TAG - TRIP-BOXES FOR AUTOMATIC EVENT GENERATION

BY JANANI CHANDRASEKARAN

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Marco Gruteser

and approved by

New Brunswick, New Jersey

May, 2010

ABSTRACT OF THE THESIS

TAG - Trip-Boxes for Automatic event Generation

by Janani Chandrasekaran

Thesis Director: Marco Gruteser

The increased use of location-aware mobile devices has given rise to a need to trigger actions at certain geographic locations. For example, when conducting outdoor vehicular experiments, data frequently needs to be collected within one geographic region. The data collection can be initiated manually, however, it is a tedious process and prone to manual errors. Our framework is designed to use Global Positioning System location updates to automatically trigger actions upon entering and leaving these geographic zones. We mark these zones using trip-boxes, a rectangular box enclosing a geographical area with its orientation towards geographic north and east direction, which can be defined in Google Earth and exported into the trip-box system. The system also uses guard intervals to prevent repeated triggering of trip-boxes due to GPS oscillations.

In our evaluation, we experimentally determine the chances of skipping the Trip-Box for a given speed, sampling rate of GPS and the length and the width parameters of the Trip-Box. We conducted experiments for finding a suitable width of the Trip-Box for speeds between 0 - 80mph and by varying sampling intervals from 1 - 5Hz. We also determine the guard flags based on distance traveled and elapsed time calculated after crossing the boundaries of the Trip-Box to prevent the repeated triggering of same events. Evaluating experiments using a GPS with 5Hz sampling rate and a standard deviation of 7.5m for a 95 percent accuracy, our results show that on crossing a Trip-Box of width 2m and length 28m is 95 percent reliable when the object moves with a speed less than or equal 20mph.

Acknowledgements

I sincerely express my heartfelt gratitude to my guide Professor Marco Gruteser, Department of Electrical and Computer Engineering, for his guidance and continued assistance for completion of my thesis. I thank Professor Predrag Spasojevic and Professor Yanyong Zhang for accepting to be in my defense committee.

I also thank Suhas Mathur, Tong Jin, Chandrasekharan Raman and Sanjit Kaul for helping me to collect huge amounts of data for performing my experiments. I wish to express my sincere thanks to all the faculty and staff members of ECE Department for their unstinting support.

I am extremely indebted to my sister Gayathri Chandrasekaran and all my family members for all that they meant to me during the crucial times of my thesis completion. I also thank my friends for their support.

Dedication

Dedicated to my family

Table of Contents

List of Figures

Chapter 1

Introduction

Advances in location based technologies has enabled a variety of interesting applications. Some of these applications require activating events based on geographic location of the object both for indoor and outdoor settings. For instance, Triggering the starting and stopping actions for data collection as the vehicle approaches and leaves a certain geographical location can significantly reduce the need for human intervention in this process thereby enhancing driver's safety. With the development of different vehicular applications, there is a need for collecting enormous amounts of data under different traffic conditions at various speeds both on the highways and city roads. This process could involve acquiring data on specific road segments repeatedly over long periods of time. Such repetition could become challenging since it requires the experimenter to be proactive in starting and stopping the data acquisition process several times during the drive. For example, halting in a highway traffic to initiate the data collection may cause unnecessary traffic blocks and accidents. The automation technique that we propose in this thesis would mitigate the dependency on the end-user to perform such actions explicitly.

With the increasing useage of smart phones enabled with the GPS navigation systems, there is wide range of applications which require updates or event triggering upon reaching certain geographical locations. Few examples of such applications are

- Advertising upon crossing a particular geographic location.
- Provide location-based warning to the users to give updates about environmental conditions of a particular location.
- Automating the opening and closing of the barriers as the vehicle approaches the

toll booths can eliminate long queues near the toll booths.

For all the above applications, there is a need to trigger some event as the desired object gets closer to some geographic location. In this thesis, we propose the concept of a Trip-Box for automating location based triggers and demonstrate its effectiveness for the vehicular data collection application. Trip-box is a rectangular box enclosing a geographical area with its orientation towards geographic north and east direction. We can define Trip-Box on Google Earth and import it into the trip-box system to perform specific action as the object enters or leaves the area enclosed by the defined trip-box.

In order to perform such location based task initiation, it becomes necessary to know the current location of the object. There are several proposed techniques to determine the location of the object outdoors. [16] can determine the location of a Wi-Fi enabled device outdoors with a median error of 20m. [4] detects the location of a GSM enabled cell phone outdoors with a median accuracy of 96m. Global Positioning System (GPS) is the most common outdoor localization system with a location determination error in the orders of a few meters. Thus, we choose the GPS system for location determination and use the obtained location to automate events. However, even the small location errors introduced by the GPS device can make the automation technique harder to solve. For example, the location errors introduced by consecutive GPS measurements can make the desired object appear within or outside the area enclosed by the trip-box thereby erroneously starting and stopping the experimental data collection several times. Here, in this work, we present the solutions to prevent such repeated event triggers near the boundaries of the trip-box by initializing guard parameters.

