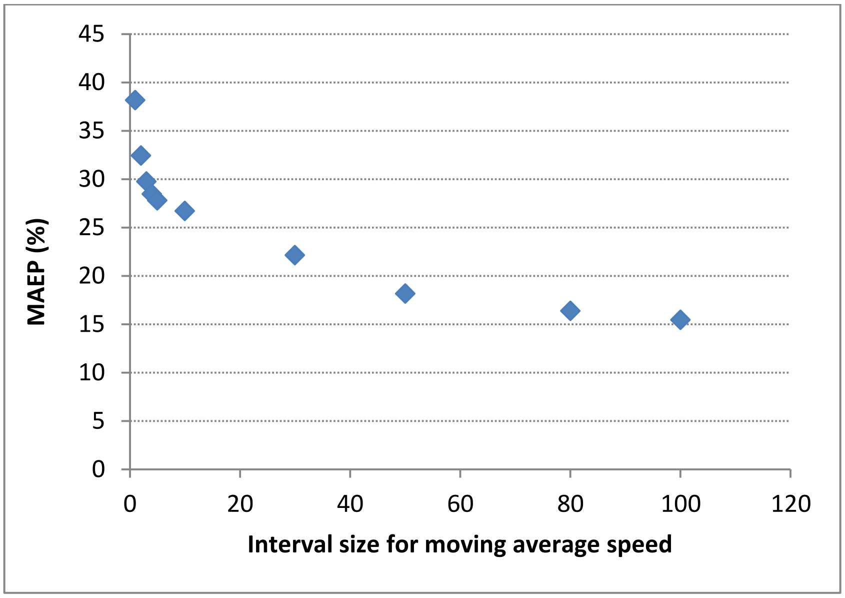


Figure 41: MAEP between Torch smoothed speed and Bold speed as a function of interval size for moving average speed – The walk trip

Comparison of the reported "accuracy"

As explained in the Methodology chapter, the Blackberry reports a measure of GPS "accuracy" when a GPS position is captured. We place the term "accuracy" in quotations because the term "accuracy" is provided in the Blackberry documentation, but the way in which this value is computed is not known. In this experiment, the "accuracy" data recorded by the Blackberry Bold 9700 and Blackberry Torch 9810 are compared. For this purpose, an auto trip has been conducted for the duration of 1 hour. The two devices were used simultaneously to capture the travel data at capture interval (I_c) of 5 seconds and using Assisted GPS mode.

Figure 42 illustrates the "accuracy" reported by Bold 9700 versus the "accuracy" reported by Torch 9810. It is expected that for a given location, both devices would report the same (or at least similar) values for "accuracy". However, the results show that the "accuracy" values reported by Torch 9810 are typically much larger compared to the values reported by Bold 9700. The values from the Bold device range from approximately 2m to 8m with an average of 3.3m. The values from the Torch device range from approximately 7m to 107m with an average of 33.7m. The values suggest that the position errors in the GPS data obtained from the Torch device are approximately 10 times those of the Bold device. However, a visual examination of the reported positions when plotted on a satellite image suggests that the position accuracies are approximately equal.

Figure 42 also indicates that there is no correlation between the "accuracy" reported by the two devices and therefore a correction factor cannot be used to adjust the "accuracy" reported by Torch.

As a conclusion, the "accuracy" reported by Torch is not reliable and is not recommended to be used in analyzing the accuracy of positions.

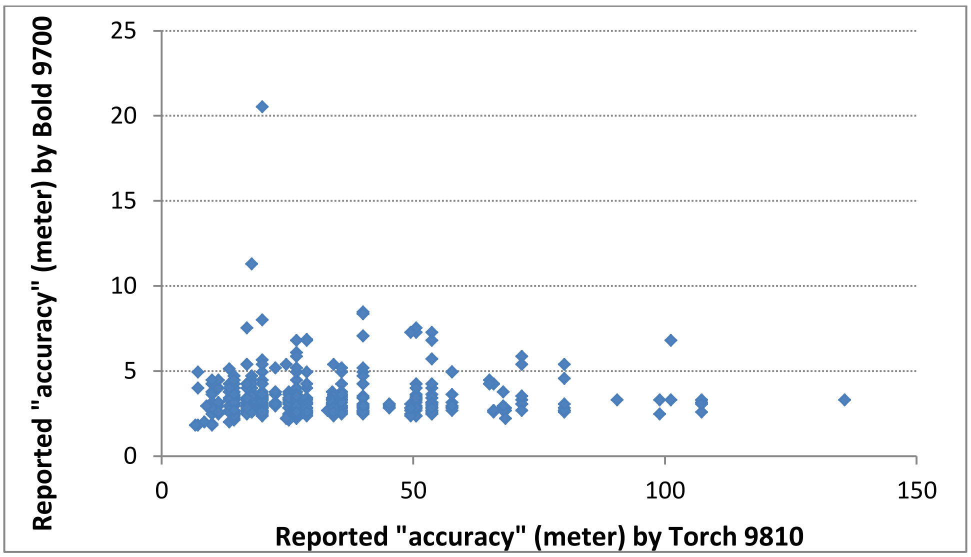


Figure 42: Correlation of "accuracy" reported simultaneously by Bold 9700 and Torch 9810

4.2 Evaluation of Optimization Algorithms

In this section the performance of the State Machine (SM) algorithm and Critical Point (CP) algorithm discussed in the Methodology chapter is analyzed.

4.2.1 State Machine (SM) Algorithm

The performance of this algorithm can be examined in two stages: the initial version and the revised version which are described as follows.

4.2.1.1 Initial State Machine Algorithm

As explained in the SM algorithm section of the Methodology chapter, for the initial version of this algorithm no error handling was considered. Moreover, G_{th} (explained in section 3.3.2.3) was assumed to be zero and stationary speed threshold ($S_{th,s}$) was considered as 0.5 m/s. This section investigates the impact of these initial settings on the performance of the SM algorithm. For this purpose, two Blackberry Torch 9810 devices were utilized to capture the GPS positions at a stationary location with clear view of satellites. The stationary mode was considered for this test because at this mode SM algorithm reaches to its last state where the majority of battery saving is gained. Therefore, using this test the performance of the SM algorithm can be investigated at the optimum case, when SM algorithm reaches to its last state. The reason for utilizing two of the same Blackberry devices is to analyze the impact of SM algorithm on position acquisition and battery consumption rate. Therefore, one of the devices has been configured to capture the data while SM algorithm is on (i.e. non-fixed capture interval) and the other device was configured to capture the data while SM algorithm is off (i.e. fixed capture interval). For the first case, mminimum capture interval ($I_{c,min}$) was set to 5 seconds and maximum capture interval ($I_{c,max}$) was set to 5 minutes. For the second case, the capture interval (I_c) was set to 5 seconds. For this experiment, a transmission interval (I_t) of 1 minute was used, the accelerometer was turned on, and the Assisted GPS mode was selected. The tests started at 100% battery level for each device.

The results of this experiment are illustrated in Figure 43 to Figure 47. In Figure 43 the intervals at which positions are captured are plotted as a function of elapsed time. This figure shows that for the device with fixed intervals the points are mostly captured in less than 10 seconds. The reason for exceeding 5 seconds for some of the points is that a new GPS point has not been available at the specified 5 seconds capture interval and therefore the device has attempted to retrieve a new point in the next few seconds until reaching to time-out (C_{TO}) threshold (see Time-Out in section 3.3.2.1). If a new GPS point is not available in time-out period, then it will be attempted at the next capture interval. As a result, though the capture interval (I_c) is set to a fix value, the time interval between the captured GPS points might vary depending on the availability of a new GPS point at the specified capture interval (I_c) and time-out (C_{TO}) period. As for the device with non-fixed intervals, different states (State 0 to State 6) of the state machine through which the GPS fixes are retrieved are illustrated. The figure shows that many times during the test the state machine has moved forward from the first state toward the last state and then has returned back to the first state without reaching to the last state. This result is surprising since the device has been at stationary mode throughout the test. In fact, we expect to see that the state machine starts from the first state and moves forward through the last state and remains in the last state for the entire period of the test without any return to the first state.

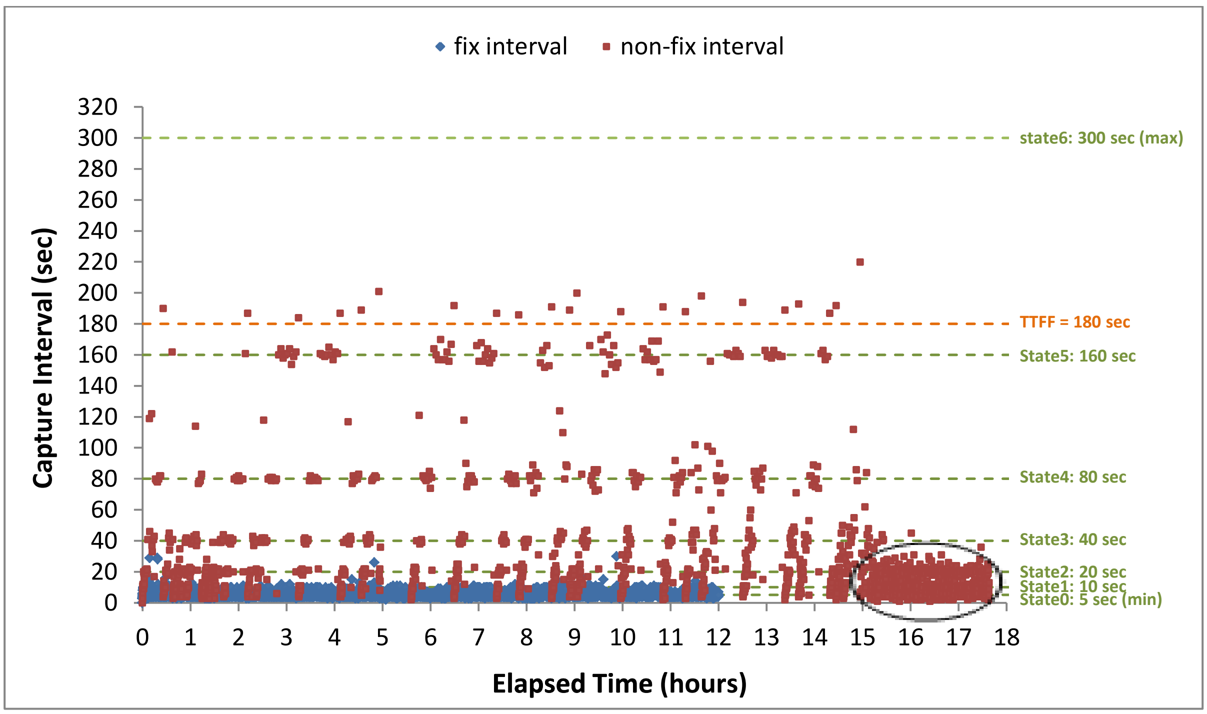


Figure 43: Capture interval as a function of elapsed time

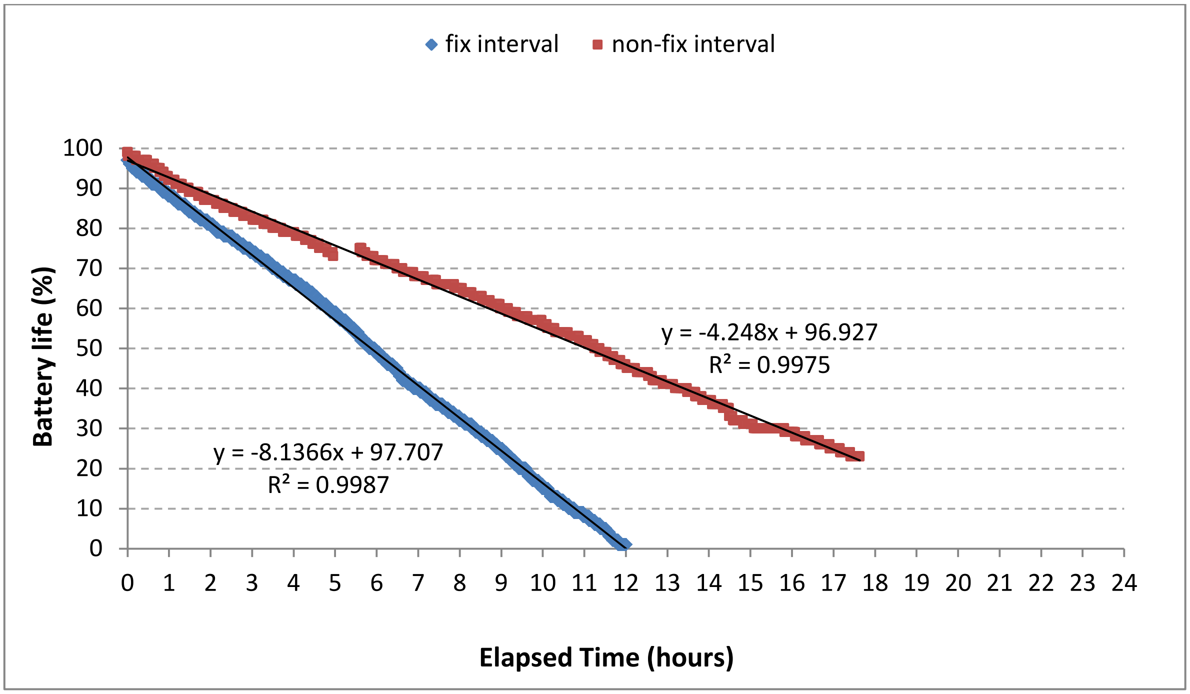


Figure 44: Battery life over time

Investigating the collected raw data shows that throughout the test different positions are reported with speed higher than $S_{th,s}$ even though the device has been in stationary mode (i.e. the reported speed contains a large error component). In these cases, the state machine has assumed that the person is moving and has returned back to the first state to collect the data more frequently. In this experiment, errors in the reported speed have occurred frequently and therefore the state machine has repeatedly returned back to the first state and started the whole process from the beginning. Moreover, the captured raw data shows that the error occurred more frequently in the last hours of the test, which can also be seen in the last part of the figure marked with circle. At this part, the state machine has only moved through the first few states and it has not reached to the next states due to frequent error occurrence.

Another point identified through the captured data was that although different times the reported speed exceeded the defined stationary speed threshold ($S_{th,s}$) of 0.5 m/s, for most of the cases it it was less than 0.75 m/s. This suggests that using 0.75 m/s as stationary speed threshold ($S_{th,s}$) improves the performance of the SM algorithm since it reduces the number of returns to the first state due to the error in the reported speed at stationary mode.

In this experiment, in addition to error occurrence, the return back to the first state has happened when state capture interval (I_n) has exceeded TTFF. As explained in Restart

Module in the Methodology chapter, initially G_{th} was considered as zero and as a result TTFF was the only parameter considered for restarting the connection. Therefore, whenever state capture interval (I_n) has exceeded TTFF in this experiment, Restart Module has restarted the connection, and data capturing has been started from the first state. For this reason, SM algorithm never has reached to state 6.

Figure 43 also shows that the device with fix interval has recorded the data for 12 hours, and the device with non-fix interval has recorded the data for 18 hours. Analysis of the captured data shows that the device with fix interval has reached to 0% of battery level after the 12 hours of data capturing. However, the device with non-fix interval has reached to 20% of battery level after the 18 hours of data capturing (Figure 44). Even though the battery was still at 20% of a full charge, no more data were received after this time because the device was configured to use a Wi-Fi connection when available to connect to wireless network for capturing Assisted GPS data. After 18 hours of operation, this connection failed.

Figure 45 illustrates the zoomed-in graph of capture interval versus elapsed time for the duration of 30 minutes. This graph shows the performance of the SM algorithm for a case when an error has occurred (i.e. reported speed > $S_{th,s}$). In this case, the state machine has reached to state 3 when the reported speed > $S_{th,s}$. Then it has returned back to the first state, and the forward movement (i.e. movement to higher states) has restarted from the beginning (marked with circle). The reported speed for this 30 min period of time is also illustrated in Figure 46. This figure shows that when the elapsed time reaches to approximately 101 minutes a speed greater than $S_{th,s}$ is reported. At this moment state machine returns back to the first state (Figure 45).

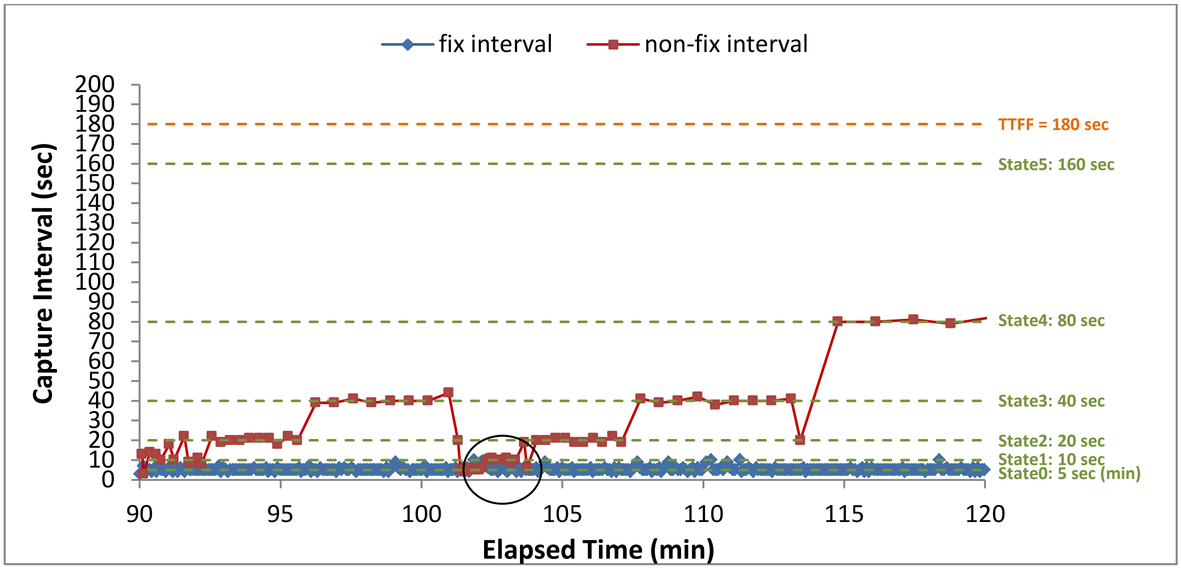


Figure 45: Capture interval as a function of elapsed time, showing error occurrence

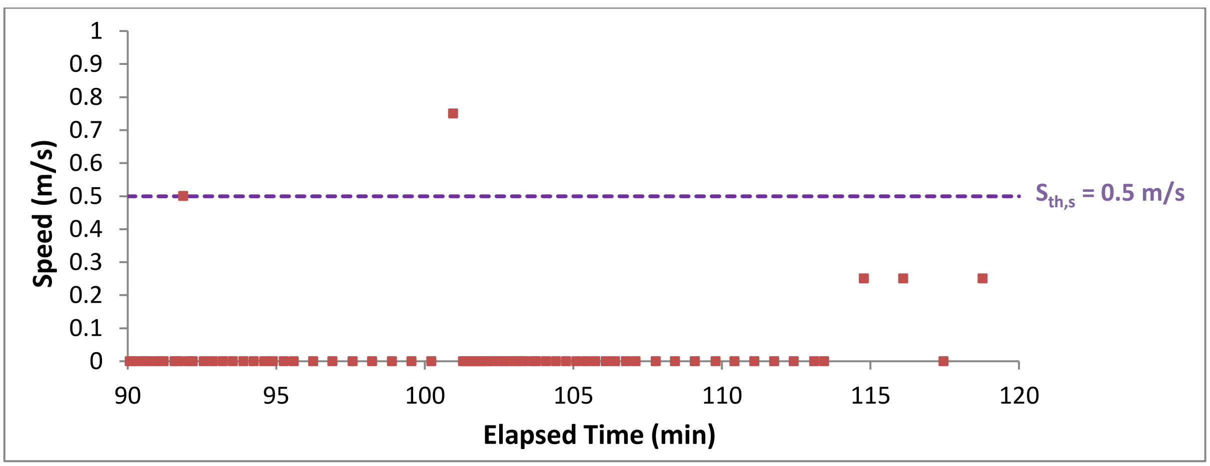


Figure 46: Reported speed versus the elapsed time (showing speed exceeding $S_{th,s}$)

Figure 47 illustrates the zoomed-in graph of capture interval versus elapsed time for the duration of 1 hour. This graph shows the performance of the SM algorithm for the cases when capture interval has exceeded TTFF. In the illustrated cases, the state machine has reached to state 5, but it could not reach state 6 because the capture interval at this state is larger than TTFF. Therefore, when the time interval reached to TTFF the state machine has been restarted (marked with circle) by the Restart Module.