In the literature we have seen several other techniques that were proposed to automate event triggers. For uploading traffic information to the server, the concept of Virtual Trip-Line (VTL) [6] was proposed. The Trip-Line is a virtual line segment drawn across the road to mark a geographic location. The idea of VTL is very much similar to our method, however, it is necessary to define a start and a stop location explicitly. Also in case of any deviation from the assumed path due to some hindrance where the Trip-Line is undefined, the data collection process needs to be stopped explicitly by the end user. There are also chances of missing a Trip-Line when there is a deviation in the GPS readings. However, these problems can be overcome by enclosing this geographical region as a box, where on entering this region would automate the action of collecting the data and leaving this area would stop this process.

To summarize, we make the following contributions in this thesis:

1. We propose the concept of Virtual Trip-box which makes use of location updates from the GPS device attached to the object for automating events.

2. We implement our system and demonstrate its effectiveness using real experiments for the vehicular data collection application.

3. We also experimentally evaluate the Guard parameters to overcome the effects of the errors introduced by the GPS system.

The rest of the thesis is organized as follows. Chapter 2 describes the related techniques presented in the literature and provides its advantages and disadvantages over the scheme proposed in this thesis. Chapter 3 presents our system requirements and system architecture and explains in detail the Virtual Trip-Box detection algorithm. In chapter 4, we describe the various experimental goals and state the procedure to evaluate the different parameters. In the later part of this chapter, we provide a detailed analysis of the our results. Conclusion and future work are described in the Chapter 5.

Chapter 2

Related Work

Numerous applications have been developed, both for the indoor and outdoor settings, based on the location updates. This includes applications such as access to a restricted parking lot on approaching the barrier, triggering the opening of toll gates, facilitating lab experiments on crossing a particular location, tracking systems that requires regular location updates, security access to the doors of the buildings etc. Most of these applications require automatic triggering of event when the object crosses certain location. With the proliferation of commodity smartphones equipped with GPS navigation system opens up the attractive possibility of using position samples from drivers phones for several location based applications. We, in this paper, propose the use of these position samples from the driver's phone in combination with the virtual trip-boxes for the applications that require the automation of events.

2.1 Localization based techniques

Localization is the process of determining the location of a wireless device. Localization modalities can vary from using Received Signal Strength to Time of Arrival to Angle of Arrival, etc. Using RSS is the most common technique where the transmitter estimates the signal strength from all the surrounding receivers and calculates its location using the process of triangulation. Several localization techniques have been developed in the past that can detect the location of GSM, Wifi, Zigbee or bluetooth enabled devices. We will here discuss the accuracy limitations of such techniques and analyze its usefulness for applications requiring location based event triggers.

2.1.1 GSM for localization

The GSM-based localization which uses cell phone signals can be used for both indoor and outdoor applications. It uses lesser battery power and [13] reports the median accuracy for the GSM for indoor localization is about 4m. However the outdoor localization has an accuracy of about 44m [8]. The poor accuracy outdoor makes the GSM based systems unsuitable for automatic event triggers. In comparison, the GPS has a higher accuracy of 10m [2] making it much more useful.

2.1.2 Wifi based localization

[3] [16] discuss different techniques to detect the location of wifi enabled devices. Different accuracies ranging from 1 - 10m have been reported in these papers. Placelab [16] has studied the accuracy of wifi localization outdoors and the median reported accuracy is 20 - 30m. Again, its accuracy limitation outdoors make wifi less useful for applications requiring automated triggers based on locations.

2.1.3 Sensor Networks

The Boundary detection technique citied in paper[9] and the location tracking approach discussed in MoteTrack [10] uses RF localization method in order to locate the moving object. It uses a Zigbee Network for location estimation and the localization error reported in [10] is about 3.9m. This scheme also requires a proper infrastructure set up. In contrast the GPS based systems do not require any special infrastructure set up.

2.2 Boundary Detection

The paper [9] describes the boundary detection scheme, where the target obtains its location information by exchanging messages through becons. In order to reduce the energy consumption, they use an adaptive sampling mechanism where the sampling rate is adjusted based on the speed obtained from the accelerometer that is attached with the target. When the target comes closer to the critical region the sampling rate is increased to obtain the exact location where as when the target is away from the boundary the sampling rate is lowered. The scheme described here has an infrastructure set up and used indoors where as our scheme, which obtains the location updates from the GPS requires no infrastructure set up and can be used outdoors. This paper is concerned about energy efficiency and hence has used adaptive sampling technique however, in our method energy gets generated from the vehicular batteries and hence we use the sampling rate of the GPS device.

2.3 Virtual Trip-Lines

The Virtual Trip-Line [6] uses a similar idea to our trip-box based solution. Trip-Line are imaginary location on road which when crossed by the moving vehicle initiates a communication from the phone to the central system with the following information (carid, GPS location). This paper discusses the traffic monitoring application where a VTL database on the server side stores the trip-line's GPS coordinates. The client (mobile device) downloads the tripline location on its phone and sends an update to the server whenever it crosses the trip-line. In order to get the traffic updates and the speed estimate, the placement of the trip-lines is a major factor. The Virtual Tripline paper proposes an even placement approach for placing the trip-lines and describes the tripline spacing parameters and exclusion area for the privacy purposes. In contrast to the tripline, trip-box is a collection of trip-lines placed in the form of rectangle. The use of a box-like placement enables clustering of trip-lines which makes more sense for applications that require explicit start or stop triggers on reaching certain geographical region. In this work we discuss the use of trip-box for triggering events where on crossing the one side of the rectangular box automates an event and the opposite side would automate a different event.