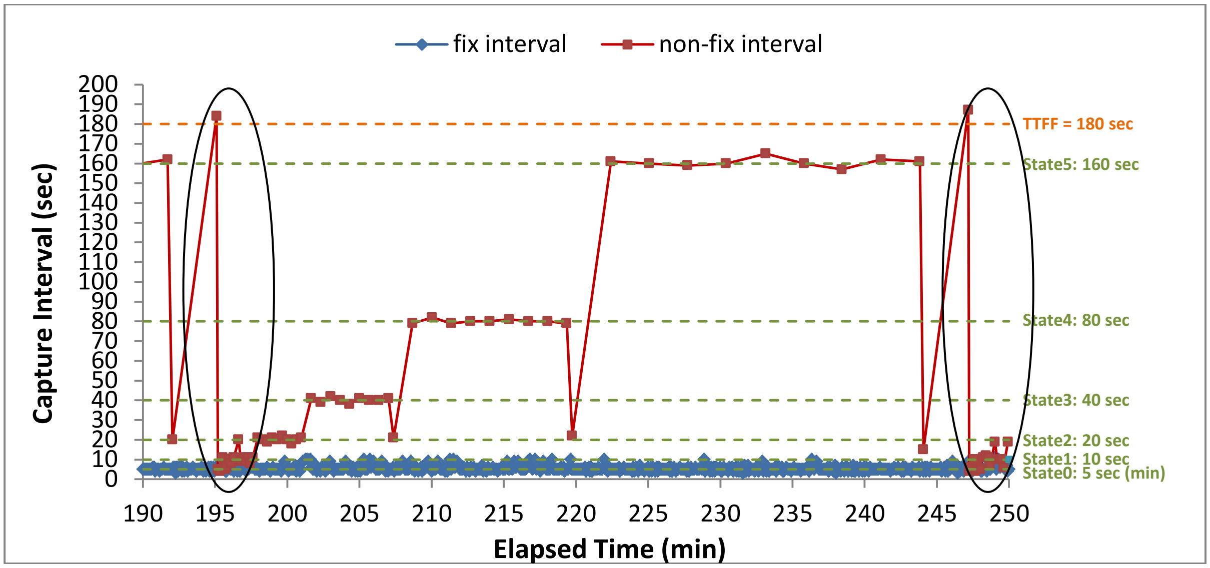


Figure 47: Capture interval as a function of elapsed time, showing restart when exceeding TTFF

As a conclusion, the results of this experiment show that SM algorithm needed to be extended to handle the occurrence of errors in reporting the stationary speed. Moreover, a gap threshold (G_{th}) should be considered in the Restart Module to allow GPS positions to be captured when state capture interval (I_n) is greater than TTFF. The stationary speed threshold ($S_{th,s}$) should also be increased to 0.75 m/s to reduce the probability of a reported speed exceeding the threshold when the device is stationary. Finally, in order to avoid the limitation of Wi-Fi connection in capturing GPS data, for the next experiments in this research the automatic Wi-Fi switch will be turned off. Through one of the experiments, the impact of the turned off Wi-Fi on battery consumption will also be analyzed.

Although the data acquisition by the device with non-fixed interval has been cut at 20% of battery level and also the SM algorithm did not perform as expected, the results of this experiment show that the battery consumption has been improved by the SM algorithm.

4.2.1.2 Revised State Machine Algorithm

This section analyzes the performance of the revised SM algorithm in which errors in the reported speed are handled as explained in Section 3.3.2.2, a gap threshold (G_{th}) is considered in Restart Module to allow GPS positions to be captured when state capture interval (I_n) exceeds TTFF, and the stationary speed threshold ($S_{th,s}$) is increased to 0.75 m/s. To examine the performance of the revised algorithm, an experiment was conducted in which one

Blackberry Torch 9810 was utilized to capture the GPS positions when the SM algorithm is on. The GPS data were captured at stationary mode where there was a clear view of satellites. The minimum capture interval ($I_{c},_{min}$) for the state machine was set to 5 seconds and the maximum capture interval ($I_{c},_{min}$) was set to 5 minutes. For this experiment, transmission interval (I_t) of 1 minute has been used, the accelerometer has been turned on, and the Assisted GPS mode has been selected. Finally, the test has been started at 100% of battery level.

Figure 48 and Figure 49 illustrate the results of this experiment. In Figure 48 the intervals at which positions are captured by the state machine are plotted versus the elapsed time. The figure shows that the state machine starts at state 0 and moves forward to the last state and stays in the last state until the battery drains. At 3 points speed is reported with values higher than $S_{th,s}$ (marked with circles). At these points, the state machine jumps to state 0 to check if a speed error has happened or not. Therefore the average speed for N_{th} number of points is calculated at this state (as explained in section 3.3.2.2). Since the average speed is less than $S_{th,s}$, it can be inferred that the reported speed (exceeding $S_{th,s}$) was an error and the state machine returns back to the previous state without having to spend any time in the middle states. Figure 49 is a zoomed-in graph which illustrates the error handling for one of the cases.

According to the results from this experiment the number of points at which reported speed is greater than $S_{th,s}$ has been reduced significantly, since $S_{th,s}$ has been increased to 0.75 m/s. For the three cases when error has occurred, the SM algorithm has identified and handled the errors. Moreover, due to considering a gap threshold (G_{th}) in Restart Module, the performance of SM algorithm has not been interrupted when reaching to the states with capture interval (I_n) greater than TTFF.

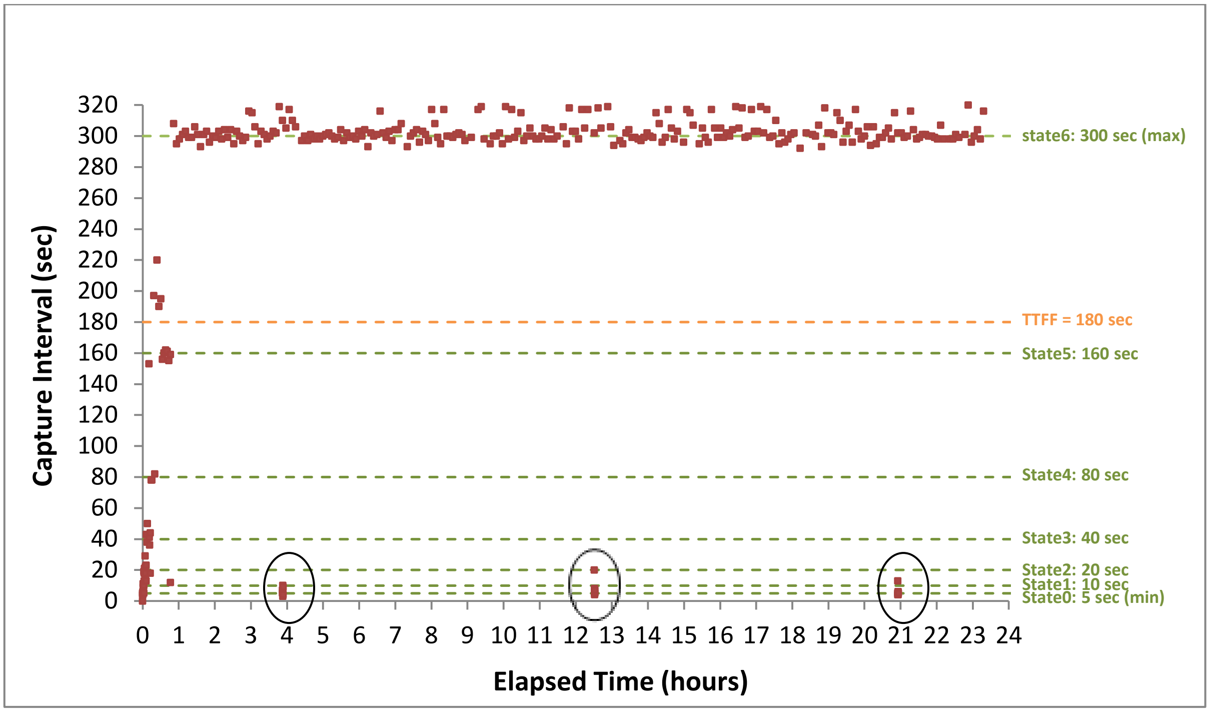


Figure 48: Capture interval as a function of elapsed time – Revised State Machine

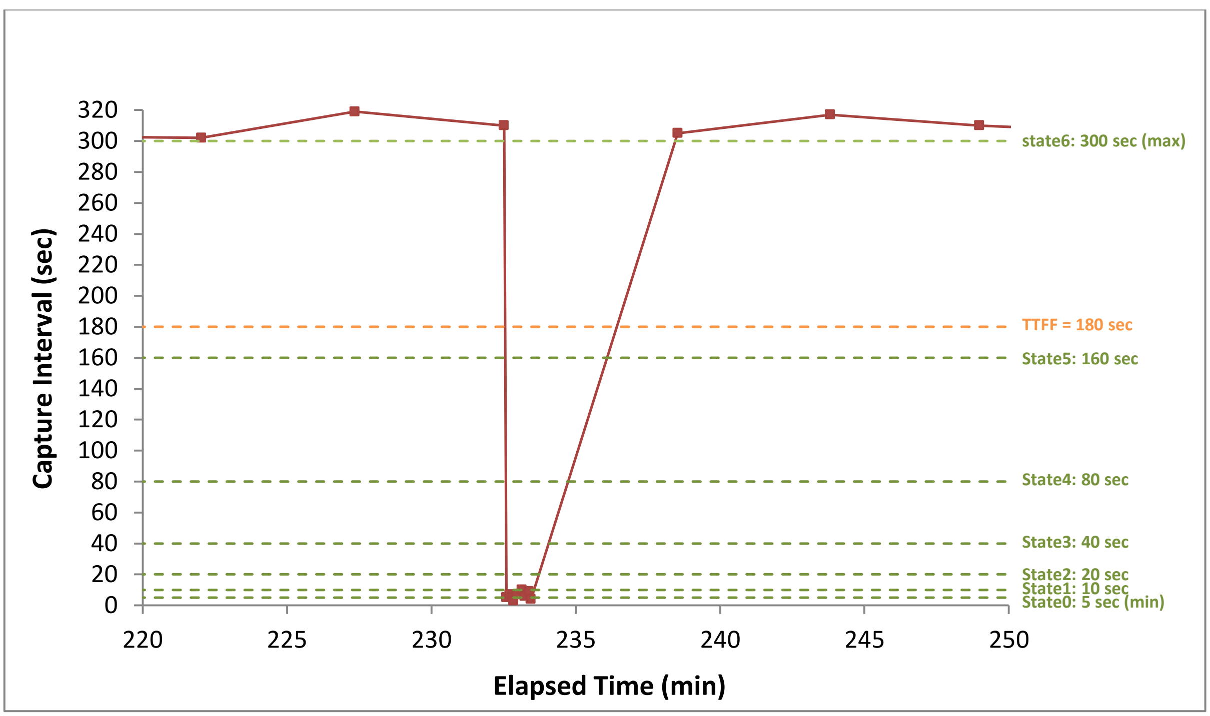


Figure 49: Capture interval as a function of elapsed time, showing error handling

4.2.2 Critical Point (CP) Algorithm

As described in section 3.3.2.4, two parameters are used in the CP algorithm to identify the critical points: Angle-threshold (θ_{th}) and Time-threshold (t_{th}). The next sections investigate the impact of these two parameters on the performance of the CP algorithm.

4.2.2.1 Angle Threshold

The impact of the angle-threshold (θ_{th}) in the performance of the CP algorithm is quantified in terms of two performance measures:

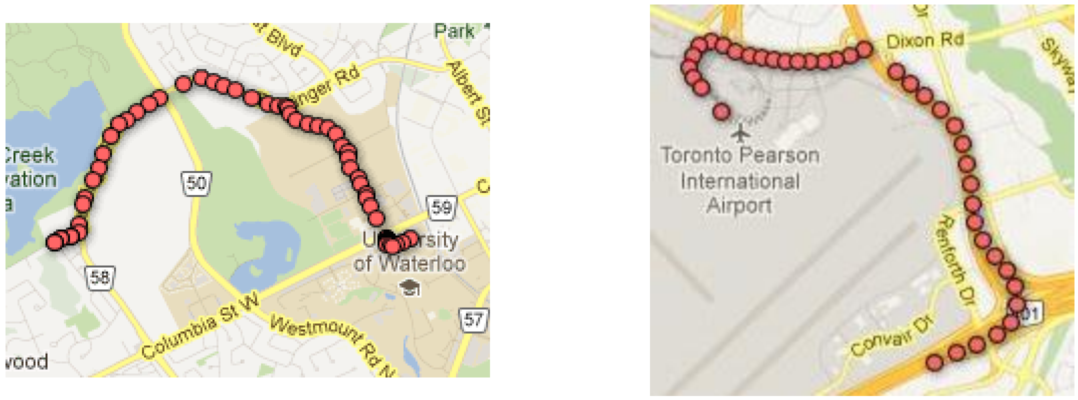
1. The reduction in the number of location points which must be transmitted to the server, and
2. The error introduced in capturing the user's trajectory.

Calibration methodology

A data set containing over 200 daily trajectories[1] was compiled for a small number of users (i.e. 5 users) over a period of several months. Data were collected using the Blackberry Bold 9700 and Blackberry Torch 9810 models. As a default, GPS location data were acquired every 5 seconds. Each daily trajectory contained one or more trip trajectories as well as the location points associated with activity stops. Trips were conducted using a variety of modes including transit (bus), walk, bike, and auto. Trips traversed a variety of infrastructure types including freeways, arterials, local roads, sidewalks, on-street bike lanes, and off-street trail. As discussed earlier, in this study, angle-threshold is used to identify critical points in trips with higher speeds. Therefore, the angle-threshold is calibrated using data collected from trips made by vehicle modes rather than non-vehicular modes (e.g. walking).

The angle threshold (θ_{th}) in the CP algorithm only has an impact when the trajectory is not straight. Consequently, it is not appropriate to evaluate the impact of θ_{th} on data containing a large proportion of straight trajectories. Therefore, a sample of 20 trajectory segments was extracted from the larger set of trip trajectories in order to represent a range of roadway types (e.g. freeway and arterial) and curvatures. Figure 50 illustrates two typical trajectory segments (All the trajectory segments are provided in Appendix B).

[1] A "daily trajectory" is the time series of location data for a single user over the course of a whole day.



a) Bearinger Road, Waterloo, ON, Canada b) Macdonald-Cartier Freeway, Toronto, ON, Canada

Figure 50: Two typical trajectory segments used for calibration of time-threshold (Source: Google map)

Results

The CP algorithm was applied to each of the 20 trajectory segments for a range of different values of θ_{th}. In total the 20 trajectory segments contained 637 location points representing a total of 64 minutes of travel. The individual trajectory errors (e_i) were computed for different values of θ_{th}. To compute the individual errors, as discussed in "Error Measurement" in section 3.3.2.4, first the Geodetic Longitude and Latitude were converted to Cartesian ECEF and then the error calculation method was applied.

Figure 51 illustrates the cumulative relative frequency distribution of these individual trajectory errors for four values of θ_{th}, and Figure 52 shows the relative frequency distribution function of trajectory errors for the four levels of θ_{th}. As expected, the results in these figures indicate that as the value of θ_{th} increases, the trajectory errors also increase. In particular in Figure 51, it is notable that not only does the mean error increase, but the proportion of large errors also increases dramatically.

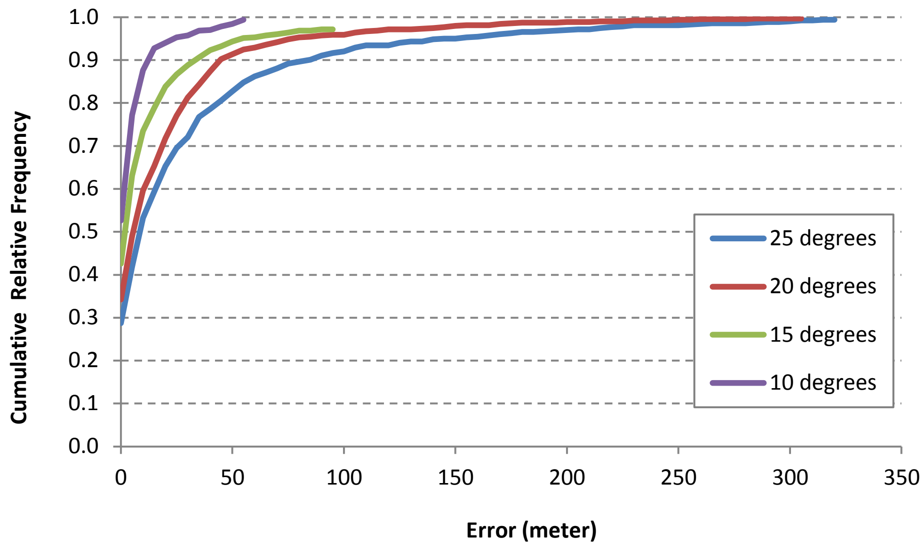


Figure 51: Cumulative distribution of trajectory errors (e_i) as a function of angle threshold

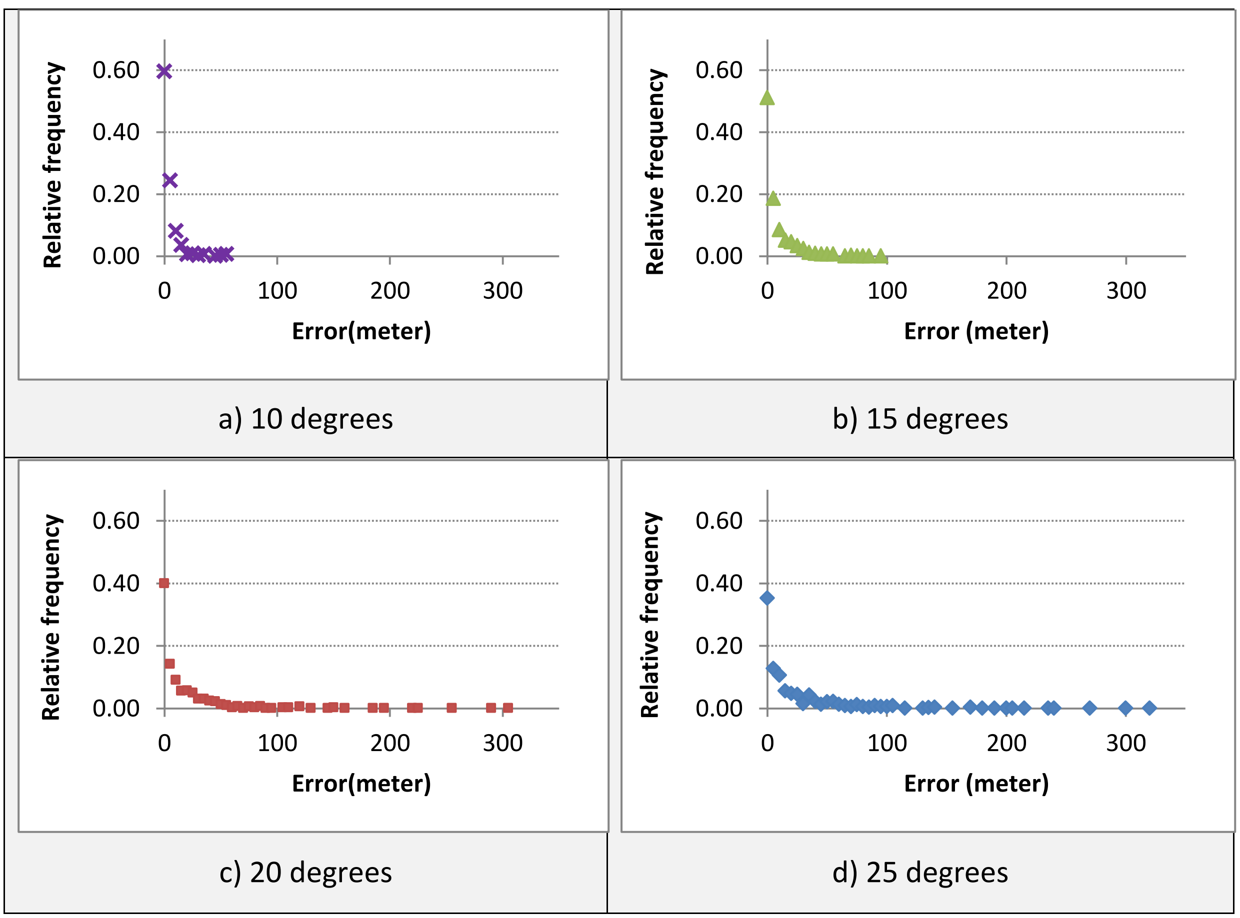


Figure 52 – Relative frequency distribution function of trajectory errors (e_i) as a function of angle threshold

From a practical perspective, in general, we are more concerned about a relatively small number of large errors than a larger number of very small errors, and therefore, the mean trajectory error is not an appropriate measure of performance. In this case, the 85th percentile error was used as the measure of performance. Figure 53 illustrates the impact of the angle threshold in terms of the 85th percentile error (dis-benefit) and in terms of reduction in the number of location points that must be sent to the server (benefit). The results show that the benefits of using the CP algorithm (in terms of the reduction in points that must be sent to the server) continue to increase over the full range of θ_{th} evaluated (2 to 60 degrees). However, the marginal benefits decline as θ_{th} increases.

Conversely, the dis-benefits of using the CP algorithm (in terms of the 85th percentile trajectory error) monotonically increase at an increasing rate as θ_{th} increases. As the benefits and costs are not expressed in the same units, it is not possible to directly determine the optimal value for $\theta_{th.}$ It is possible to devise a conversion rate between the benefit and cost in order to express them in the same unit; however, this transformation would be subjective and therefore difficult to justify. Rather, it is suggested that in practice, for a given application, there is an upper bound on the magnitude of the trajectory error that is acceptable and this places an upper bound on the value of θ_{th} that can be used. Then, in order to maximize benefits, the value of θ_{th} should be set equal to the upper bound. In the context of collecting individual travel data, a reasonable choice for θ_{th} appears to be 15 degrees as this results in an 85th percentile trajectory error of 21.8m but provides a 57% reduction in the number of location points that need to be sent to the server. Table 10 presents 85th percentile error and reduction in number of points transmitted to server for each θ_{th} evaluated in the range of 2 to 60 degrees. This table also illustrates the number of bytes saved as a result of the reduction in the number of points transmitted to the server.

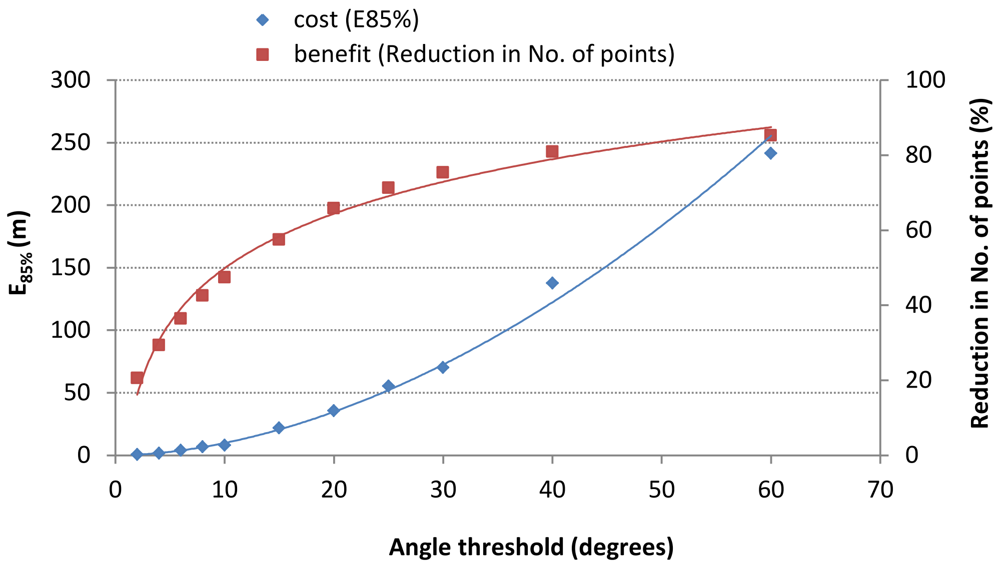


Figure 53: Impact of angle threshold on 85th percentile trajectory error and % reduction in number of location points that must be sent to the server

Table 10: 85th percentile error, reduction in number of points and bytes transmitted to server as function of angle threshold

Angle threshold (degrees)	E85% (meter)	Reduction in No. of points (%)	Bytes saved
2	0.60	21	2509
4	1.68	29	3581
6	3.91	36	4443
8	6.76	43	5190
10	7.96	47	5784
15	21.83	57	7010
20	35.51	66	8025
25	55.15	71	8695
30	70.21	75	9193
40	137.70	81	9863
60	241.48	85	10400

4.2.2.2 Time Threshold

The second parameter in the CP algorithm is the time threshold (t_{th}). This threshold has an impact when the user is moving in a straight line or is stationary for a long period of time. The main motivation for implementing the time threshold algorithm is to maximize battery life. Consequently, in this section the battery consumption is quantified as a function of the time threshold (t_{th}).

Calibration methodology

Unlike the methodology used for calibrating the angle threshold in which it was necessary to extract a sample of trajectory segments, the time threshold impacts trajectories that are straight or when the user is stationary. Consequently, a number of tests were conducted in which the time threshold value was varied and the device remained stationary. The remaining battery power, as reported by the Blackberry API, was recorded at the start and end of each 1 hour test. The device was configured to obtain GPS locations every 5 seconds (I_c = 5 seconds). Data were stored on the device and transmitted to the server every 15 minutes (I_t = 15 minutes). Six levels were considered for the time- threshold (t_{th} = {5, 15, 30, 60, 120, and 300 seconds}). Each test was repeated 25 times. Two identical Blackberry Torch 9810 devices were used for the tests. There was no specific pattern in selecting the time of the day for the tests.

Results

The results from the tests are presented in Table 11 and Figure 54. Table 11 provides the average and standard deviation of the battery consumption per hour as a percent of the battery life. The average battery consumption rate ranges from 5.9%/hr when t_{th} = 300 seconds to 7.3%/hr when t_{th} = 5 seconds. Note that when t_{th} = 5 seconds all points are labeled as critical points.

Figure 54 illustrates the battery consumption rate obtained from all of the individual tests (25 for each value of t_{th}). It should be noted that the battery consumption rate is reported by the Blackberry device as an integer value and therefore many of the points have the same value.

Based on the results in Table 11 and Figure 54, there appears to be a large variation in the battery consumption rate which is somewhat surprising. Therefore additional analysis was conducted to identify the source of the large variation in the results.

Table 11: Summary results of battery consumption as a function of time threshold (t_t)

Time threshold (sec)	Average battery consumption in 1hr (%)	Standard deviation of battery consumption in 1hr	Min required No. of runs*
5	7.28	1.59	20
15	6.20	1.22	17
30	6.48	1.16	14
60	5.84	1.14	16
120	5.48	1.33	25
300	5.88	1.36	23

*Calculated according to Appendix D, Confidence interval = 95%, Maximum acceptable error = 10% of average battery consumption, and the initial number of runs = 25.

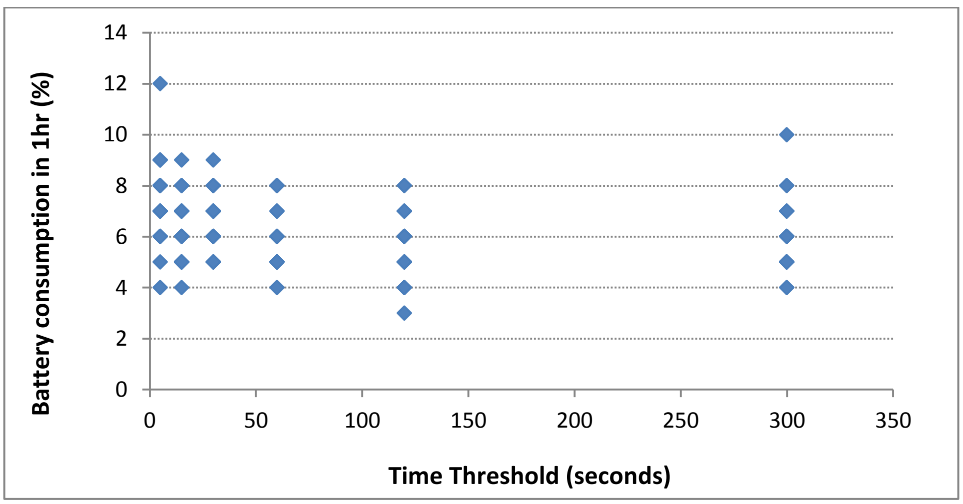


Figure 54: Battery consumption as a function of time threshold (t_t)

Investigation of the Source of High Variation

The large variance of battery consumption at each level of time threshold (t_{th}) might result from different sources which are studied in this section:

Initial battery level

One possibility for the variation in data might be that the battery consumption rate is dependent on the initial level of the battery. As an example, the power consumption rate may be lower when the device is fully charged versus when the battery is already partially depleted. In order to investigate this possibility, the power consumption rate (battery level consumed per hour) is plotted versus the battery level at the start of the hour, and a linear regression is conducted to check if the coefficient of the slope in the plotted graph is statistically different from zero or not. If the slope is zero, it is concluded that the battery consumption rate is constant over time and it is not the source of variability in the data.

The results of the regression analysis are illustrated in Figure 55 and Table 12. The results indicate that the slope is not statistically different from zero suggesting that there is no evidence that initial battery life is the source of the variation in Figure 54. The regression results for individual levels of time threshold (t_{th} = {5, 15, 30, 60, 120, and 300 seconds}) are also illustrated in Appendix C. These results show that none of the coefficients of the slopes for different levels of time threshold is significant, suggesting that the battery consumption rate is constant at each level of time threshold.

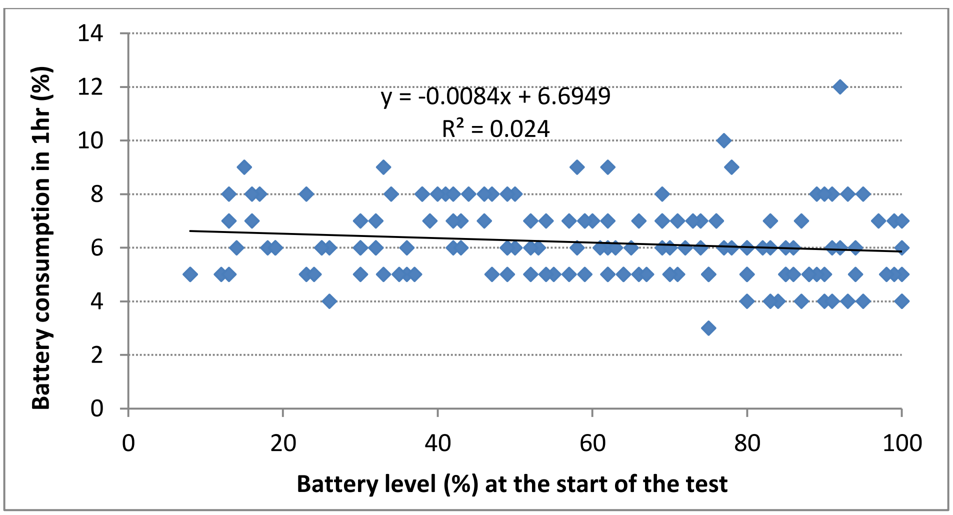


Figure 55: Battery consumption rate versus initial battery level

Table 12: Regression summary for battery consumption against initial battery level

Variable	Coefficient Value	T-Statistic
Initial battery level	-0.01	-1.91
Intercept	6.69	23.37
No. of Obs. = 150	Corrected R2= 0.02	

<u>Difference between the devices</u>

Although the two devices used for this experiment were both Blackberry Torch-9810, it was suspected that for the same operating conditions the power consumption rate for one of the devices was larger than for the other. In order to check if the devices are identical in terms of battery consumption or not, a sample data was collected for each device and then the mean battery consumption of the two devices has been compared through a hypothesis test (details are provided in Appendix D).

H_0: $\mu_1 = \mu_2$ H_1: $\mu_1 > \mu_2$

μ_1: Mean battery consumption by Device 1

μ_2: Mean battery consumption by Device 2

For this experiment, the both devices were configured to capture the data at capture interval (I_c) of 5 seconds and transmit the data to server at transmission interval (I_t) of 15 minutes. The time threshold level (t_{th}) of 2 minutes was considered for the CP algorithm. The accelerometer has been turned on for both devices and the devices remained stationary at the same location with a full view of the GPS satellites. For one of the devices data was captured for 16 hours and for the other one data was captured for 12 hours[1].

If each hour of data capturing is considered as a single observation, then the results from the two data sample can be summarized as follows in Table 13:

Table 13: Summary results of battery consumption for two Blackberry Torch-9810 devices

Device No.	Average battery consumption in 1hr (%)	Standard deviation of battery consumption in 1hr	Initial No. of runs	t_{n-1}	Min required No. of runs*
1	6.06	0.85	16	2.13	9
2	6.42	0.67	12	2.20	5

*Calculated according to Appendix D, Confidence interval = 95%, Maximum acceptable error = 10% of average battery consumption

The results of the hypothesis test can be summarized as follows:

$\nu = 25.92$, $|(t_o = -1.23)| < (t_{0.05,\nu} = 1.706)$

Therefore we fail to reject the null hypothesis, and as a result with 95% confidence we accept that the battery consumption rate of the two devices is the same.

[1] Note that the reason for difference in length of time in capturing the data by the two devices is that they did not start from the same battery level. Moreover, in this experiment we do not need the same length of time for capturing the data by the two devices.

Signal quality of GPS satellites

Signal quality of satellites might also be a reason for the variation in data in terms of battery consumption. When signal quality is weak, it results in frequent attempts by the device to obtain a GPS position from the satellite signal and therefore more battery is consumed. In order to analyze the impact of signal quality on the battery consumption rate, two identical Blackberry devices were employed to collect the data at the same time and at the same location. When the devices are placed at the same location and are tested during the same time, the signal quality of the satellites should affect the battery consumption of the two devices equally. Therefore, by comparing the results of the collected data it can be identified if there is a correlation between the devices in terms of battery consumption or not. For this experiment, the data captured through the previous experiment, “Difference between the devices”, was used, and the 12 hours of simultaneous data collection by the two devices were analyzed.

The results of this analysis are illustrated in Figure 56[1] and Figure 57. Based on these results, there is no evidence that the two devices have correlation in battery consumption rate, while receiving the same satellite signals. Therefore, we cannot conclude that the source of variability in the battery consumption rate is the signal quality of the satellites.

[1] Note that in Figure 56 the battery consumption rates are divided by average battery consumption rate of the related device

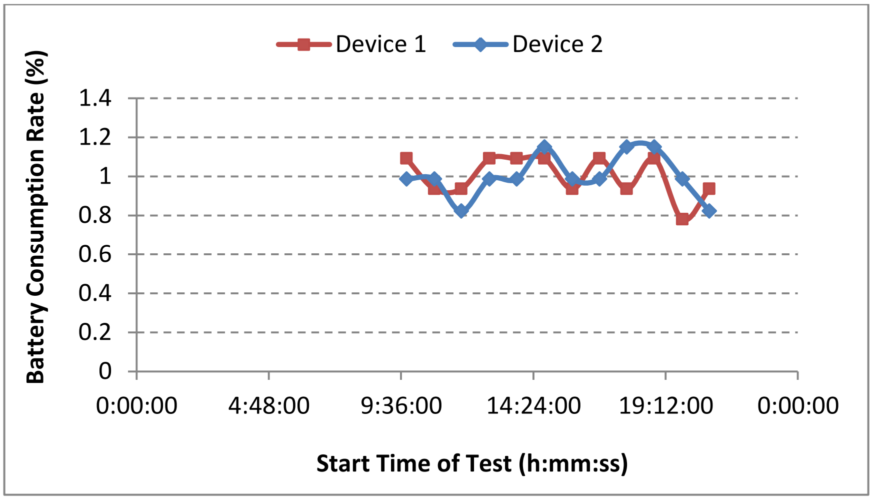


Figure 56: Comparison of battery consumption rate by the two devices receiving same satellite signals

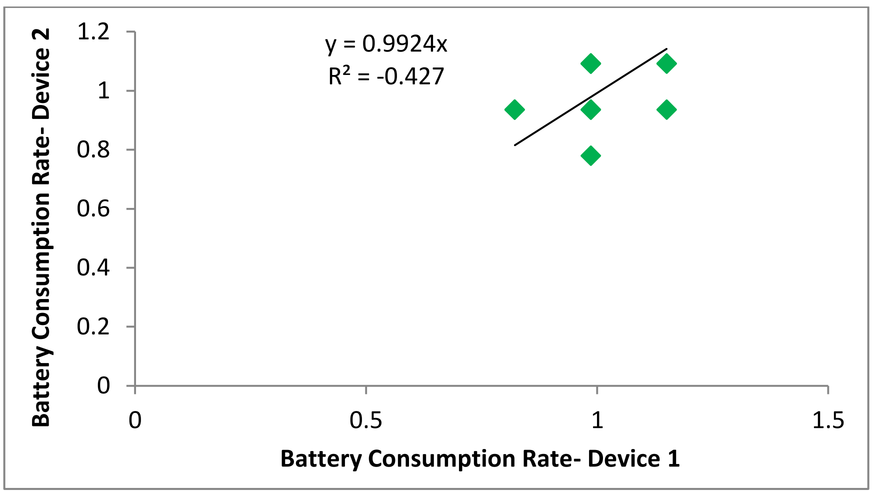


Figure 57: Correlation of battery consumption by the two devices receiving same satellite signals

Transmission Interval

When the value of transmission interval (I_t) is large (e.g. 15 minutes) and the value of capture interval (I_c) is small (e.g. 5 seconds), if for any reason the GPS data is not transmitted to the server at the specified frequency then a large amount of data is queued up on the device to be sent to server at the next transmissions. Consequently transferring a large amount of data might affect the battery consumption rate. In order to identify how a low frequency in transmission time affects the battery consumption rate, the correlation between the transmission intervals and battery consumption rate can be examined. For this purpose, the data captured in section "Difference between the devices" analyzed. Examining the captured data shows that for the whole duration of the tests for the two devices (12 + 16 = 28 hours) the data were transmitted to server exactly at 15 minutes intervals. Therefore, no low frequency in transmission times occurred in the collected sample. However, as illustrated in Figure 56, both devices still reveal variability in battery consumption rate during the test hours. As a conclusion, there exist some other sources which cause variability in battery consumption rate of the device.

Inaccuracy of reported battery level

Having eliminated the identified possible sources of the variation, we conclude that the variation results from a source which is not examined. One such possibility may be limitations in the accuracy of the reported battery level.

Regression Analysis

In order to analyze the impact of time threshold level on battery consumption, a linear regression has been conducted on the data. The results of the regression can be seen in Table 14 and Figure 58 which show that the value of the time threshold coefficient is not statistically significant. In other words, the time threshold does not appear to have a significant impact on the battery consumption rate.