The trip-box idea can also be used for applications which require the uploading of information periodically to the central server as discussed in [6], [7], [17]. The [7] and [17] uses an intermittent Wifi connection to up load the information to the central server but the scheme proposed in the [6] is similar to our scheme where the information is uploaded to the server only on crossing the tripline or the trip-box. The former ones cannot control when to send an update to the server, which could lead to heavy battery drain. In contrast, our trip-boxes can define when to send the update to the central server.

2.4 Applications requiring Automation Techniques

There are various applications which require automatic triggers such as opening of office doors inside the building, opening and closing of the barriers in the parking lots and toll booths, facilitating lab experiments etc. Here in this section we discuss the automation techniques used for such applications and state its limitations.

2.4.1 Facilitate Data Collection

A great challenge is involved in collecting the huge amounts of data for performing the lab experiments. The applications discussed in [5] and [11] there is a need to periodically record the GPS and sensor informations. These papers initiate the recording of the information manually, while this is feasible but its prone to manual error. It is necessary for the driver to to stop driving everytime in order to start or stop the experiment or the driver has to be accompanied by another person to perform this action. Overall, this can be tedious. We on the other hand, can automate the periodic collection of the data by defining the trip-boxes using Google Earth at appropriate geographic locations.

2.4.2 Automating barriers

[12],[14] explains how Radio Frequency IDentification (RFID) technology can be used in parking barriers application. The RFID technology requires an RFID transmitter to be embedded in front of the car to send a signal to the RFID reader for verification. On receiving authentication from the RFID reader and it signals to lift the arm to let the car to enter or leave the parking lot. In this technique there is no need for the driver to explicitly initiate the opening of the barrier since the tag is detected as soon as it is in close proximity to the barrier. But this also requires the car to slow down significantly in order for the RFID reader to detect the RFID transmitter. Further modification or

updation of the parking lot's access control policy requires significant effort since this is a decentralized system. Since the trip-box based solution proposed here gets GPS location updates, it can trigger the opening of the parking arm as soon as the vehicle enters the trip-box corresponding to the parking lot. This eliminates the need to slow down the vehicle for entering the parking lot thereby reduces the traffic congestion problem of waiting in front of the barriers. The updation in our technique becomes simpler and with the software based solution changes can be incorporated easily.

2.4.3 Accessing doors inside buildings

[1] discusses the process of automating the access to the doors inside the building using bluetooth technology. The users are equipped with a Grey phone which is bluetooth enabled and the door to be accessed must be fitted with a bluetooth enabled computer. When the user needs to access a particular door, the door that user needs to access must be chosen from the list and a PIN has to be given as input. Otherwise it is also possible to enable the bluetooth far away for example from the parking lot and when the user along with the Grey phone comes to the proximity of the bluetooth range, it would trigger the opening of the door. This technique which is similar to our system, where it uses a bluetooth technology for performing access control while we use location updates from the GPS to automate triggering of events.

Chapter 3

System Design, Algorithm and Implementation

In this chapter we explain our system architecture and describe the Virtual Trip-Box Detection Algorithm. We discuss the challenges we faced and present the solutions for overcoming them. At last we give the implementation details of our system.

3.1 The Virtual Trip-Box Concept

In this thesis we propose the concept of Virtual Trip-Box for automating the events based on geographic locations. A Virtual Trip-Box is a rectangular box that encloses a geographical region and defined by the vertices of one of the diagonal GPS coordinates. The area enclosing the trip-boxes can be separate or overlapping over one another. It has its orientation of towards the geographic north and geographic east directions.

3.2 System Architecture

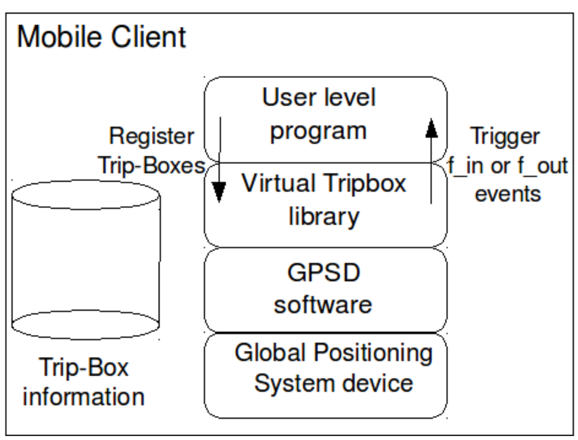

Figure 3.1: System Architecture

Our system architecture is shown in the figure 3.1. The base of our system architecture is the GPS receiver, as the working of the system depends on the signal received from the GPS device. On connecting this GPS device we receives the information from the visible satellites. The GPSD software reads information received from the GPS receiver and makes the data available for other applications. This provides the necessary information about the current location such as latitude, longitude, speed, timestamps, number of visible satellites etc. Our Virtual Trip-Box library reads the GPS information passed by the GPSD daemon process and interprets the position of the object with respect to the Trip-Box. On receiving the location updates after defining the placement of the Virtual Trip-Box on Google Earth and including the trip-box library in the user defined program, we invoke the detection of trip-boxes function from the Virtual Trip-Box library.