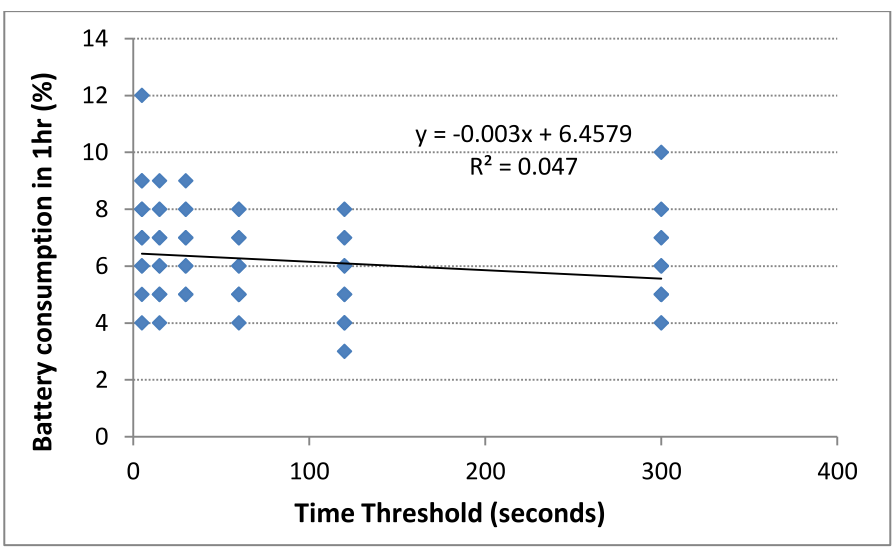


Figure 58: Battery consumption as a function of time threshold for 25 replications

Table 14: Regression results of battery consumption against time threshold (t_t) for 25 replications

Variable	Coefficient Value	T-Stat
Time-threshold	-0.003	-2.70
Intercept	6.46	43.19
No. of Obs. = 150	Corrected R^2= 0.04	

However, consider Figure 59, which illustrates the average battery consumption rate as a function of time threshold (t_{th}). A log-linear regression model was calibrated to these data and examination of the regression results shows all coefficients to be statistically significant (Table 15). Moreover, the negative sign of the time threshold (t_{th}) coefficient (-0.36) makes intuitive sense, because we expect that as the time threshold increases, fewer points are transmitted to the server, and the battery consumption rate decreases.

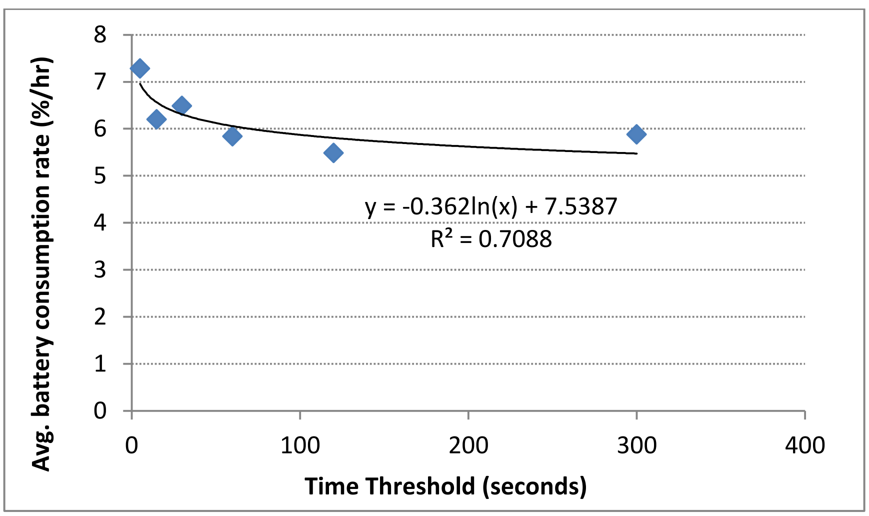


Figure 59: Mean battery consumption rate as a function of time threshold

Table 15: Regression results of battery consumption against logarithm of time threshold

Variable	Coefficient Value	T-Stat
Ln (Time-threshold)	-0.36	-4.47
Intercept	7.54	23.55
No. of Obs. = 150	Corrected R^2= 0.7	

Residual Analysis

As a means to check if the residulas are independent and are identically distributed (i.i.d.), residuals are plotted versus the fitted values of battery consumption and also versus Ln (time threshold). The results are illustrated in Figure 60 and Figure 61 which do not reveal any unusual pattern. Figure 62 also shows the normal probability plot, which indicates that the error is normally distributed. As a conclusion none of our assmptions regarding the residuals to be i.i.d. is violated.

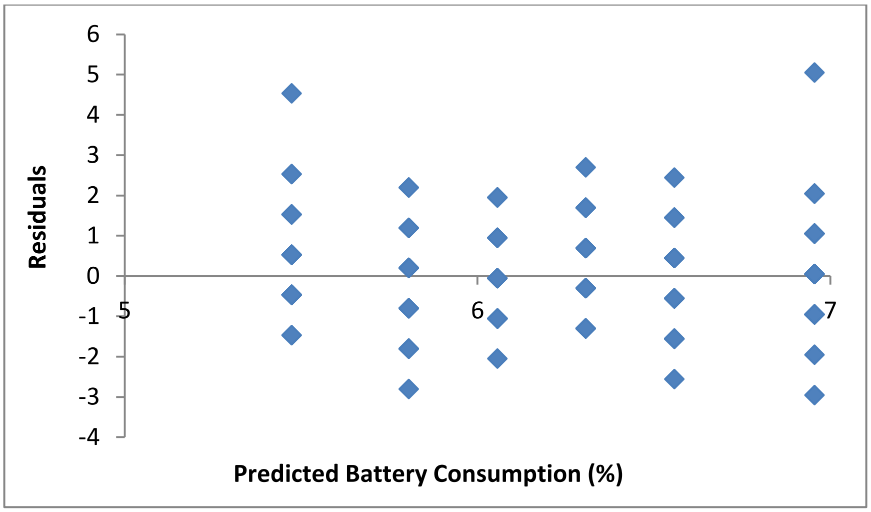


Figure 60: Residuals versus fitted values of battery consumption

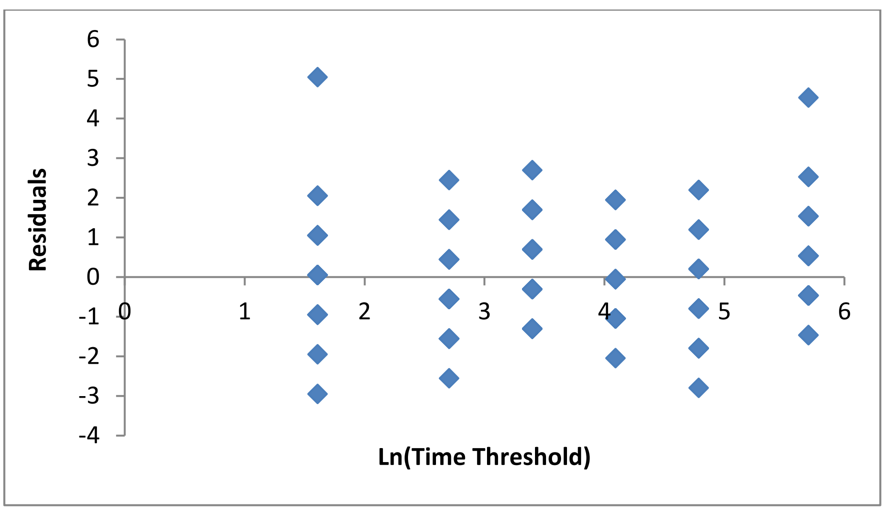


Figure 61: Residuals versus logarithm of time threshold

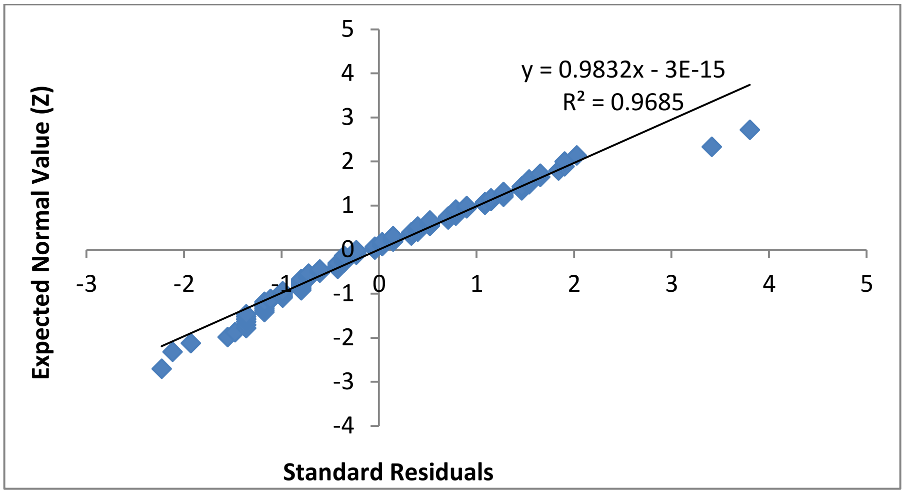


Figure 62: Normal Probability Plot

Selecting the Optimum time-threshold

As a result of our statistical analysis in the previous section we realized that the relationship of the battery consumption and the logarithm of the time threshold is linear. The objective in this section is to select the optimum time threshold such that the battery consumption is minimized. For this purpose, a comparison analysis (detailed calculations are provided in Appendix D) is conducted on the data to compare each two time threshold (t_{th}) levels in terms of the battery consumption rate.

H_0: $\mu_1 = \mu_2$ H_1: $\mu_1 > \mu_2$

μ_1: Mean battery consumption rate when using the smaller time threshold (t_{th})

μ_2: Mean battery consumption rate when using the larger time threshold (t_{th})

Table 16 shows the calculated degree of freedom (ν) for each pair of time threshold level. t student value ($t_{\alpha, \nu}$) for the calculated ν with the confidence level of 97.5% is also displayed in Table 17. Finally observed t (t_o) is calculated for each pair and then is compared to $t_{\alpha, \nu}$. The calculated observed t and the results of comparison are illustrated in Table 18.

Table 16: Calculated degree of freedom (ν) for each pair of time threshold level

Time threshold level (sec)	5	15	30	60	120	300
5	---	45	44	44	46	47
15	---	---	48	48	48	47
30	---	---	---	48	47	47
60	---	---	---	---	47	47
120	---	---	---	---	---	48
300	---	---	---	---	---	---

Table 17: t student value ($t_{\alpha,\nu}$) for the calculated ν and with the confidence level of 97.5%

Time threshold level (sec)	5	15	30	60	120	300
5	---	2.014	2.015	2.015	2.013	2.012
15	---	---	2.011	2.011	2.011	2.012
30	---	---	---	2.011	2.012	2.012
60	---	---	---	---	2.012	2.012
120	---	---	---	---	---	2.011
300	---	---	---	---	---	---

Table 18: Calculated observed t for each pair of time threshold level

Time threshold level (sec)	5	15	30	60	120	300
5	---	2.69*	2.03*	3.67*	4.34*	3.34*
15	---	---	0.83	1.07	1.99	0.87
30	---	---	---	1.97	2.84*	1.68
60	---	---	---	---	1.03	0.11
120	---	---	---	---	---	1.05
300	---	---	---	---	---	---

*$|t_o| > t_{\alpha,\nu}$ meaning the null hypothesis is rejected and the time threshold levels are significantly different

According to the results presented in Table 18, battery consumption rate for t_{th} = 5s is significantly higher than all other levels of time threshold. The results also show that battery consumption is different between t_{th} = 30s and t_{th} = 120s; however, it can be ignored since there is no other evidence of difference between battery consumption rate when t_{th} is greater than 5s. Given that we wish to acquire data as frequently as possible, and there are not significant battery consumption reductions to implementing t_{th} larger than 15s, it is recommended to use t_{th} = 15s.

4.3 Battery Consumption Analysis

4.3.1 Battery Consumption when device is idle

This section investigates the amount of battery consumption when the device is idle, i.e. the application is off and no other operation is performed by the device. The purpose of this experiment is first to find out how much battery is consumed per hour in the idle case, and second to realize if the battery consumption rate is linear over time or not. For this purpose, a Blackberry Torch 9810 was employed while it was fully charged. The device was left idle until the battery was drained. At different time points during the test the battery level of the device was recorded.

The results of this experiment showed that the battery was drained completely after 119.5 hours. Figure 63 illustrates the recorded battery levels versus the elapsed time. According to the results, battery consumption rate is linear over time and it is 0.9 % per hour.

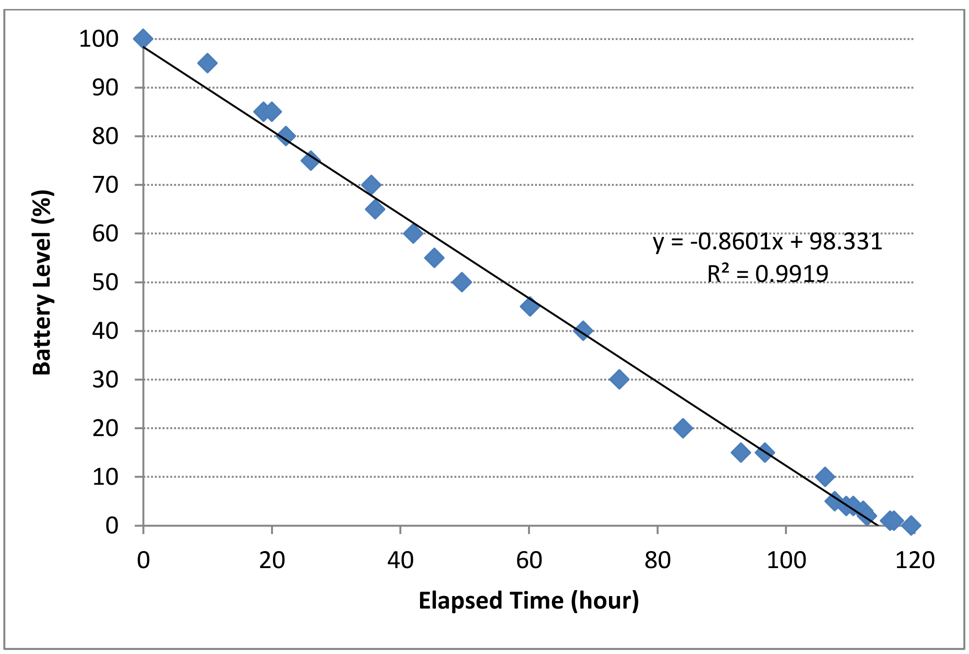


Figure 63: Idle battery consumption per hour

4.3.2 Battery Consumption in base case

The purpose of this experiment is to analyze the battery consumption rate of the device when the application is running in the base case. In the base case, no optimization algorithm is running to reduce the battery usage (SM algorithm and CP algorithm are off). Also transmission interval is frequent (I_t = 1 minute) and accelerometer is on which increases the battery consumption. This case can be considered as the worst case of data collection by the application in terms of battery consumption. For this purpose an experiment conducted using a Blackberry Torch 9810. The device was utilized at stationary mode at a location with full view of satellites. The GPS data was captured at the fixed capture interval (I_c) of 5 seconds (SM algorithm: off), and Assisted GPS mode was used to capture the data. The CP algorithm was turned off, the accelerometer was turned on, the transmission Interval (I_t) of 1 minute was used, and the Wi-Fi was turned off to avoid wireless connection failure. The test started at 100% battery level and it continued until the battery drained.

The results of this experiment are illustrated in Figure 64 and Figure 65. Figure 64 shows the battery life versus the elapsed time. According to this figure, at two points (green arrows) GPS location updates were not available as per request. However, the figure shows that battery consumption rate is linear over time when positions are captured at fixed time intervals. According to the linear regression conducted, about 13% of battery is consumed per hour at the base case, and battery drains after 7.5 hours.

It should be noted that the current experiment is conducted in the same condition as the “fix interval” experiment conducted in section 4.2.1.1, except that in the current experiment Wi-Fi is turned off. However, the experiment with Wi-Fi on resulted in 12 hours of battery life and the current experiment resulted in 7.5 hours of battery life. Although according to these results Wi-Fi has significantly increased the battery life, it is not recommended for connection to wireless network. The reason is that as explained previously Wi-Fi does not provide a reliable network connection as data capturing might fail due to the failure of the connection. Moreover, the purpose of the application is to collect travel behavior data and Wi-Fi is typically not available to devices in a mobile environment.

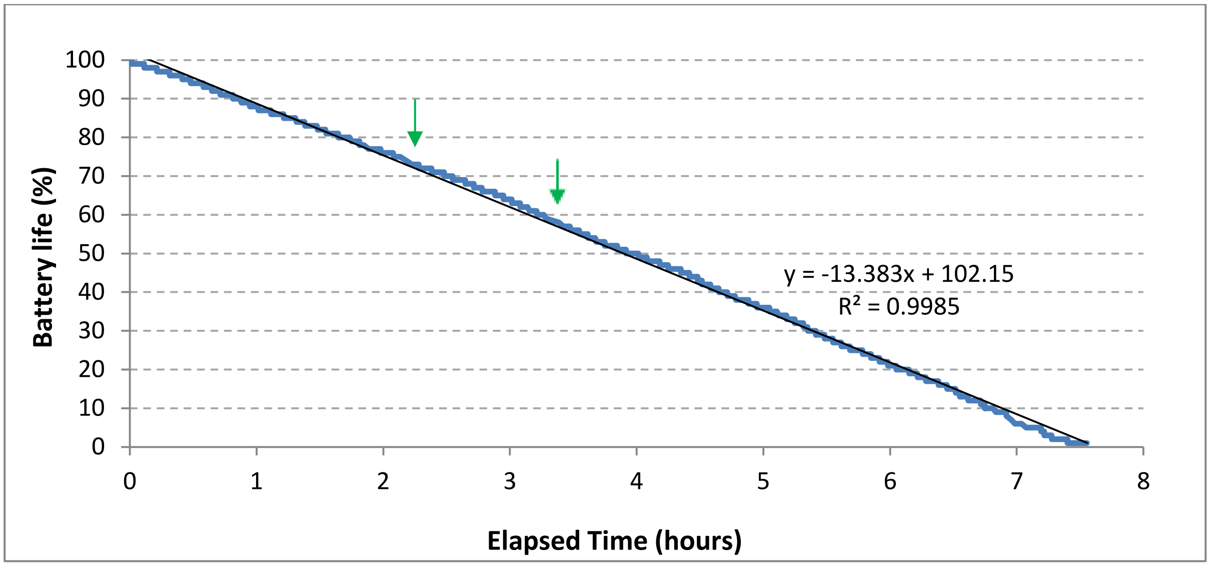


Figure 64: Battery life as a function of elapsed time – base case

Figure 65 shows the battery life as a function of cumulative number of fixes. Green arrows on this figure show the points at which frequent GPS updates were not available and as a result fewer position fixes were obtained. At these points the battery still has been consumed, because the device has attempted to capture new fixes. This figure shows that battery consumption rate is linear as a function of cumulative number of fixes when the positions are captured at fixed time intervals.

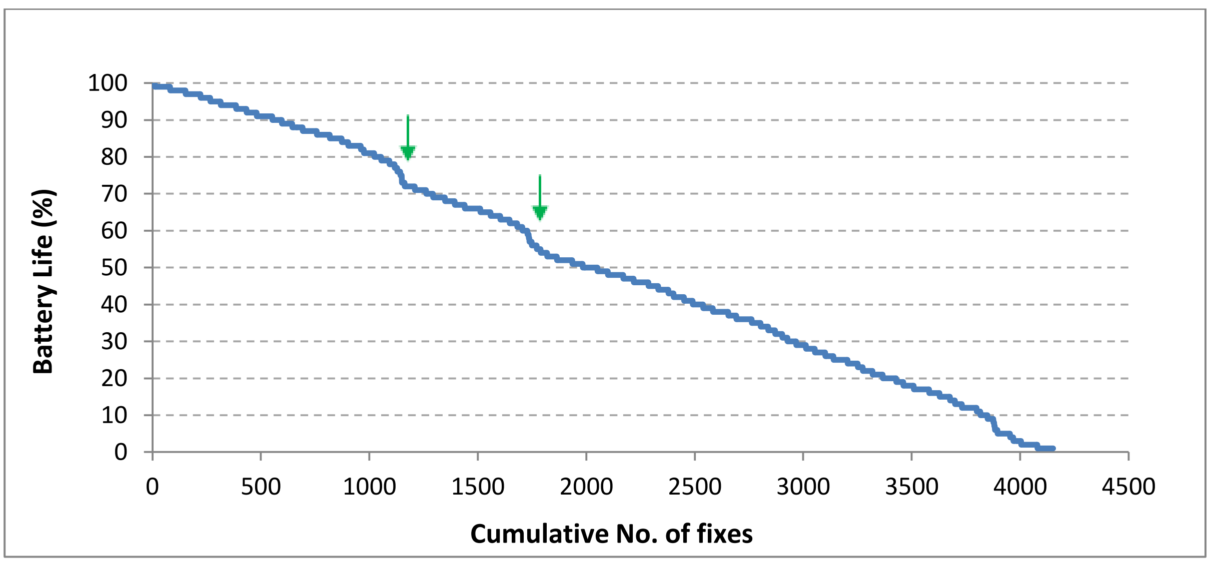


Figure 65: Battery life as a function of cumulative number of fixes – base case

4.3.3 Impact of Transmission Interval (I_t) on battery life

In this experiment, the impact of transmission interval (I_t) on battery consumption rate is analyzed. For this purpose, an experiment was conducted using a Blackberry Torch 9810. The device was utilized at stationary mode at a location with full view of satellites. The GPS data captured at the fixed capture interval (I_c) of 5 seconds (SM algorithm: off), and Assisted GPS mode was used to capture the data. The CP algorithm was turned off, and the accelerometer was turned on. In fact, the experiment has been conducted in the base case, except that six values have been considered for transmission interval (1, 5, 10, 15, 20, 30 minutes) and the test has been repeated for each level. Each test started at 100% battery value and it continued until the battery drained.

The results of this experiment are illustrated in Table 19 and Figure 66. According to the results, by increasing the transmission interval from 1 minute to 15 minutes the battery life significantly increases. This result is expected, because as explained in Literature Review frequent transmissions to the server increase the battery consumption rate (due to frequent wireless network usage). On the other hand, Figure 66 shows a drop off in battery life when transmission interval is greater than 15 minutes. The reason for this result is that when transmission interval becomes large, many points are collected on the device which should be transmitted to server altogether. For example when $I_t > 15$ min, more than 180 points are transmitted to server at each transmission (15min * 12point/min = 180 points). The transmission of large number of points also increases the battery usage. This is similar to the problem of finding optimum value for time threshold (t_{th}) parameter of CP algorithm which is discussed in section 4.2.2.2. In the analysis we conducted in section 4.2.2.2, we assumed that I_t = 15 minutes, then we found that at t_{th} = 5 sec the battery consumption significantly increases comparing to the larger t_{th} values. At t_{th} = 5 sec, 180 points were transmitted to server at each transmission (15 min * 12 point/min = 180 points), and at larger t_{th}, less points were transmitted to server.

As a conclusion, 15 minutes is recommended to be used as optimum transmission interval (I_t) value, since it maximizes the battery life.

Table 19: Impact of Transmission Interval (I_t) on battery life

Transmission Interval (min)	Battery Life (hours)
1	7.5
5	9.5
10	14.25
15	15.83
20	15.17
30	14

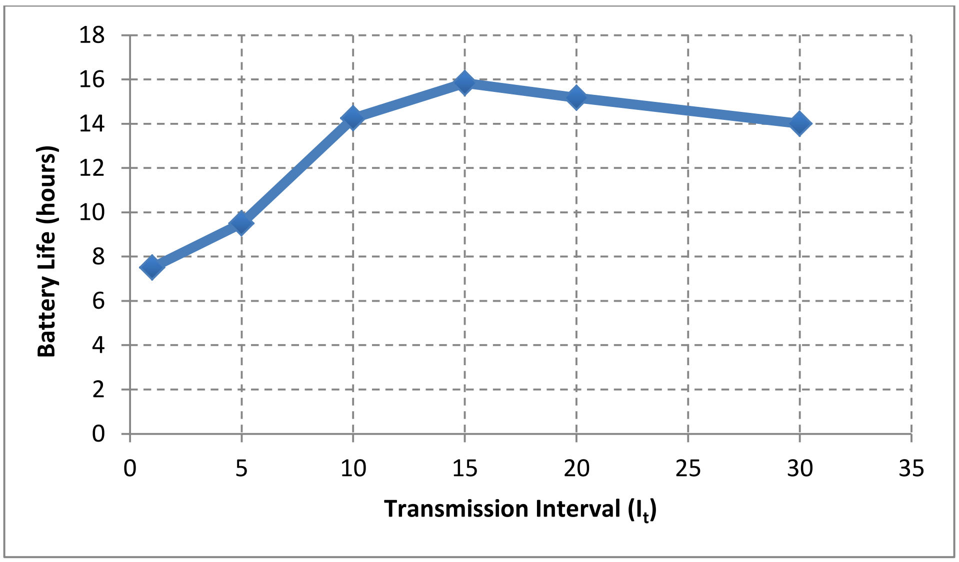


Figure 66: Battery life as a function of Transmission Interval (I_t)

4.3.4 Impact of Accelerometer on battery life

In this experiment, the impact of accelerometer on battery consumption rate is analyzed. For this purpose an experiment has been conducted using two Blackberry Torch 9810 devices. The devices were utilized at stationary mode and at a same location with full view of satellites. The GPS data was captured at the fixed capture interval (I_c) of 5 seconds (SM algorithm: off), Assisted GPS mode was used to capture the data, the CP algorithm was turned off, and the transmission interval (I_t) of 1 minute was used to transmit the data to server. In fact, the experiment has been conducted in the base case, except that two conditions for accelerometer were considered (on/off): one of the devices was configured to use the accelerometer (accelerometer on), and the other was configured to not use the accelerometer (accelerometer off). Both tests started at 100% battery level and they continued until the battery drained.

The results of the experiment showed that the device with accelerometer on drained the battery after 7.5 hours. This result is expected because this test has been conducted in the same condition as the base case test discussed in section 4.3.2, and we expect to see the similar results for both tests in terms of battery consumption. The results of the experiment also showed that the device with accelerometer off drained the battery after 9.5 hours. Therefore, accelerometer has decreased the total battery life by 21.1%.

4.3.5 Battery Life and travel behavior data collection in optimum case

The purpose of this experiment is first to measure the battery consumption rate per hour for the optimum case, i.e. the SM algorithm and CP algorithm are on, accelerometer is off and transmission interval is at its optimum value (i.e. 15 min), then, to conduct an analysis on travel behavior data collection in optimum the optimum case.

Given that SM algorithm behaves differently when the user is at stationary or non-stationary mode, the battery consumption results of this experiment depend on the amount of time spent in stationary and non-stationary modes. Furthermore, the CP algorithm's impact on battery consumption rate depends on the specific characteristics of the route being travelled. Therefore, a simplified experiment first has been conducted in a simulated case to roughly measure the battery consumption rate when user is at non-stationary mode. Next, the experiment has been conducted for a few numbers of real daily trips, which typically contain a limited time of non-stationary modes. Then, the battery consumption rate and travel behavior data collection have been analyzed for the captured real daily trips.

Simulated case

For the simulated case, the non-stationary mode has been simulated using the "Speed Simulator" explained in section 3.2. For this case, two tests have been conducted using two Blackberry Torch 9810 devices. Each test included both stationary and non-stationary modes. Test 1 started with 9 hours of stationary mode (speed $\leq S_{th,s}$), then continued with 8 hours of non-stationary mode (speed $> S_{th,s}$), and finally ended with 29 hours of stationary mode (speed $\leq S_{th,s}$). Test 2 started with 4 hours of non-stationary mode (speed $> S_{th,s}$) and ended with 39 hours of stationary mode (speed ≤ $S_{th,s}$). In fact, for both tests the devices were stationary throughout the test, at a location with clear view of satellites. Then the non-stationary sections were configured by changing the speed to a value greater than $S_{th,s}$ using "Speed Simulator". For this experiment, SM algorithm has been turned on and configured to minimum capture interval ($I_{c,min}$) of 5 seconds and maximum capture interval ($I_{c,max}$) of 5 minutes. CP algorithm has also been turned on and time threshold (t_{th}) was set equal to1 minute[1]. The transmission interval (I_t) was set to the optimum value of 15 minutes, and accelerometer was turned off. The non-stationary mode simulated in this condition represents a trip condition at which user is traveling on a straight line for a long time and without any interruption.

The results of the simulated experiment are illustrated in Figure 67 and Table 20. Figure 67 shows the battery life of the two devices versus elapsed time. The parts with sharp slope in the two tests show the non-stationary parts of the tests at which battery is consumed at a higher rate. The parts with lower slopes show the stationary parts of the tests at which battery is consumed with lower rate. The two tests indicate that the battery life versus elapsed time is not linear for the entire duration of the test, as it includes both stationary and non-stationary parts.

Table 20 shows average hourly battery consumption calculated for stationary and non-stationary parts separately, based on the results from the two tests. The calculation of battery consumption for stationary part is itself divided into two parts: the first part is related to the time when the state machine starts from the first state to reach to the last state. The second part is related to the time at which state machine is in the last state. It is expected that the battery consumption rate for the non-stationary mode is higher than for the stationary mode, and also for stationary mode the battery consumption rate in the first part is higher than the battery consumption rate in the second part. The reason, as explained earlier, is that at each part the GPS points are captured at different frequencies due to frequency management by SM algorithm. The two tests combined included 12 hours of non-stationary mode and 77 hours of stationary mode. From these 77 hours, 4 hours were related to the initial parts of the stationary modes (i.e. transition of state machine from first state to last state). In the

[1] Note that although the optimum value obtained for t_{th} is 15s, other values greater than 15s could also be used for this experiment because previously we found that there is not a statistical difference between battery consumption of t_{th} = 15s through t_{th} = 300s.

calculation of battery consumption rate for both parts of stationary mode, it was possible for the reported speed to contain error and therefore be greater than the stationary speed threshold (i.e. speed > $S_{th,s}$). This condition occurred 10 times throughout the tests, and in each case, the state machine returned back to the first state to handle the error (as explained in section 3.3.2.2).

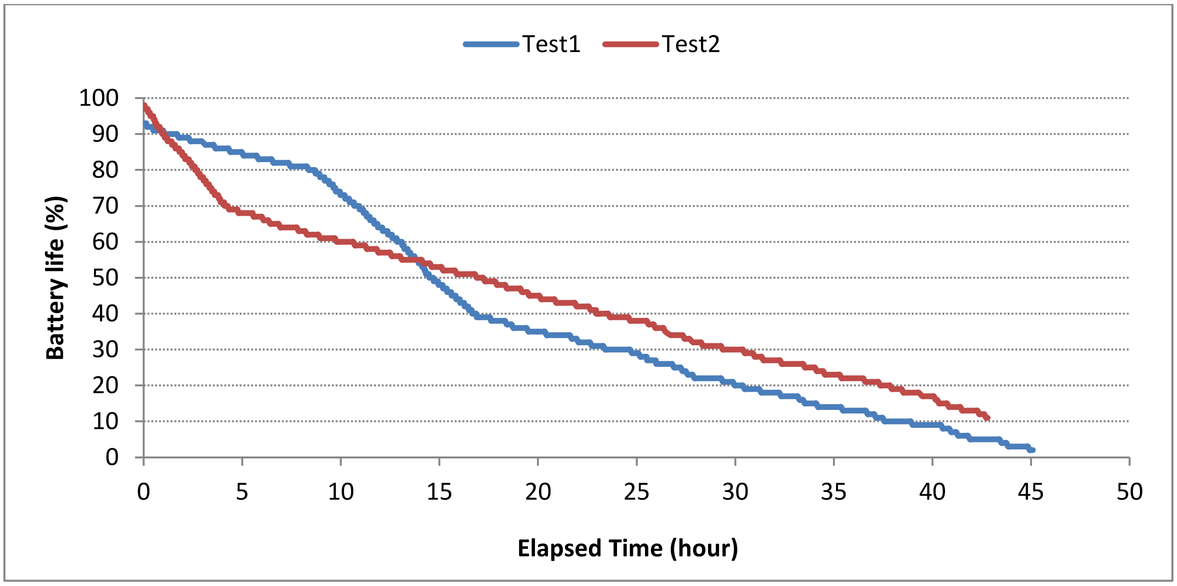


Figure 67: Battery consumption as a function of time for two designed experiments, including stationary and non-stationary modes

Table 20: Average battery consumption for stationary and non-stationary modes

Trip Section	Average battery consumption (%/hr)
Non-stationary mode	5.75
Stationary mode – first part	3.33
Stationary mode – second part	1.41

As a conclusion, comparing the stationary and non-stationary mode shows that the battery consumption rate is reduced by 75.5% (from 5.75% to 1.41% per hour) as a result of applying the SM algorithm. Moreover, the battery consumption rate at stationary mode, 1.41 %/hr, is only 0.5 %/hr more than the case when the device is idle (explained in section 4.34.3.1). Each full transition from the first state to the last state (which takes 52.5 minutes and happens when person is stationary for this period of time) consumes 3.33 %/hr of the battery life, and frequent data capturing (every 5 seconds) consumes 5.75 %/hr of the battery life. As illustrated in section 4.3.2, for the "base case" at which no optimization algorithm is used, the battery consumption rate for frequent data capturing was calculated as 13.3 %/hr. Therefore, the developed optimization algorithms and features reduce the battery consumption rate by 57%. (from 13.3 %/hr to 5.75 %/hour) when person is moving.

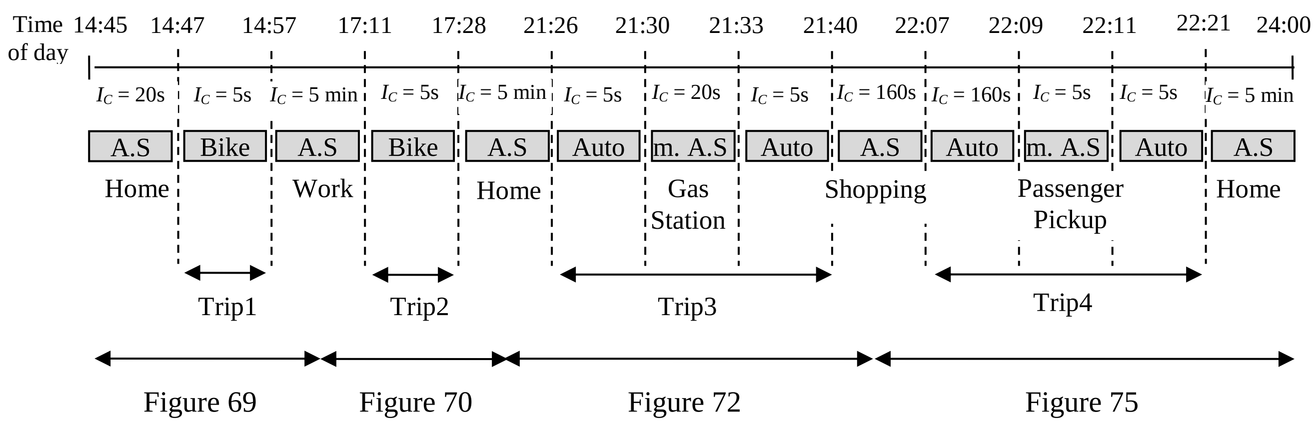
Real case

For this experiment, two Blackberry Torch 9810 devices were employed. One device was utilized in the base case, i.e. SM and CP algorithms: off; accelerometer: on; and I_{t} = 1 minute. The other device was utilized in the optimum case, i.e. SM algorithm: on; $I_{c,min}$ = 5 seconds; $I_{c,max}$ = 5 minutes; CP algorithm: on; t_{th} =15 seconds; θ_{th} = 15 degrees; accelerometer: off; and I_{t} = 15 minutes. Then the two devices were used simultaneously to record a small sample of daily trips. This section provides an analysis on the recorded daily trips by a user and the amount of battery consumed through the data collection.

Figure 68 illustrates the timeline of the recorded daily trips by the user (as well as the capture intervals (I_c) in the optimum case). The trip maker starts the first trip (Trip 1) from home with the trip purpose of travelling to work. The biking mode is utilized for this trip and it takes about 10 minutes to reach to the destination (Figure 69). Then, the trip maker spends about 2.25 hours at the work place, and then makes a work to home trip (Trip 2), again using the bike mode. This trip has a duration of approximately 17 minutes (Figure 70). The trip maker spends about 4 hours at home, and then takes an auto trip (Trip 3) to a retail location (trip purpose = shopping), but along the way makes a mid-trip activity stop[1] (at a gas station) for a duration of 3 minutes (Figure 72). The duration of this trip (Trip 3) is approximately 14 minutes. The trip maker spends about 27 minutes at the retail location and then drives back home (Figure 75). This return trip (Trip 4) takes approximately 14 minutes and includes another mid-trip activity stop. This mid-trip activity stop had the duration of 2 minutes and consisted of picking up/dropping off a passenger (note that the action taken at this mid-trip activity stop was reported by the trip maker and has not been inferred from the travel behavior data recorded by the system described in this thesis). At the end of Trip 4 the

[1] A mid-trip activity stop is an activity stop which is NOT a trip destination, and usually trip maker spends a short period of time in it (e.g. gas station or coffee shop)

application continues to record GPS data for 1.7 hours while the person is at home until the device automatically turns off at 24:00.