3.3 Algorithm

3.3.1 Virtual Trip-Box Detection Algorithm

The Virtual Trip-Box detection gets input from the GPS receiver that is serially connected to the PC and produces event triggers as output. On receiving the latitude and longitude information of the object from the GPS, the location of the object is compared with the coordinates of the vertices of all the registered trip-boxes to determine the position of object with respect to each of the trip-boxes. We define four different states of the object with reference to the trip-boxes. These are

- State-1: Entering the Trip-Box -The previous GPS coordinate is outside the Trip-Box and the current GPS coordinate is inside the Trip-Box. In this state we inititate the fin event which for data collection process starts collecting the data.

- State-2: Inside the Trip-Box - Both its previous GPS coordinate and current GPS coordinate inside the Trip-Box. In this state we stay idle and just receive the next available GPS coordinate.

- State-3: Leaving the Trip-Box - The previous GPS coordinate is inside the Trip-Box and current GPS coordinate is outside the Trip-Box. On leaving the Trip-Box we trigger the fout function and in data collection process, it triggers the stopping of this event.

- State-4: Outside the Trip-Box - Both the previous GPS coordinate and the current GPS coordinate outside the Trip-Box. As we are already outside the Trip-Box region, we do not initiate any event.

We refer State-1 as start state, State-3 as end state and State-2 and State-4 as intermediate states. Figure 3.2 shows the state diagram based on the location of the object.

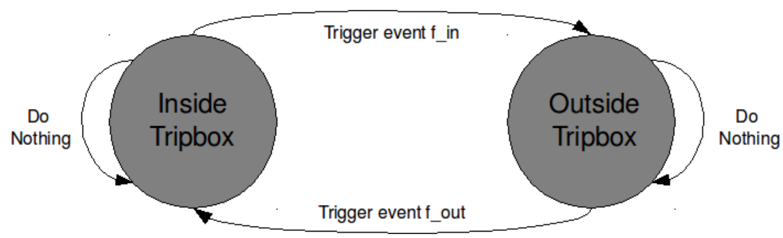

Figure 3.2: State Diagram for an object under motion

For instance, when the object is State-1, where previous state is outside the Trip-Box and the current state is inside the Trip-Box as shown in 3.2, the function fin is triggered. Similarly when the object is in State-3, where it has a previous state as inside the Trip-Box and current state as outside the Trip-Box, fout function gets triggered. When the object is in State-2 or State-4, the system waits for the next location update from the GPS receiver.The above steps are repeated on receiving the location updates from the GPS.

In this algorithm we store the previous and current state information of the object as an entity of the Trip-Box. The state information includes the GPS coordinates, timestamps and the location of the object in regards to each of the Trip-Box. Using this recorded state information of the object and the current location updates from the GPS, the algorithm initiates the event triggers. As shown in figure 3.3 we determine the position of the object, by initially locating the object corresponding to the Trip-Box and check if the received coordinate is the first coordinate inside or outside the Trip-Box, in which case we initiate the triggering of the fin or fout functions respectively and update the current state information.