A.S = Activity Stop, m. A.S = Mid Activity Stop

Figure 68: Timeline of the daily trips recorded by a user

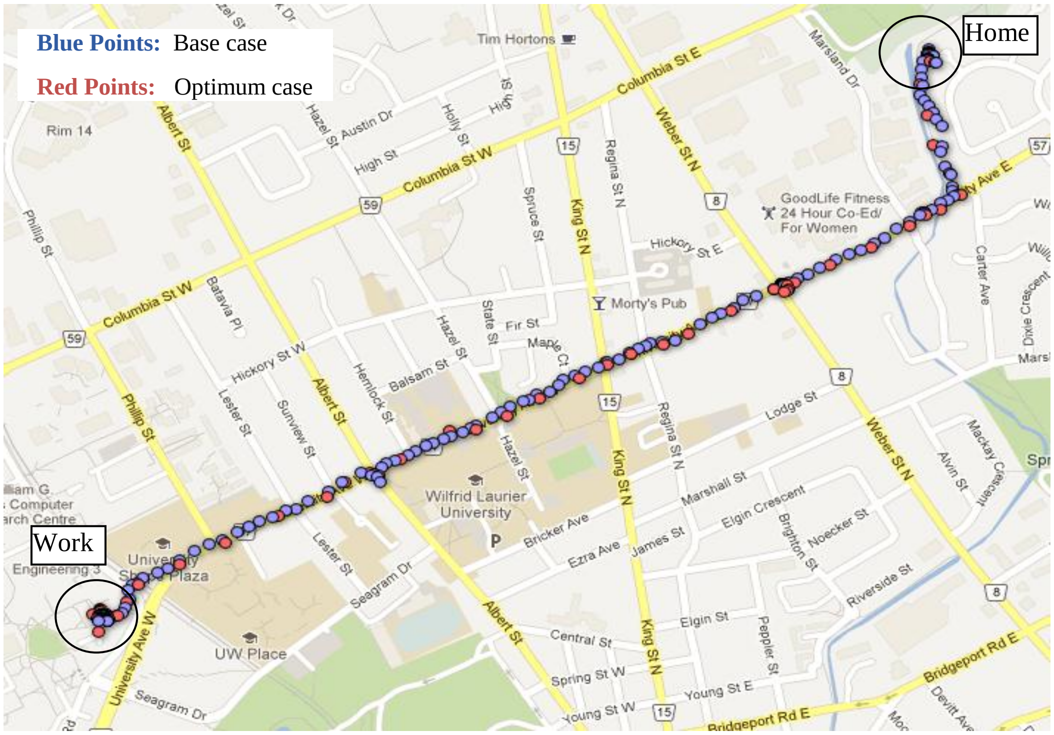


Figure 69: Home to work trip (Trip 1) - Source: Google map

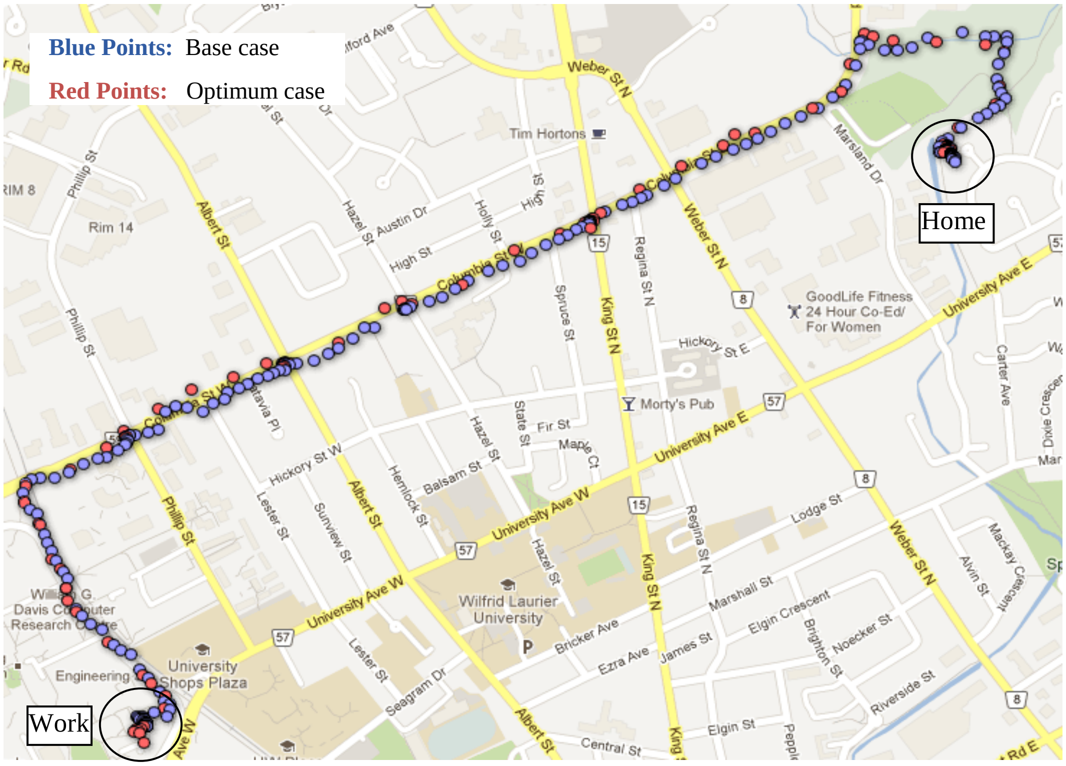
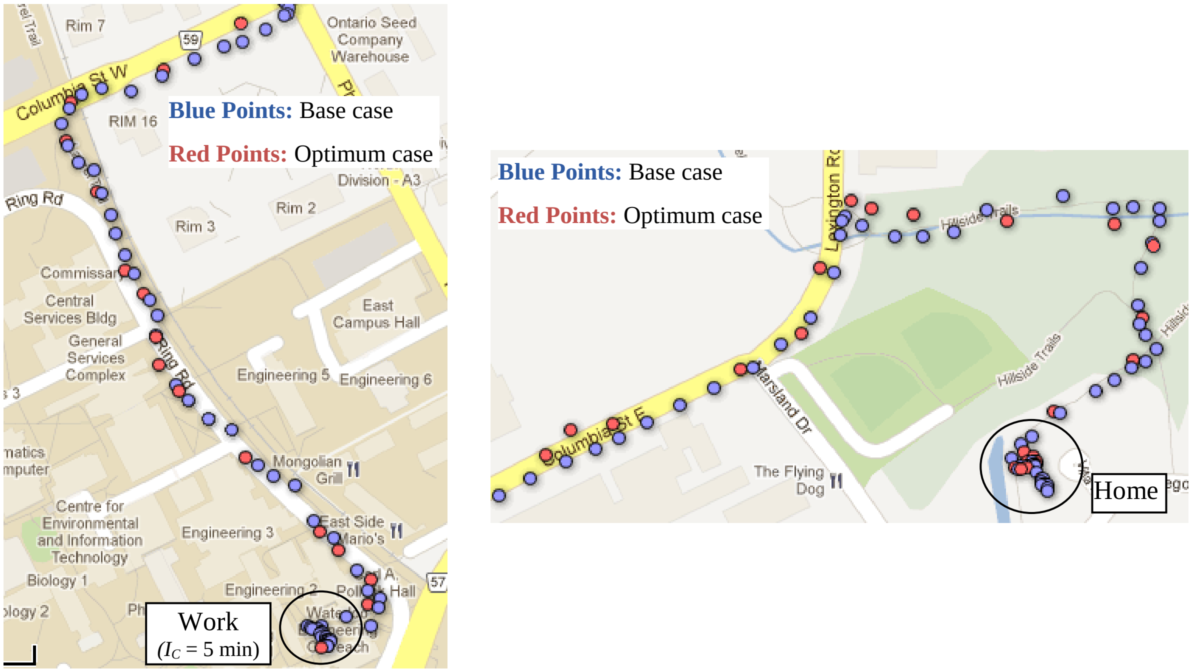


Figure 70: Work to Home trip (Trip 2) - Source: Google map



a) Trip 2 start part
(no data loss in optimum case)

b) Trip 2 end part
(no data loss in optimum case)

Figure 71: Trip 2 close-up (no data loss) - Source: Google map

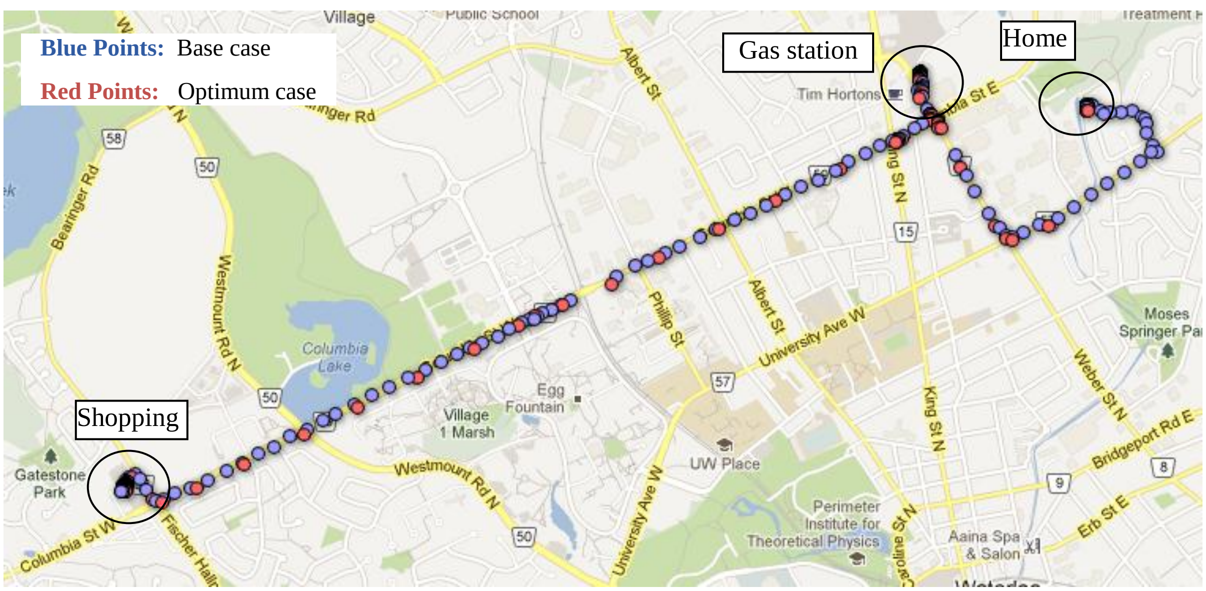
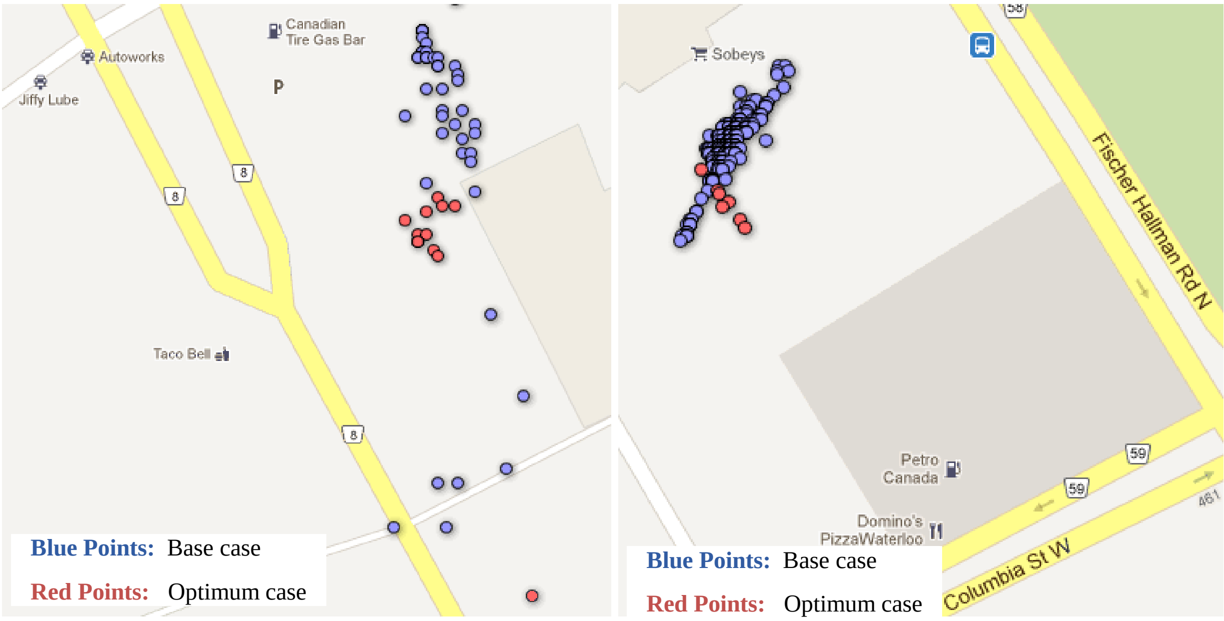


Figure 72: Home to shopping trip (Trip 3) including mid-trip activity stop (Gas Station) - Source: Google map



a) Gas station mid activity stop
(Canadian Tire Gas Bar)

b) Shopping activity stop
(Sobeys)

Figure 73: Identification of activity stop using the land use data - Source: Google map

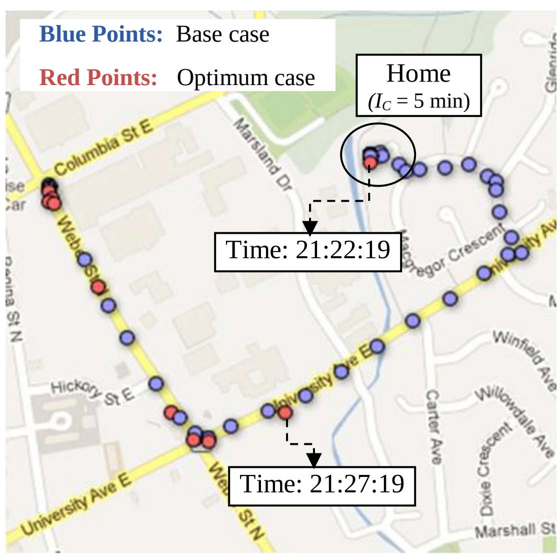


Figure 74: Trip 3 start part (data loss due to SM algorithm) - Source: Google map

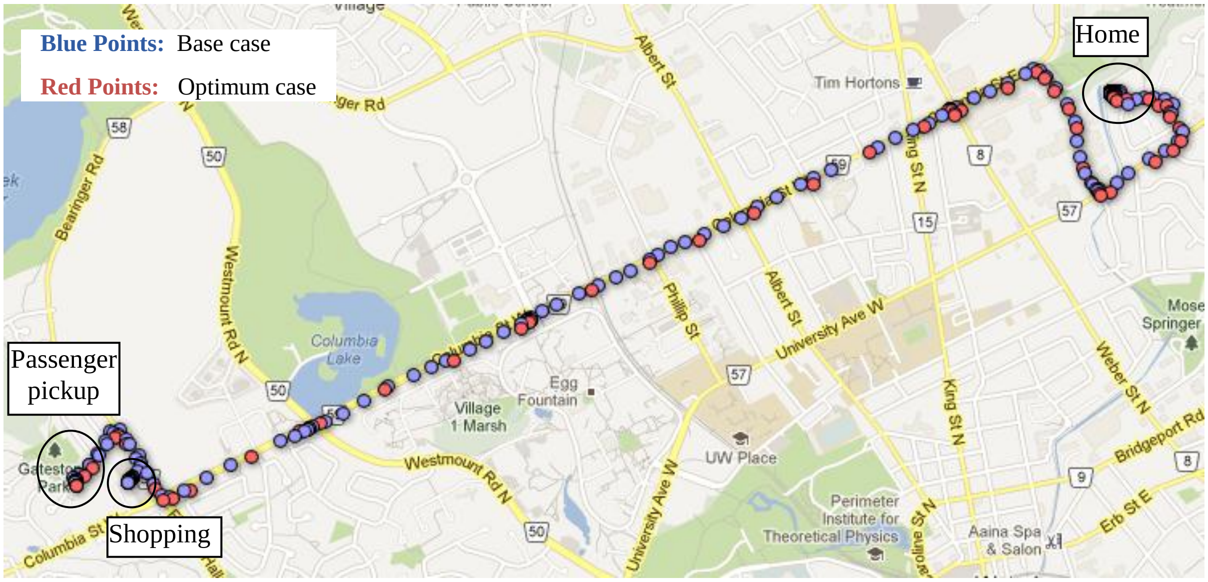


Figure 75: Shopping to home trip (Trip 4) including a mid-trip activity stop (passenger pickup) - Source: Google map

Using the developed travel behavior data collection application in this thesis, we are able to identify the position of a trip maker and the amount of time that the trip maker spent in a location. Although in the current thesis a trip inference model is not developed to identify the trip characteristics such as the trip purpose, using the land use data and the amount of time that a trip maker spent in a location, in many cases it is possible to identify the type of activities that the user's trip consist. As an example, Figure 73 shows a close-up of the activities in Figure 72. We know that the trip maker has spent 3 minutes at the location illustrated in Figure 73a. Based on this amount of time and the land use data presented in Figure 73a (Canadian Tire Gas Bar), it is very probable that this mid activity stop is gas station. Moreover, based on the collected data the trip maker has spent 27 minutes at the location illustrated in Figure 73b. Based on this amount of time and the land use data presented in Figure 73b (Sobeys), it is highly probable that the type of activity at this location is shopping. Therefore, the trip purpose of Trip 3 can be identified as shopping.

One purpose in this experiment is to ensure that after applying the SM and CP algorithms (optimum case) the user travel data is still recorded with an acceptable level of resolution. Figure 69, Figure 70, Figure 72 and Figure 75 show that the user trajectory created by connecting consecutive points in the optimum case (critical points) in general is very similar to the user trajectory created by connecting consecutive points in the base case (all critical and non-critical points). However, to identify the impact of SM and CP algorithms on the recorded data, more detailed investigation is required. The impact of CP algorithm should be analyzed on identifying the critical points both on straight and non-straight paths and the impact of SM algorithm should specifically be analyzed at the initial parts of the trips which are taken after user has been stationary for a while. As explained earlier, in order to reduce the battery consumption rate SM algorithm increases the capture intervals when the user is stationary which might cause data loss in the initial part of the trip when user starts to move.

As an example, since trip maker has spent 2.25 hours in the work place at the end of Trip 1, in the optimum case the SM algorithm has reached to the last state in which the GPS capture interval is 5 minutes (I_c = 5 minutes). Therefore, we need to analyze the impact of this large capture interval (I_c = 5 minutes) on initial part of the next trip (Trip 2). Figure 71a illustrates a close-up of the initial part of Trip 2. According to this figure when the user leaves the work place to take the trip back home, the GPS points (in optimum case) are not missed at the start of the trip due to the large capture interval. In other words, in this case the state machine algorithm has snapped back to the first state (I_c = 5s) before a part of user's travel path is missed.

Moreover, Figure 71a shows that in the optimum case the CP algorithm has identified the critical points throughout the curves and straight parts of the path, and by connecting the critical points the user path can easily be identified. Note that the error created on user trajectory by connecting the critical points has been measured in section 4.2.2.1, and based on an acceptable level of accuracy we calibrated the CP algorithm parameters. Therefore the errors created on user trajectories due to applying the CP algorithm meet acceptable level of accuracy. At the end part of Trip 2 which is illustrated in Figure 71b again it can be observed that by connecting the points in the optimum case (critical points) the user path can easily be identified.

Therefore, Figure 71 demonstrates a good performance of both SM algorithm and the CP algorithm at the start and end part of Trip 2. As for the middle part of Trip 2, which has taken on a straight path, the critical points are recorded at least every 15 seconds (i.e. the calibrated value for time threshold (t_{th})). Moreover, by each speed change between the values lower and higher than stationary speed threshold ($S_{th,s}$) a new critical point is recorded (e.g. at signalized intersections or at congested areas when the vehicle switches between stationary and non-stationary modes). Therefore, though the CP algorithm records only a subset of all the captured points, this level of resolution should still be sufficient for identifying travel characteristics such as congestion on road networks or measuring delays in signalized intersections or waiting times in bus stops.

Although in Figure 71a we illustrated that the large capture interval (I_c = 5 minutes) at the activity location (work place) did not affect the initial part of Trip 2, this case is not always true. As an example, Trip 3 starts after the trip maker spends 4 hours at home (SM reaches to I_c = 5 minutes). Figure 74 shows the close-up of initial part of Trip 3. According to the recorded data, it exactly has taken 5 minutes between the last captured point at home and the first captured point in Trip 3 in the optimum case. Therefore, the first few minutes of Trip 3 are missed in the optimum case. This condition could be worsened when by capturing the first point at non-stationary mode the SM does not return back to the first state to capture the GPS points frequently (every 5 seconds). This case might happen when capturing the first point is not successful (the point cannot be obtained within the predefined timeout (C_{TO}) period due to cold GPS), or when the speed of the first captured point at non stationary mode is reported as a value less than $S_{th,s}$. This (speed < $S_{th,s}$) can happen due to inaccuracy of the device or that the user speed is low at that specific moment (e.g. trip maker stopped at a traffic signal or stop sign). In this case state machine algorithm does not return back to the first state to capture the points every 5 seconds when the trip maker has started to move. Therefore, the next point will still be captured with the capture interval of 5 minutes which might result in missing a major part of a short trip.

According to the discussed limitations of the SM algorithm in recording the first part of a trip, the developed algorithm (SM) might not be suitable for recording short trips or for measuring the travel behavior characteristics such as access time (e.g. the time to get into bus stop) if the trip maker has been at stationary mode for a while. As another example, the first part of a transit trip might be missed due to the waiting time at the bus station (at which SM algorithm increases the capture interval). The SM algorithm might not provide accurate results in measuring travel behavior characteristics such as delays in signalized intersections or in congested areas since it increases the capture intervals in these conditions. In general, the limitation of SM algorithm in collecting travel behavior data is minimum for longer trips. This algorithm is mostly effective when the person is stationary and GPS signals are available (outdoors or indoors). As an example, in the discussed daily trips in this experiment, as well as the data collection outdoors, the device was able to capture the GPS points at work place and home since the GPS signals have been available to the device at those locations. The following is a discussion on the amount of battery saved throughout the daily trips recorded for this experiment, which clearly shows the benefit of the SM algorithm in saving the battery at stationary modes.

In this experiment, the whole duration of the recorded four trips as well as the included activities is 9.25 hours. During this time, the device in the base case has consumed 86% of the battery life, and the device in the optimum case has consumed 23% of the battery life. If we exclude the home activity spent between Trip 2 and Trip 3 (which took 4 hours) and the home activity at the end of Trip 4 (which took 1.7 hours) then the battery consumption by the device in the base case is computed as 24% for the first two trips (Trip 1 plus Trip 2) and 8% for the last two trips (Trip 3 plus Trip 4). On the other hand, in this case the battery consumption by the device operating under the optimum case is computed as 11% for the first two trips (Trip 1 plus Trip 2) and 5% for the last two trips (Trip 3 plus Trip 4). In fact, in total the home activities (5.7 hours) consumed 54% of battery life in base case and 7% of battery life in optimum case (benefit of SM algorithm). The battery consumption results in this experiment show that in the optimum case the battery consumption has been improved significantly.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

The following provides a summary of the conclusions drawn from this study:

1. Collecting individual travel behavior data through the use of GPS enabled smart phones is technically feasible and would address most of the limitations associated with using other survey techniques.
2. Evaluation of the accuracy of the position information acquired from the developed travel data collection application for smart phones has shown that the Blackberry Bold 9700 provided a level of accuracy of speed and position data that is comparable to the same data obtained via GPS loggers. The position data obtained via the Blackberry Torch 9810 also showed comparable accuracy. However, the speed data collected by Torch 9810 was less accurate at low speeds (i.e. speeds associated with walk mode)
3. The assessment of the data acquisition application has shown that Assisted and Autonomous GPS modes report accurate position data. Acquiring locations using the cell site mode does not provide acceptable levels of accuracy. Furthermore, the examination of the battery consumption by the three GPS modes has indicated that cell site method has the lowest power consumption rate of 10% per hour, and the rate of power consumption for the Assisted and Autonomous modes are 2.5% per hour and 6% per hour higher than cell site respectively. Therefore, power consumption management in travel data collection by smart phones remains a significant challenge.
4. Several techniques have been proposed in this study to help reduce power consumption by the travel data acquisition application. The results of the study have shown that the proposed techniques reduce the battery consumption rate from 13.3% per hour to 5.75% per hour (i.e. 57% reduction) when the trip maker is non-stationary. The battery consumption rate is also reduced from 5.75% per hour to 1.41% per hour (i.e. 75.5% reduction) when the trip maker is stationary (SM algorithm effect).
5. Moreover, the quantitative evaluation of the proposed CP algorithm has shown that the performance of this algorithm is optimized in terms of the battery performance and

accuracy of the recorded trajectory when the angle threshold (Δ_{th}) is 15 degrees and time threshold (t_{th}) is 15 seconds. Analysis of the time threshold (t_{th}) parameter also has indicated that battery consumption rate decreases with the logarithm of this parameter. Therefore, values larger than 15 seconds might also be selected for the time threshold depending on the desired accuracy and the required amount of savings in network transmissions and data storage space. The optimized CP algorithm (i.e. Δ_{th} = 15 degrees, t_{th} = 15s) has shown to decrease the number of points transmitted to the server by 57% on average. Therefore, this algorithm is an effective solution for reducing network transmissions and data storage space while still meeting a sufficient level of accuracy in terms of travel data collection. According to the results of battery analysis for this algorithm, the battery consumption rate exhibits a much higher than expected variability. The source of this variability has been investigated and several potential sources have been ruled out as causing the variability. The accuracy of the reported battery level has remained as a possible source.

6. Examination of the SM algorithm has indicated that the speed value of 0.75 m/s is a reasonable value for stationary speed threshold ($S_{th,s}$) which is used to distinguish between stationary and non-stationary modes. Furthermore, it has been identified that due to GPS errors, in some cases, the reported speed by Blackberry device might exceed the $S_{th,s}$ even though the trip maker is stationary. Therefore, an error handling algorithm has been developed to identify the errors in these cases and as a result to increase the performance of SM algorithm.
7. Evaluation of the impact of transmission interval (I_t) on battery consumption rate has shown that the optimal value for this parameter is 15 minutes which minimizes the battery consumption rate. This optimal value decreases the battery consumption rate from 13.3% per hour to 6.3% per hour when no other optimization algorithm is running. Therefore, the majority of the battery saving at non-stationary mode explained in conclusion #4 is related to using the optimal value for transmission interval (I_t).
8. It should be noted that different smart phones have different hardware configurations and therefore have different baseline power consumption levels and different sized batteries. Therefore, the absolute magnitude of the battery consumption rates reported in this study may not be directly applicable to other hardware platforms.

9. Analysis of the impact of SM algorithm on travel behavior data collection has indicated that SM algorithm is mostly effective in saving the battery at stationary modes when the GPS signals are available. The limitation of this algorithm is in collecting the initial part of trips which are recorded after the trip maker has been stationary for a while. For measuring the travel characteristics which require high level of accuracy (e.g. delays at intersections, access times) this algorithm might not provide sufficient accuracy. However, it is likely that the proposed system is suitable for obtaining travel data for most trips.
10. Analysis of the impact of CP algorithm on travel behavior data collection has indicated that the level of resolution provided by this algorithm should be sufficient for identifying travel characteristics such as congestion on road networks or measuring delays in signalized intersections or waiting times in bus stations.

5.2 Recommendation

The following provides a list of recommendations which might be considered for further study in order to provide a more efficient travel data acquisition application for smart phones.

1. The proposed CP algorithm should be extended to consider a distance threshold in addition to the existing time threshold as a means of providing a more efficient algorithm.
2. The proposed SM algorithm should be extended to improve the response time in identifying when the user begins to move. In the current version, the response time can be as long as the capture interval (i.e. 5 minutes for the last state). This version may result in the failure to capture the initial portion of a trip if the trip maker has been stationary for a relatively long time and the SM is in state 6. An enhanced method would reduce the response time for identifying that the trip maker was no longer stationary and the capture interval could be reduced, reducing the amount of travel data that would be missed.
3. Given that the major battery savings at non-stationary mode is related to using the transmission interval (I_t) at its optimal level (every 15 minutes) and the most accurate data is captured when the SM algorithm and CP algorithm are off, it is recommended to only apply the transmission interval method (I_t = 15 minutes) as battery optimization

method when travel data collection requires high accuracy (i.e the accuracy at operation level). Nonetheless, as explained in conclusion #10, the CP algorithm still provides sufficient accuracy for identifying many travel characteristics.

4. Since different smart phone devices appears to report speeds with different levels of accuracy, the value of the stationary speed threshold ($S_{th,s}$) should be dependent on the device rather than a constant for all devices.
5. Further investigation should be carried out on newer Blackberry devices and technologies to seek opportunities to make additional improvements to the power consumption management and GPS data accuracy. These investigations should focus on complementary GPS modes and Blackberry transport techniques (section 3.2).
6. Since the Direct TCP is used for transporting the data to server side and this technique utilizes the service provider gateway, the data collection should be examined for different service providers[1].
7. Though the implemented system architecture has been designed to accommodate a large number of handsets, in the current research the system has been tested only for a small number of handsets. Therefore, the travel behavior data collection application should be tested for a large number of users, simultaneously.

[1] Currently, the data collection application is tested for Rogers service provider.

References

Aguilar, D.P. 2008. A framework for evaluating the computational aspects of mobile phones, Ph.D. dissertation, university of South Florida.

Assisted GPS, *Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/wiki/Assisted-GPS.

Ballantyne, W., G. Turetzky, G. Slimak, and J. Shewfelt. 2005. Achieving low energy-per-fix in cell phones. *Proceedings of the 18th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2005)*, Long Beach, CA: 2234-2242.

Barbeau, S., M. A. Labrador, A. Perez, P. Winters, N. Georggi, D. Aguilar, and R. Perez. 2008. Dynamic management of real-time location data on GPS-enabled mobile phones. UBICOMM'08 2008. The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies.

BlackBerry Java Application. Location-based Services. 2010. v5.0. Research In Motion.

BlackBerry Java Development Environment. GPS & BlackBerry Maps. 2009. v4.7.0. Canada: Research In Motion.

BlackBerry java SDK. Network communication. 2010. v6.0. Research In Motion.

Casas, J., and CH Arce. 1999. Trip reporting in household travel diaries: A comparison to GPS-collected data. Paper presented at 78th Annual Meeting of the Transportation Research Board, Washington, DC.

Casello, J. M., A. Nour, K. C. Rewa, and J. Hill. 2011. An Analysis of Stated Preference and GPS Data for Bicycle Travel Forecasting. Paper presented at 78th Annual Meeting of the Transportation Research Board, Washington, DC.

Chung, E. H., and A. Shalaby. 2005. A trip reconstruction tool for GPS-based personal travel surveys. *Transportation Planning and Technology* 28 (5): 381-401.

Doherty, S. T., N. Noël, M. L. Gosselin, C. Sirois, and M. Ueno. 2001. Moving beyond observed outcomes: Integrating global positioning systems and interactive computer-based travel behavior surveys. (E-C026): 449-466.

ECEF, *Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/wiki/ECEF.

Fabris, Andre. 2009. *A14 Network Transports*. *For Blackberry Smartphones*. Research In Motion.

Gonzalez, P., J. Weinstein, S. Barbeau, M. Labrador, P. Winters, N. Georggi, and R. Perez. 2008. Automating Mode Detection Using Neural Networks and Assisted GPS Data Collected

Using GPS-enabled Mobile Phones. 15th World Congress on Intelligent Transportation Systems, New York.

Leonhardi, A., C. Nicu, and K. Rothermel. 2000. A map-based dead reckoning protocol for updating location information. Paper presented at Proceedings of the 16th IEEE International Parallel and Distributed Processing Symposium.

Leonhardi, A., and K. Rothermel. 2002. Protocols for updating highly accurate location information. *Geographic Location in the Internet*: 111-141.

Moeglein, M., and N. Krasner. 1998. An Introduction to SnapTrack Server-Aided GPS Technology. *Institute of Navigation GPS Meeting Proceedings*: 333–342. http://systems.cs.colorado.edu/grunwald/MobileComputing/Papers/snaptrack-overview.pdf.

Murakami, E., and D. P. Wagner. 1999. Can using global positioning system (GPS) improve trip reporting? *Transportation Research Part C: Emerging Technologies* 7 (2): 149-165.

Murakami, E., D. P. Wagner, and D. M. Neumeister. 1997. Using global positioning systems and personal digital assistants for personal travel surveys in the united states. Paper presented at International Conference on Transport Survey Quality and Innovation, Grainau, Germany.

NavCen GPS user. 1996. 3.5.3 Almanac Collection.

Ortuzar, J. de D. and L. G. Willumsen 1994. Modeling Transport, New York: John Wiley and Sons.

Reddy, S., M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava. 2010. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)* 6 (2): 13.

S. Microsystems. 2008. *Java specification request (jsr) 179*. *Location API for j2me*. [Online]. Available: http://jcp.org/en/jsr/detail?id=179.

Stopher, P., and A. Collins. 2005. Conducting a GPS prompted recall survey over the internet. Paper presented at Transportation Research Board Annual Meeting, 84th, 2005, Washington, DC, USA.

Taylor, I. M. and M. A. Labrador. 2011. Improving the energy consumption in mobile phones by filtering noisy GPS fixes with modified Kalman filters. 2011 IEEE Wireless Communications and Networking Conference (WCNC), IEEE Press: 2006-2011.

The Astronomical Almanac. 2007. Appendix K. United States Naval Observatory and Her Majesty's Nautical Almanac Office.

Treu, G., A. Küpper, and T. Wilder. 2008. Extending the LBS-framework TraX: Efficient proximity detection with dead reckoning. *Computer Communications* 31 (5): 1040-51.

Theodore S. Rapporteur 2002. *Wireless Communications, Principles and Practice*. Prentice-Hall, Inc

Weisstein, Eric W. 2012. *Point-Line Distance--3-Dimensional.* From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/Point-LineDistance3-Dimensional.html.

Wolf, J., R. Guensler, and W. Bachman. 2001. Elimination of the travel diary: Experiment to derive trip purpose from global positioning system travel data. *Transportation Research Record: Journal of the Transportation Research Board* 1768 : 125-34.

Wolfson, O., A. P. Sistla, S. Chamberlain, and Y. Yesha. 1999. Updating and querying databases that track mobile units. *Distributed and Parallel Databases* 7 (3): 257-387.

Appendices

Appendix A
Device Application User Manual

This section provides a user manual which describes the steps through which the implemented application on device side, “TravelSurvey”, could be used:

1. “TravelSurvey” app can be accessed through Downloads screen of Blackberry device (Figure 1a).
2. Then user is entered into a user login page (Figure 1b) which includes User ID, Password text boxes and Go button
3. At this screen, user needs to enter the User ID and Password that has registered previously, and then click “Go” button.
4. The account information is then sent to server for confirmation.
5. If user information is correct user will be entered into the main screen of the app and will receive a confirmation message (Figure 1c). Then, user information will be saved into the device so user will not need to enter it for the next times to run the app.
6. If User ID and Password is not correct or user information is not valid any more, user will get an error message (Figure 1d). In this case, user can enter the information again.
7. At login screen, user can close the app using close option through the device menu.
8. In the main screen of the application, user needs to select the initial travel mode before starting the application (Figure 1e).
9. Additionally, in the main screen, “Speed Simulator” could be set to the desired speed value (Figure 1e). This is the feature implemented for system administrator and the application of this feature is to set the speed to a value below or above stationary speed threshold ($S_{T,S}$), in order to examine the performance of SM algorithm without a need to move the device. To set a speed value “Speed Simulator” checkbox should be clicked on, the speed value should be entered and the “Set Speed” button should be clicked. To turn off the “Speed Simulator” and use the real speed reported by GPS, “Speed Simulator” checkbox should be clicked off. Note: at any time during the application run, a new speed value could be set.

10. Acquisition of GPS data can be started by selecting "Start Capturing" option through the device menu (Figure 1f).
11. Through the same menu, user can select "Options" to change the settings of the application before starting data capturing (Figure 1f).
12. When "Options" is selected user is entered into "Options" screen through which different settings of the application can be adjusted. For example, minimum and maximum capture interval, transmission interval, the information to be displayed on the main screen, the status of SM algorithm and CP algorithm (on/off) and etc (Figure 1g). This screen is mainly implemented for system administrator to analyze the performance of different algorithms within the application.
13. Once the settings are changed, user can save the settings through the "save" option in the menu (Figure 1h).
14. At this stage, user can start data capturing as explained in step 10. Once the data capturing is started, user does not anymore have access to the "Options" screen.
15. If user has not selected any travel mode before starting the data capturing, an error message will be displayed on the screen to enter the initial travel mode (Figure 1i). Note: at any time when the travel mode is changed throughout the trip, user can select the new travel mode through the main screen.
16. When data capturing is started, all captured information will be displayed on the main screen. Additionally, after each transmission of the data to server, a confirmation message as well as the used TripRunId will be displayed on the screen (Figure 1j).
17. At any time during the application run, user can close the application by selecting the "Close" option through device menu (Figure 1k).


BlackBerry
20:55
TravelSurvey
1xev
Downloads
TravelSurvey

a)


BlackBerry
Please enter your UserID and Password:
UserID
test
Password
****
Go

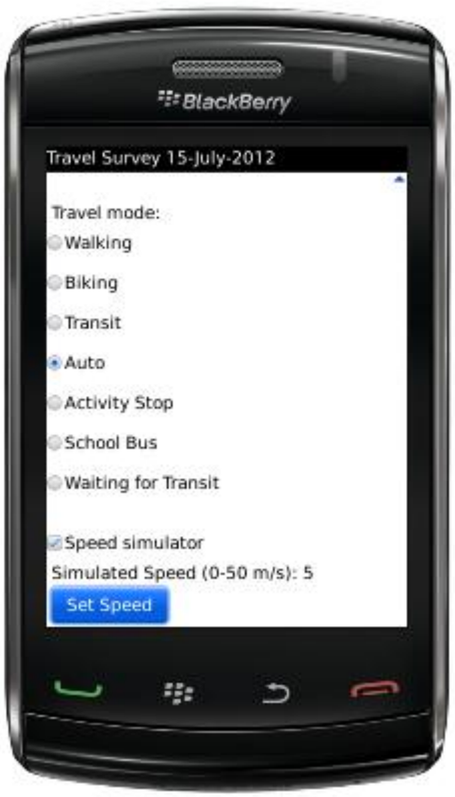
b)


BlackBerry
Travel Survey 19-April-2012
Capture Interval:
Current Time:
2012/04/27 20:57:53
UserID and Password OK
OK
Auto
Activity Stop
School Bus

c)


BlackBerry
Please enter your UserID and Password:
UserID
Test
Password
***
Go
Incorrect UserID or Password
OK

d)


BlackBerry
Travel Survey 15-July-2012
Travel mode:
Walking
Biking
Transit
Auto
Activity Stop
School Bus
Waiting for Transit
Speed simulator
Simulated Speed (0-50 m/s): 5
Set Speed

e)


BlackBerry
Travel Survey 19-April-2012
Capture Interval:
Current Time:
2012/04/27 20:59:51
Travel mode:
Walking
Options
Start capturing
Show Keyboard
Switch Application
Close

f)


BlackBerry
Options
Min Capture Interval(sec) 5
Max Capture Interval(sec) 300
Sending Interval(sec) 60
TimeToFirstFix(sec) 180
Speed/acce/dist
Satellite info
Azimuth
Battery usage
Accelerometer
Fix Capture Interval
GPSMode
Assisted
Autonomous

g)


BlackBerry
Options
Min Capture Interval(sec) 5
Max Capture Interval(sec) 300
Sending Interval(sec) 60
Select
Check Spelling
Save
Clear Field
Enable Multitap
Enable Full Keyboard
Show Keyboard
Switch Application
Close

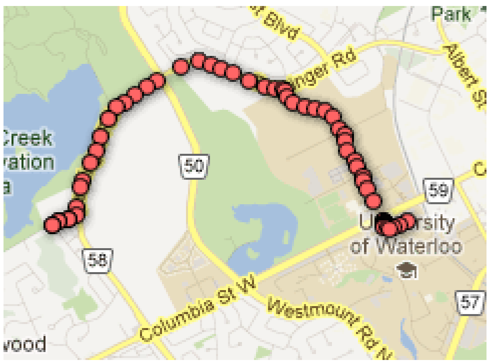
h)

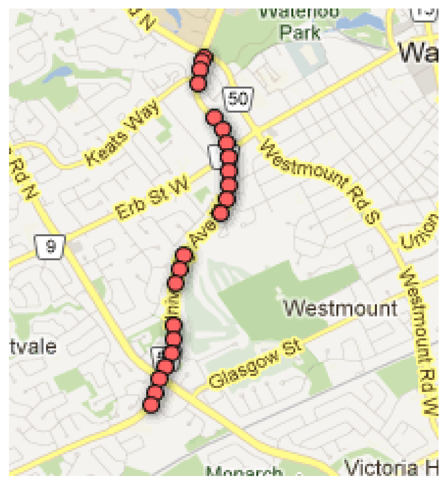
Figure 1: Snapshots of "TravelSurvey" application

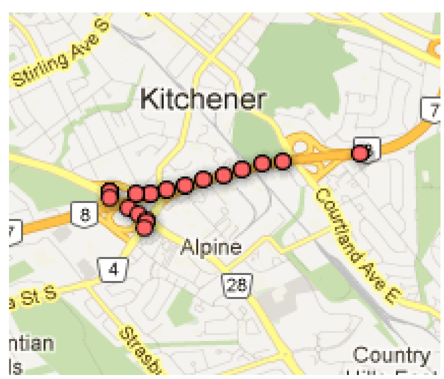
Appendix B

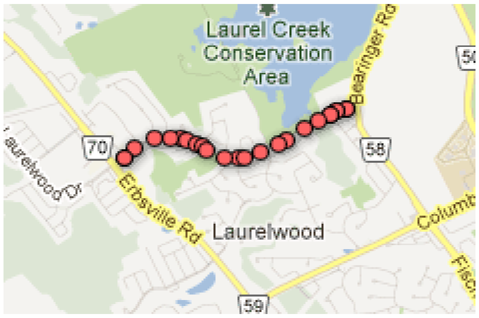
Sample Trajectories

This section illustrates the trajectories used in analysis of angle threshold (t_{th}) in section 3.3.2.4.