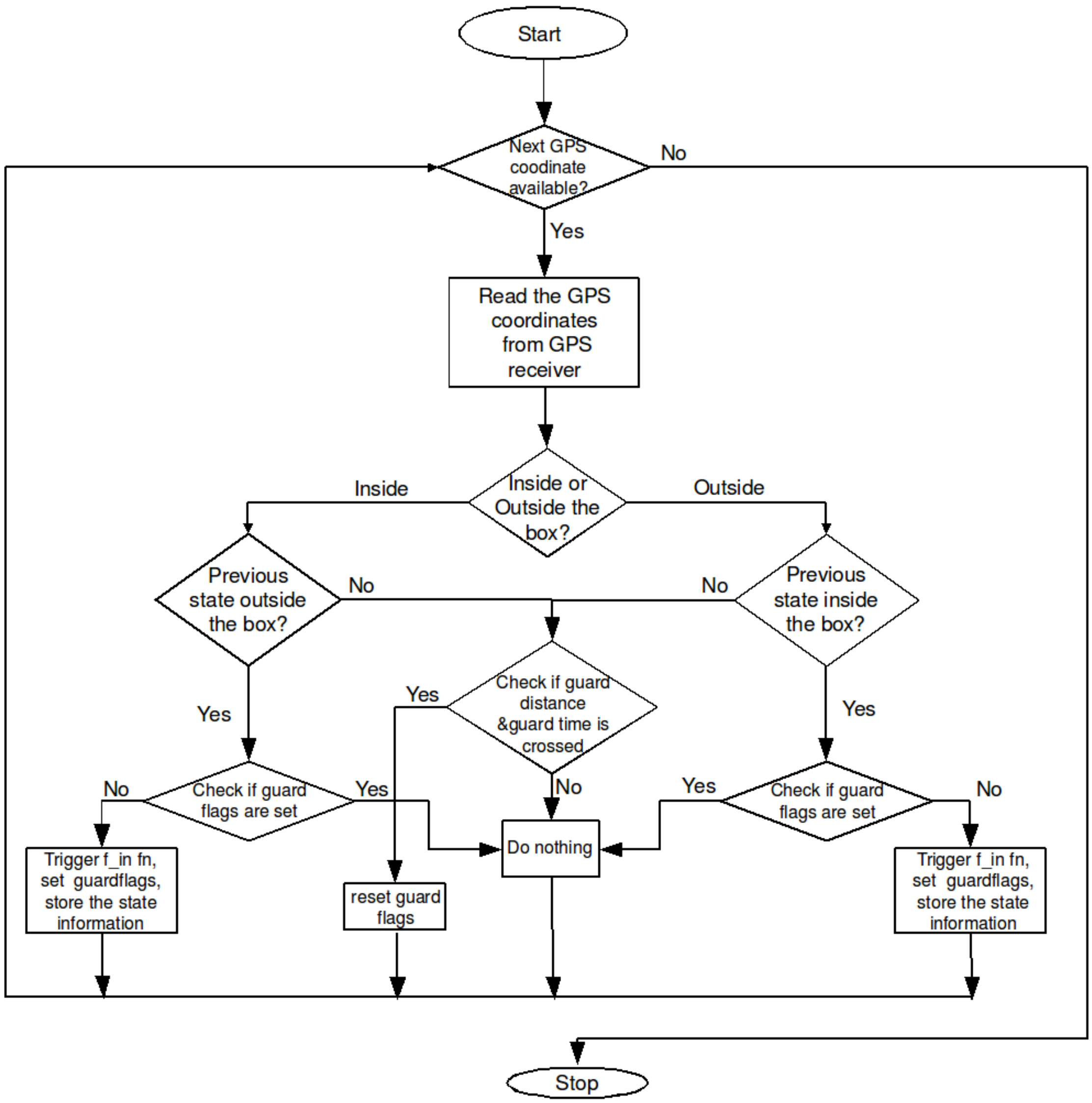

Figure 3.3: Flow Chart of Algorithm

The unexpected behavior of GPS device on consecutive GPS samples adds complexity to the detection algorithm by causing unnecessary starting and stopping of events. GPS device sporadically introduces a back and forth oscillations of the GPS readings and hence we define Guard Distance and Guard Time for the Trip-Box to eliminate repeated event triggers from the same Trip-Box. Guard Distance is defined as a minimum distance that must be traveled after crossing the boundary and before reaching the same Trip-Box. This parameter is essential to avoid triggering of same event consecutively due to jumping of GPS coordinates, in particular when the vehicle is traveling slowly while crossing the boundary of the Trip-Box. The Guard Distance parameter also prevents the vehicle to travel in reverse direction upon entering or leaving the Trip-Box and requires the object to travel a minimum guard distance inside or outside the Trip-Box respectively. Similarly Guard Time is defined as a minimum time that the vehicle must travel on entering or leaving of the Trip-Box before coming back to the same edge of the Trip-Box. For instance, the vehicle stops for signal or a stop sign and the boundary of the Trip-Box is closer to this location, there is a possibility of accidental triggering of same events repeatedly. In order to avoid such exceptional situations, we have defined a Guard Time. However, in our demonstration we used Guard Distance parameter as it helped to solve our challenges faced in case of data collection process. Guard Time can also be used when the average speed and the time of travel within the Trip-Box is known.

3.4 Implementation Details

Our system consists of a Garmin 18-5Hz GPS device, which records five GPS readings every second, a virtual trip-boxes which mark the geographical location defined on Google Earth and a mobile node with one USB slot for connecting GPS.

3.4.1 Global Positioning System

Our Garmin 18-5Hz GPS device produces five location samples in one second. It is Wide Area Augumentation System enabled and with the accuracy specified in the manual as 3m. It outputs location information as standard NMEA statements.

3.4.2 User Defined Program

The user level program is a simple ruby code, that needs to include the virtual library. Our system is designed to handle multiple trip-boxes identified with unique id. We

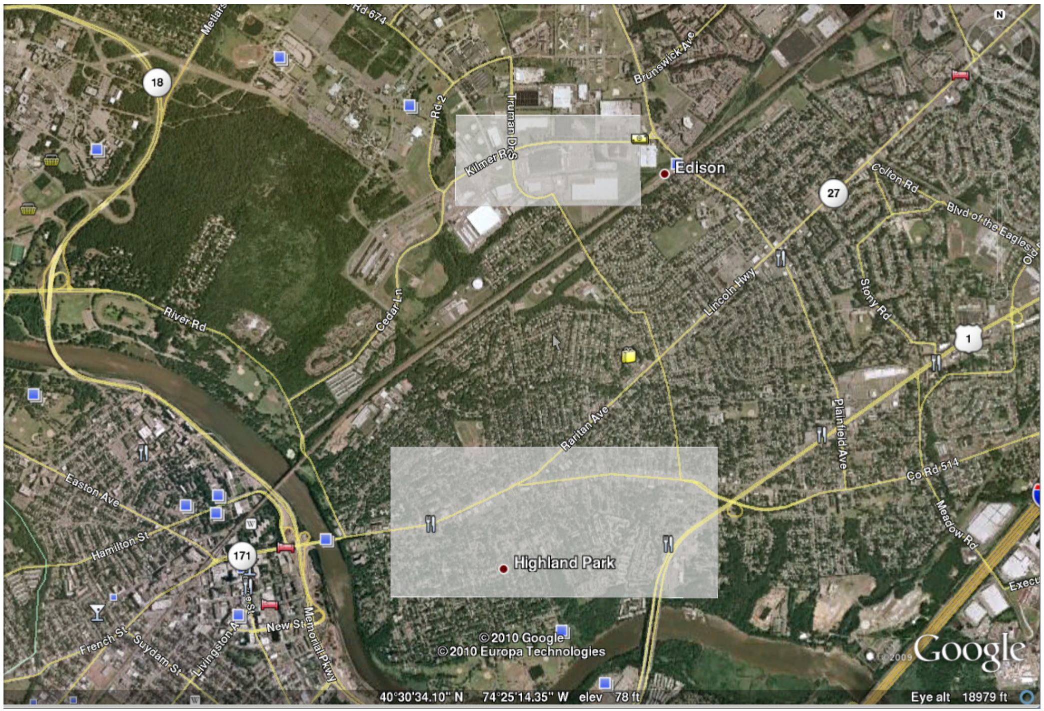

Figure 3.4: Creating Trip-Box using Google Earth

define these Virtual trip-boxes on Google Earth as shown in figure 3.4 and save the file. As it might be difficult to draw a perfect rectangle hence we let the user to use Pushpins to define the diagonal of the rectangular region to define the Trip-Box and register the id of the Trip-Box and the vertices of the diagonal with the id of the pushpins. This file, which is of XML file format, is exported to our Trip-Box library via user program. From the trip-box library, we read this file from the beginning and search for name and coordinates patterns, for identifying the Trip-Box id and its corresponding latitude and longitude coordinates.

On initializing the Trip-Box with the id, we read the respective coordinates of the Trip-Box from the file to register them. After registering the trip-boxes, the call returns back to the user program, which then initiates the procedure for detecting the Virtual trip-boxes. Here in the figure 3.5 we show the sample user code for using the Virtual Trip-Box library. On waiting for specified interval of time, the virtual trip-box library, returns the call back to the user program. In the absence of the GPS device, this trip-box library returns an error message back to the user program.

```
require "tripboxlibrary"

def start_fn
	puts "start experiment"
end

def stop_fn
	puts "stop experiment"
end

Tripbox.readFile('Tripbox.kml')
TripBoxController.register(t1=Tripbox.new('Tripbox 1','Pushpin 1','Pushpin 2',:start_fn,:stop_fn))
TripBoxController.register(t2=Tripbox.new('Tripbox 2','Pushpin 3','Pushpin 4',:start_fn,:stop_fn))
TripBoxController.register(t3=Tripbox.new('Tripbox 3','Pushpin 5','Pushpin 6',:start_fn,:stop_fn))
TripBoxController.waitForTripboxes(500)
```

Figure 3.5: User Program

3.4.3 Virtual Trip-Box Library

Our software also parses the Google Earth file that is passed from the user program for initializing and registering the trip-boxes. The Trip-Box is initialized as (id, 'Pushpin1', 'Pushpin2', fin, fout) where the id refers to the identifier of the Trip-Box, Pushpin1 and Pushpin2 refers to the identifier of diagonal vertices of the rectangular box and fin and fout are the functions associated with the Trip-Box. The fin and fout gets initiated upon entering and leaving the Trip-Box respectively. The identifiers used for defining the pushpins needs to be unique for all the defined trip-boxes. Use of Pushpins to define the diagonal vertex of the rectangle reduces the effort of manually hardcoding the coordinates and makes the system more dynamic and easier to use. Thus with (x1,y1) and (x2,y2) as the vertices of the diagonal of the trip-box, the other two derived vertices are (x1,y2) and (x2,y1) and hence the orientation of the Virtual Trip-Box is towards the geographic north and geographic east.

The Guard Distance and Guard Time are estimated from recorded state information of the object. We set the guard flags when the object is in start or end state and during the intermediate states we reset these flags upon crossing the guard Distance and guard Time. On receiving the current location update, we calculate the distance between the stored state coordinates and the current GPS coordinates and determine the distance traveled after setting the guard flag. Similarly we calculate the elapsed time from the current timestamps to the stored timestamps. The above procedure is followed for each of the trip-boxes on receiving the location update. With the absence of GPS device the system stops working.

Chapter 4

Experimental Setup and Results

In our framework, we have used Global Positioning System for providing the location updates and Virtually define Trip-Boxes as location markers. On using Garmin 18-5Hz GPS, the challenges we faced includes the possibility of skipping the Trip-Box and accidental repeated triggering of events upon crossing the boundary of the Trip-Box. The former depends on the dimensions of the Trip-Box and the latter occurs due to the unexpected behavior in the consecutive GPS samples. In this chapter we present our experimental goals and describe the different experiments we performed in detail to evaluate various parameters of the Trip-Box. Also we present solutions to resolve the above mentioned problems and in the later part of the chapter, we discuss the results obtained from the experiments we conducted.