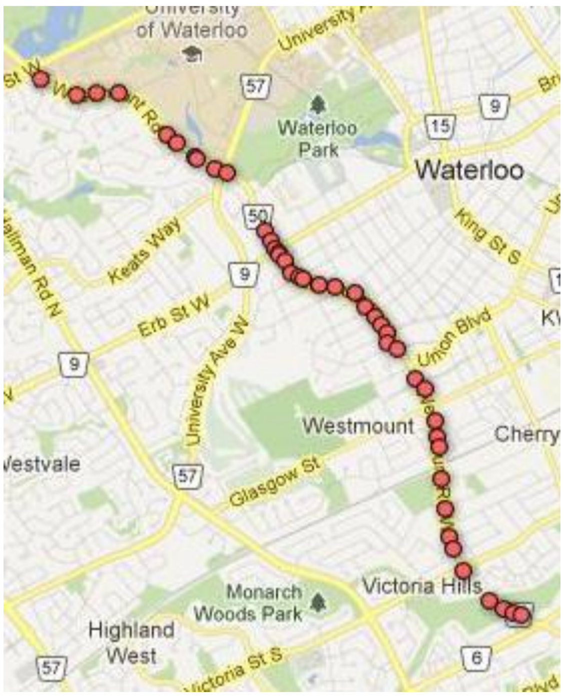


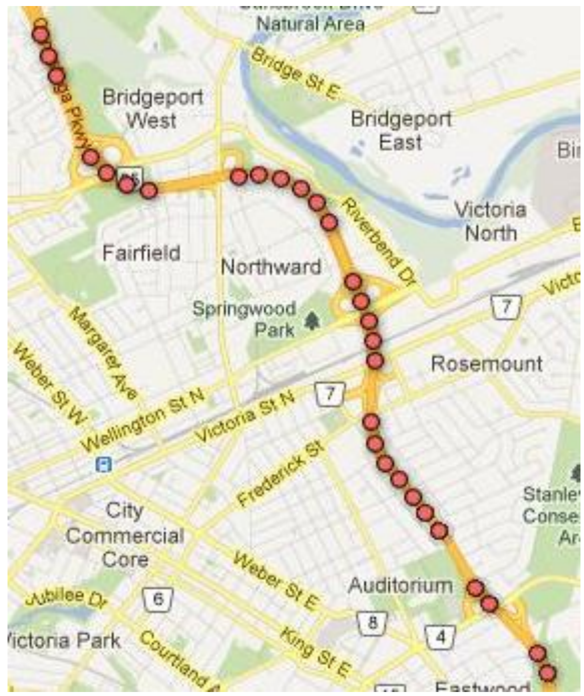


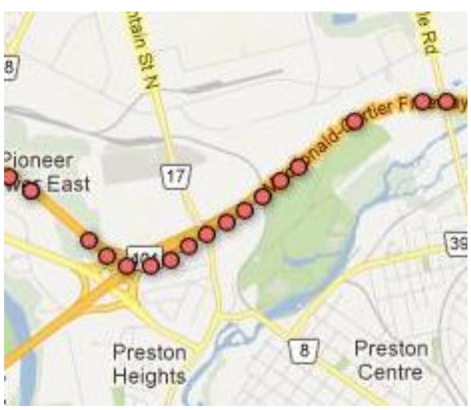



Laurel Creek
Conservation
Area
Bearinger Rd
70
58
59
Erbsville Rd
Laurelwood


Laurel Creek
Conservation
Area
Bearinger Rd
70
58
Erbsville Rd
Laurelwood

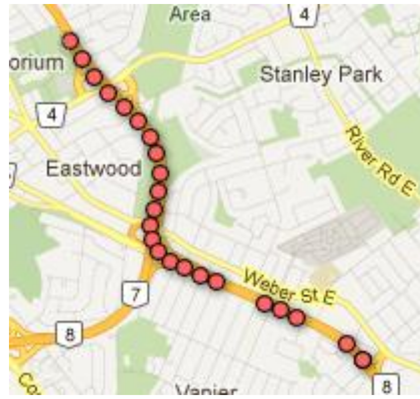

of Waterloo
57
9
15
Waterloo
Waterloo
Park
King St S
50
Keats Way
Erb St W
9
Union Blvd
University Ave W
Westmount
Cherry
57
Glasgow St
Monarch
Woods Park
Victoria Hills
Highland
West
6
57


Natural Area
Bridge St E
Bridgeport
West
Bridgeport
East
Victoria
North
Riverbend Dr
Fairfield
Northward
Springwood
Park
7
Rosemount
Margaret Ave
Weber St W
Wellington St N
Victoria St N
7
Frederick St
City
Commercial
Core
Weber St E
Auditorium
6
8
4
King St E
Courtland

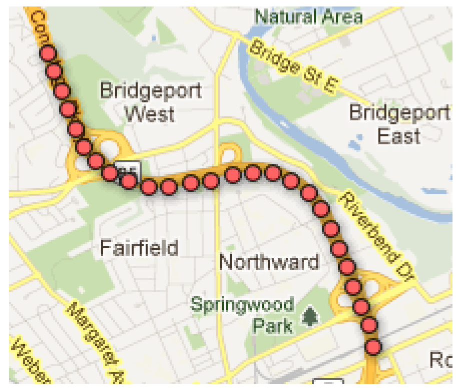

Pioneer
East
17
39
8
Preston
Heights
Preston
Centre


Dixon Rd
Toronto Pearson
International
Airport
Renforth Dr
Convair Dr


Courtland Ave E
15
East
Kitchener
8
7
Alpine
4
28
Ottawa St S
Courtland Ave


Area
4
Stanley Park
River Rd E
Eastwood
7
8
Weber St E


Cherry Hill
Park St
Jubilee Dr
Victoria St S
Victoria Park
Highland Rd W
6
58
Queen's Blvd
Meinzinger


Natural Area
Bridge St E
Bridgeport West
Bridgeport East
Fairfield
Northward
Riverbend Dr
Springwood Park
Margaret Av


25
No. 5 Sideroad
Escarpment Way
4
25
8
4
Maple Ave
Steeles Ave E
Woodward Ave
Ontario St N
Lions Sports Park
25

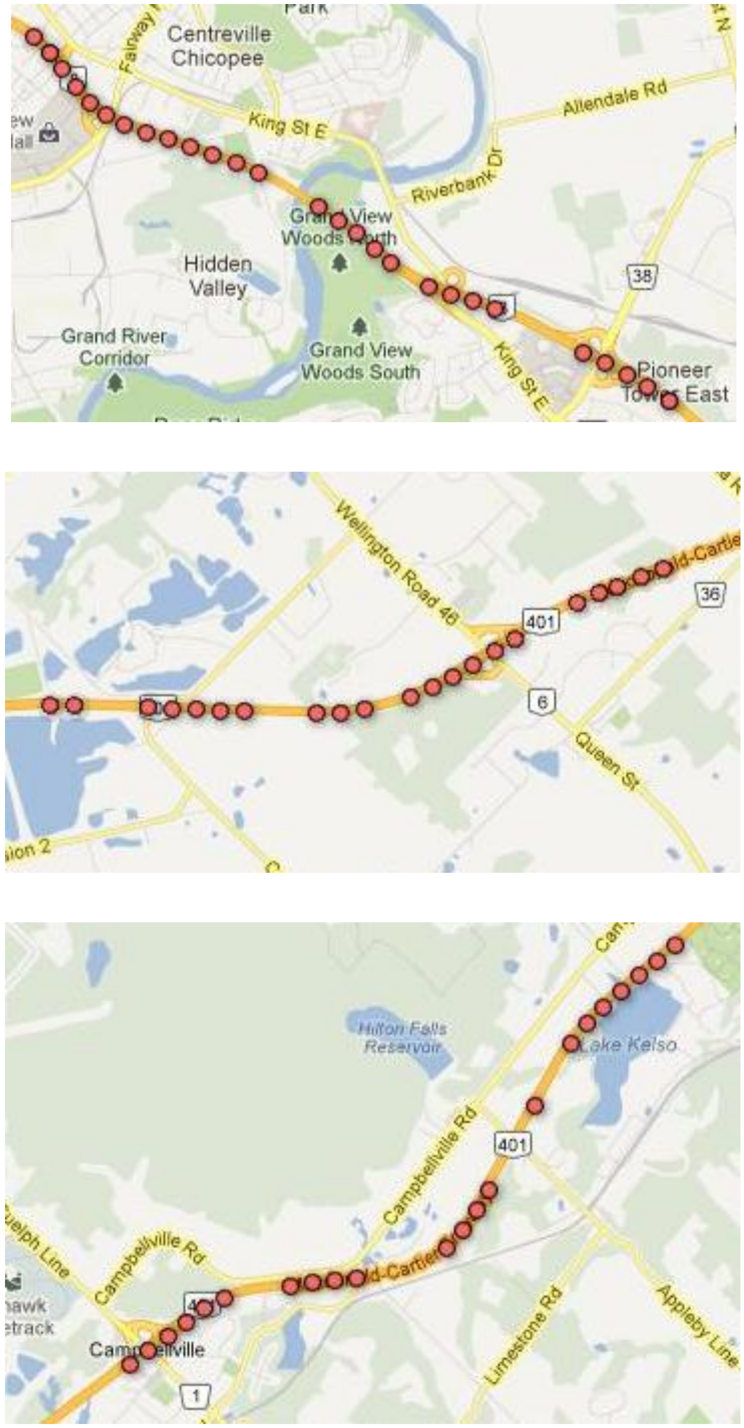
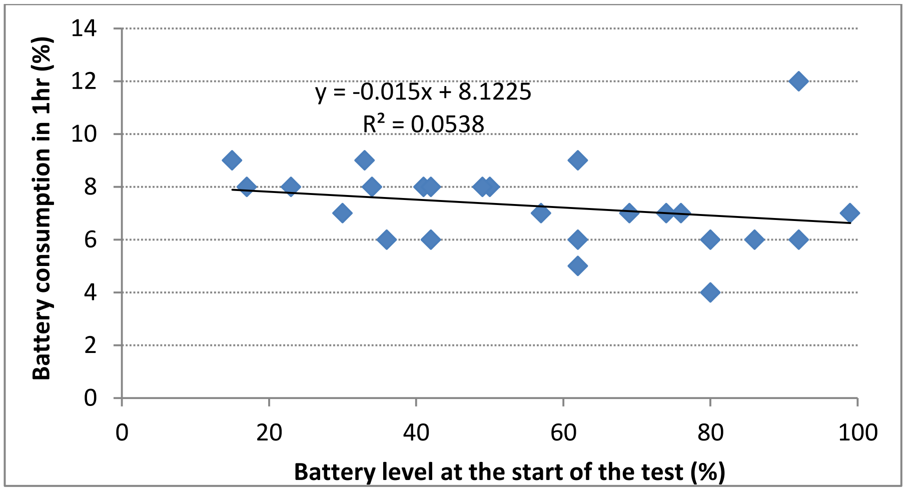


Figure 1: Trajectory segments used for calibration of time-threshold parameter in CP algorithm (Source: Google map)

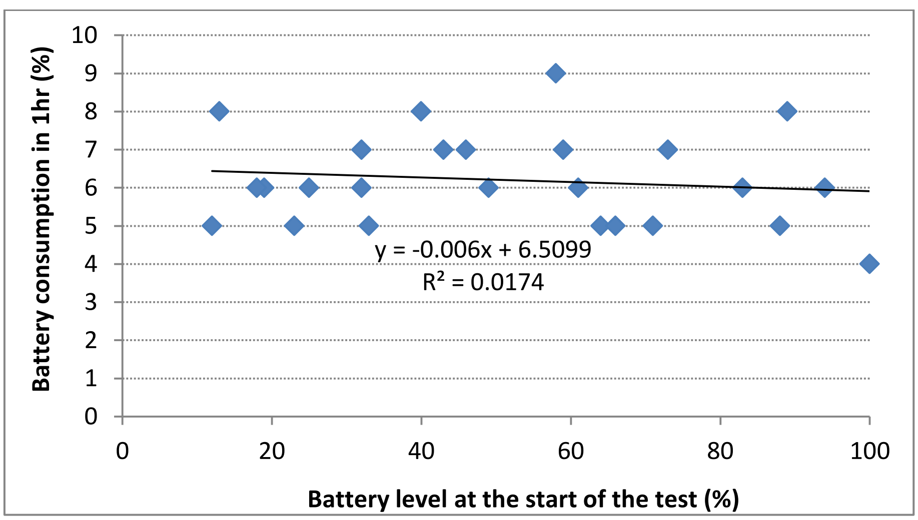
Appendix C

Regression Analysis of battery consumption against initial battery level



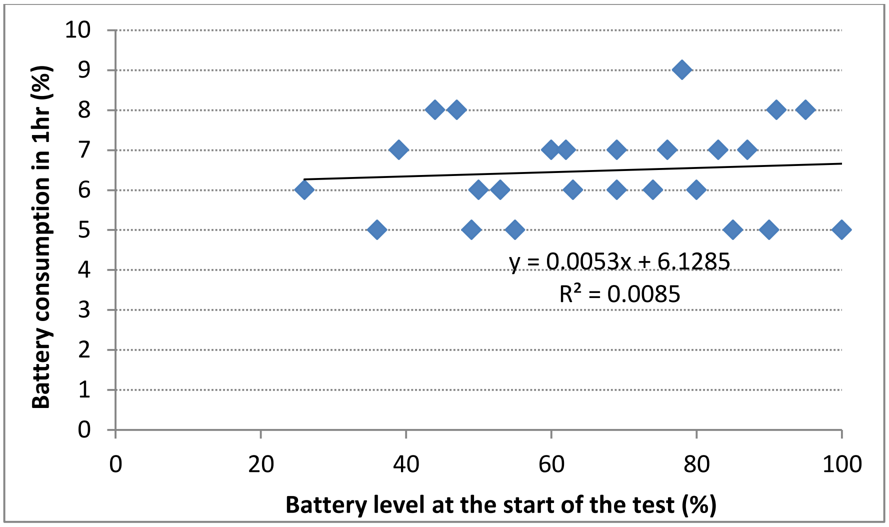
Variable	Coefficient Value	T-Statistic
Initial battery level	-0.02	-1.14
Intercept	8.12	10.13
No. of Obs. = 25	Corrected R^2= 0.01	

a. **t_{th} = 5s**



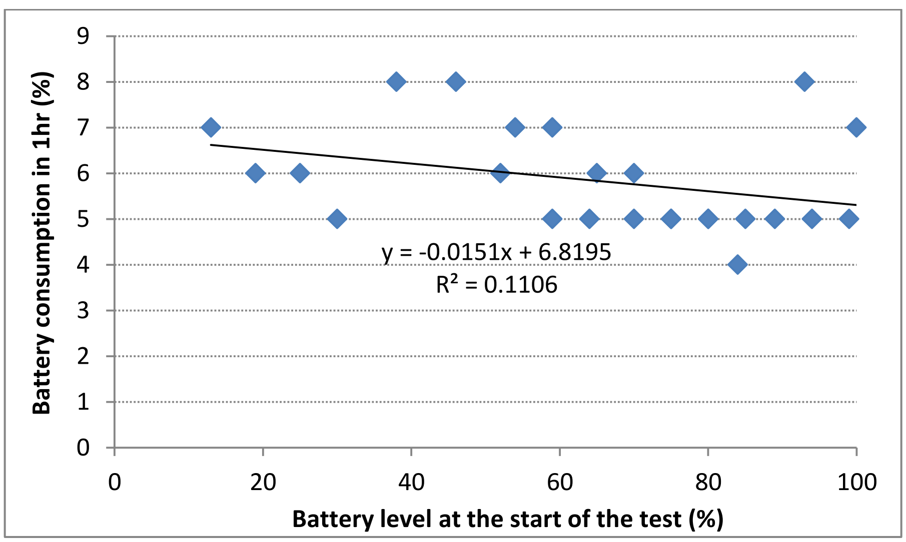
Variable	Coefficient Value	T-Statistic
Initial battery level	-0.01	-0.64
Intercept	6.51	11.93
No. of Obs. = 25	Corrected R^2= -0.03	

b. **t_{th} = 15s**



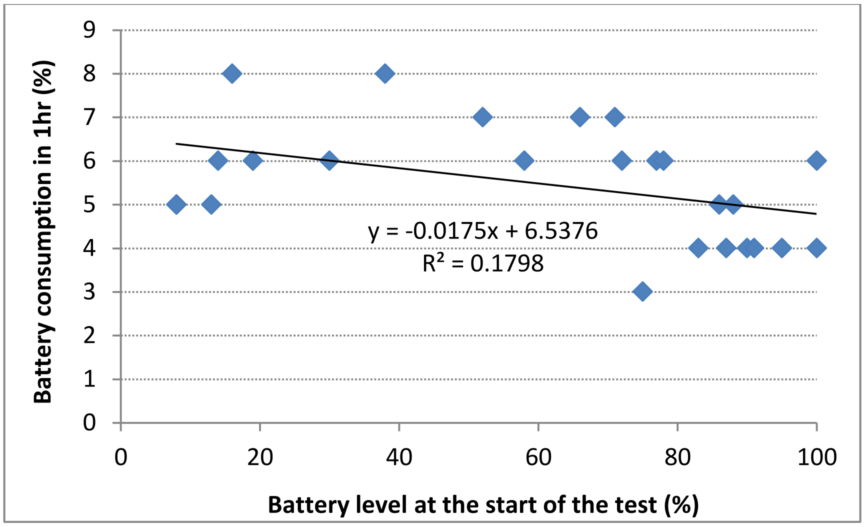
Variable	Coefficient Value	T-Statistic
Initial battery level	0.01	0.44
Intercept	6.13	7.41
No. of Obs. = 25	Corrected R^2= -0.03	

c. **t_{th} = 30s**



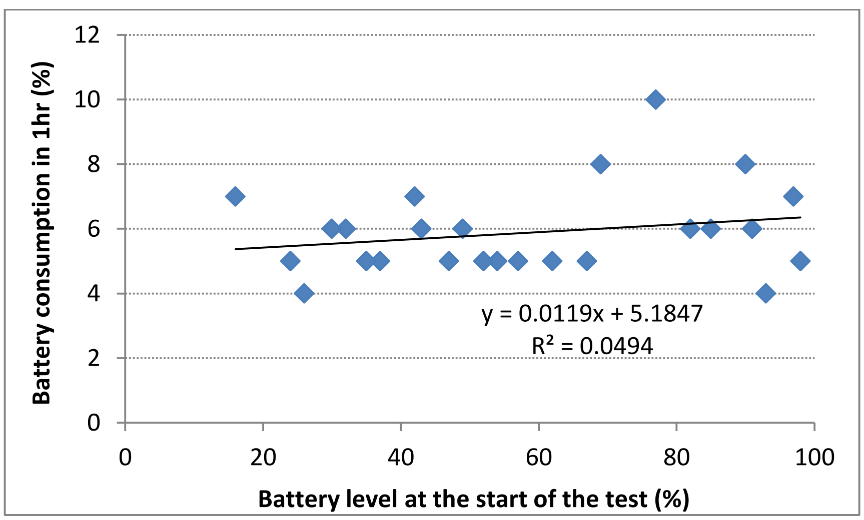
Variable	Coefficient Value	T-Statistic
Initial battery level	-0.02	-1.69
Intercept	6.82	11.00
No. of Obs. = 25	Corrected R^2= 0.07	

d. **t_{th} = 60s**



Variable	Coefficient Value	T-Statistic
Initial battery level	-0.02	-2.25
Intercept	6.54	12.31
No. of Obs. = 25	Corrected R^2= 0.14	

e. $\boldsymbol{t_{th}}$ **= 120s**



Variable	Coefficient Value	T-Statistic
Initial battery level	0.01	1.09
Intercept	5.18	7.50
No. of Obs. = 25	Corrected R^2= 0.01	

f. $\boldsymbol{t_{th}}$ **= 300s**

Figure 1: Regression analysis of battery consumption against initial battery level at individual levels of time threshold (t_{th})

Appendix D

Statistical Analysis

Calculation of Minimum Number of run

$$n' = \left(\frac{t_{\alpha/2,\,n-1} \times S}{d}\right)^2 \qquad \textbf{Equation 1}$$

Where:

n = *Initial number of runs*

d = *Maximum acceptable error*

$1-\alpha$ = Confidence level

n - 1 = Degree of freedom

n' = Minimum number of *runs*

S = *Standard deviation*

And:

Additional run = n' - n

Mean Comparison

$H_0 : \mu_1 = \mu_2$

$H_1 : \mu_1 > \mu$[1]

$\bar{x}_1 \sim N(\mu_1, s_1^2)$

$\bar{x}_2 \sim N(\mu_2, s_2^2)$

H_0 : Null hypotheses

H_1 : Alternative hypotheses

n_1: Number of observation for sample1

n_2: Number of observation for sample2

$\bar{x}_1$: Average for sample1

$\bar{x}_2$ **:** Average for sample2

μ_1 : Mean for population of sample1

μ_2 : Mean for population of sample2

s_1^2: Variance for population of sample1

s_2^2: Variance for population of sample2

If the result of Equation 2 is true, then H_0 is rejected and H_1 is accepted, otherwise H_0 is accepted.

[1] When we are just examining if the two means are different or not, then alternative hypothesis would be $H_1 : \mu_1 \neq \mu_2$ and two-sided test will be used ($\alpha/2$).

$$\left| t_o = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^{\ 2}}{n_1} + \frac{s_2^{\ 2}}{n_2}}} \right| > t_{\alpha,\nu} \qquad \textbf{Equation 2}$$

Where,

$$\boldsymbol{\nu} = \frac{\left(\frac{s_1^{\ 2}}{n_1} + \frac{s_2^{\ 2}}{n_2}\right)^2}{\left[\left(\frac{s_1^{\ 2}}{n_1}\right)^2 \left(\frac{1}{n_1 - 1}\right)\right] + \left[\left(\frac{s_2^{\ 2}}{n_2}\right)^2 \left(\frac{1}{n_2 - 1}\right)\right]} \qquad \textbf{Equation 3}$$

And,

t_o : Observed t

$t_{\alpha,\nu}$: t student value

$1-\alpha$: Level of confidence

ν : Degree of freedom