4.1 Experimental Goals

This section briefly states the major experimental goals. These were

- To determine the probability of skipping the Trip-Box for a vehicle traveling at different speeds using attributes of the Trip-Box.
- To find the effects of the GPS refresh rate on the necessary width parameter of the Trip-Box.
- To determine the Guard Distance and Guard Time for a Trip-Box to prevent triggering of same events on reaching the boundary of the Trip-Box.
- To calculate the accuracy of the GPS device.

4.2 Data Acquisition

In order to perform different experiments we collected GPS traces over three different roads at various speeds. These data collection includes the traces from inner roads, with speed 20 - 30mph and on highways, with the speed varying approximately from 35mph to 75mph. The traces were collected on roads with minimum of two lanes and by driving on either lanes of the road. We also acquired the GPS data continuously for 48 hours by placing the GPS device connected to a node in a static location. From the GPS device we collected the GPS device timestamps, latitude and longitude and the speed of the object. For conducting experiments, we used the timestamps in epoch, latitude and longitude in decimal and speed in mph. The same traces were repeatedly used for performing different experiments by modifying the script.

4.3 Implementation details

The automation of the events depends on the speed of the object, the refresh rate of the GPS and the parameters that define the Virtual Trip-Box. In this section we explain our reasoning behind the various experimental parameters choices and elaborate our experimental procedures.

4.3.1 Minimum Required Width and Length of the Trip-Box

On defining the Virtual-Trip-Box over a smaller segment of the road as shown in figure 4.1, such that on crossing the region enclosed by the Trip-Box, there is a chance of skipping the Trip-Box due to the absence of any location estimates inside this region. We evaluate two parameters that attribute to the probability of missing the Trip-Boxes while driving at any speed. These are

- the width parameter, which is measured as a minimum distance between the actual event trigger and the ideal position of the Trip-Box at different speeds.
- the length parameter, which is calculated as the minimum measure the Trip-Box needs to be extended beyond the boundaries on either side of the road.

Figure 4.1: A scenario of skipping the Trip-Box

Ideally, using the Detection of Virtual Trip-Box Algorithm, the minimum required width of Trip-Box can be rounded off to zero on receiving GPS samples continuously. But, on uing Garmin 18 - 5Hz GPS device for our system, we receive location updates every 0.2s and when the target is under motion, it covers a small unit of distance in between two consecutive GPS samples. As we know the distance traveled is directly proportional to the speed of the object, this intermittent distance covered in between two consecutive GPS samples increases with the increase in speed. Therefore, this causes the consecutive GPS coordinates to get displaced by small units and hence leads to the possibility of missing the Trip-Box in case if the width of the Trip-Box is less than the intermitted distance traveled.

We analyze the effects of speed and sampling rate of GPS on the dimensions of the Trip-Box. The width of the Trip-Box pertains to the horizontal error of the GPS samples and the length refers to the vertical error of the GPS samples with respect to the road. Horizontal error is the displacement of the GPS samples along the direction of motion of the object and Vertical error is the deviation of GPS samples from the road. As the horizontal error is affected with the change in the speed of the object and sampling rate of GPS, hence we analyze the effects of speed and sampling rate on the Width parameter of the Trip-Box.

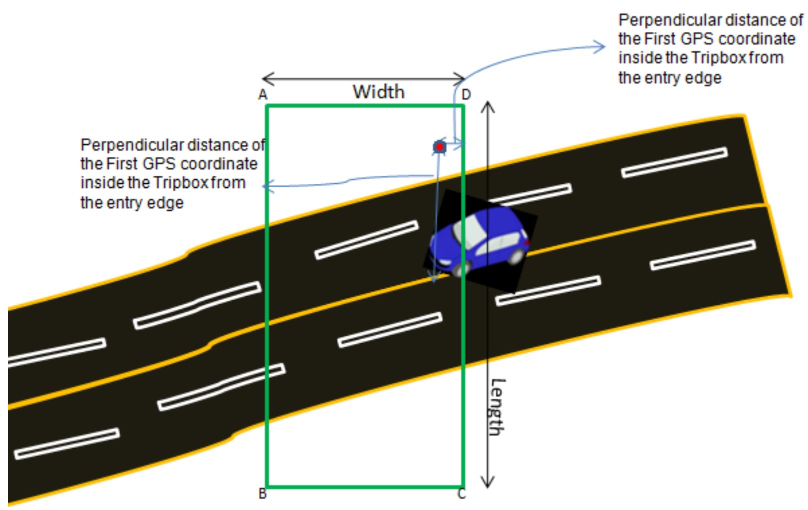

Figure 4.2: Calculating the Width of Trip-Box

In order not to miss a trip-box, the trip-box should have a minimum width equal to the horizontal error of the GPS. Firstly we describe the experiment we performed for analyzing the effects of speed on the minimum required width of the trip-box. For evaluating the minimum necessary width of the Trip-Box, we carried out experiment on a long stretch of a road of about 0.4 miles by defining six virtual Trip-Boxes each 25m wide on Google Earth as shown in the figure 4.2. For analysis of Width we used the traces acquired on both the highways and the urban roads. We calculate the minimum width required by finding the perpendicular distance between the first GPS coordinate that is located inside the Trip-Box and the ideal GPS coordinate that is located on edge of the Trip-Box through which the vehicle enters the Trip-Box. For instance if the car enters the Trip-Box through the side CD of the Trip-Box as shown in the figure 4.2, the width is determined by calculating the perpendicular distance from CD to the point E. We performed this experiment for different speeds varying from 10mph to 75mph to obtain a relation between the speed and the minimum width of the Trip-Box.

We evaluate the minimum necessary length of the Trip-Box as the vertical displacement of GPS coordinate from the road by calculating the perpendicular distance of the GPS coordinates obtained from the traces to the median of the road. The experiments were conducted by simulating it on the data set which was collected over a period of

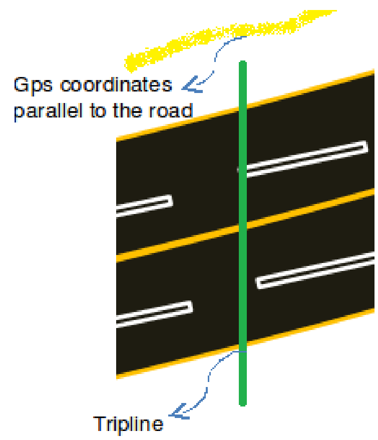

Figure 4.3: Scenario for missing Trip-Line

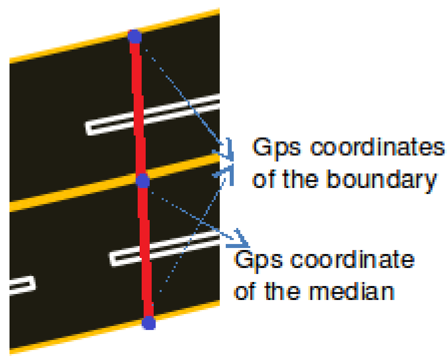

Figure 4.4: Measuring the Length of Trip-Box

30 days by driving on different lanes on the road. For evaluation of the length of the Trip-Box we selected three reference locations as (x1,y), (x2,y) and (x3,y) along the same longitude, such that, two points (x1,y) and (x3,y) were located on either side of the boundaries of the road and one point (x2,y) corresponds to the median of the road as shown in the figure 4.3. We measured the perpendicular distance between the GPS coordinates from the traces to the median point of the road over a range of 0.0000250 on the either side of the longitude(y). We simulated this experiment on 0.4 miles road using six such reference locations.

Using this experiment we also determine the chances of missing the Virtual Trip-Line [6]. The detection of virtual trip-line depends on the GPS coordinates from the GPS device along the direction of motion with respect to the Trip-Line. The detection of Virtual Trip-Line is determined using the intersection of line segments problem, with Trip-Line as one line segment and the imaginary line joining the current and its preceding GPS locations as the other line segment. It is likely that the Trip-Line is missed, when the GPS coordinates fall parallel to the road such that there occurs no intersection between the Trip-Line and the line joining the consecutive GPS locations. Figure 4.4 illustrates this scenario, where the GPS coordinates fall at few distance from one end of the Trip-Line, hence there is a possibility of missing a Trip-Line. However, the procedure for evaluating the length of the Trip-Box can be applied to determine the length of the Virtual Trip-Line in order to prevent the chances of missing the Trip-Line.

4.3.2 Sampling rate of GPS

As our system is dependent on the samples received from the GPS device, hence we calculate the effects of the GPS refresh rate on the width parameter of the Trip-Box. Using GPS device of 5Hz sampling rate, the experiments were conducted at six locations along a 0.4miles segment of road. For analyzing the effects of GPS refresh rate, we varied the sampling rate as 5Hz, 4Hz, 3Hz, 2Hz and 1Hz and calculated the perpendicular distance of the actual GPS coordinates from the traces to the ideal position that is located on the entrance or exit edge of the Trip-Box. For a 5Hz sampling rate we read five GPS readings in one second and for 4Hz sampling rate we simulated the experiment to miss every fifth GPS reading. Similarly for 3Hz, we simulated the experiment to read three GPS coordinates and ignore the fourth and fifth readings and so on.

4.3.3 Guard Distance and Guard Time

Guard Distance is defined as a minimum distance that must be traveled by the object upon crossing the Trip-Box, before reaching the same edge of the Trip-Box. This parameter is essential because of the back and forth oscillation of the GPS coordinates. For instance, when the vehicle is closer to the boundary of the Trip-Box, there is a possibility that the same event gets triggered accidentally from the same Trip-Box due to jumping of GPS coordinates in and out of the Trip-Box. Also when the traffic signal or a stop sign coincides with the edge of the Trip-Box and the vehicle comes to a halt for the traffic sign, repetitive triggering of same event might occur. For preventing such repeated event triggers from the same Trip-Box, we define guard distance. In order to estimate the guard distance we use the traces that was collected by placing the GPS in a static location and calculated the distance of all the GPS coordinates from a fixed GPS coordinate. We repeated the experiment by changing the reference point every 1 second, 10 seconds, 60 seconds as the wait time closer to the stop sign or signal will be for short interval of time. Guard Time depends on the speed of the object and time of travel within the Trip-Box including the average waiting time for the signals. Further with the same experiment we also evaluate the accuracy of the GPS device.

4.4 Results

In this section we analyze the results obtained by simulating the experiments on the collected traces. We present the probability of crossing a Trip-Box for the given dimensions. We also derive the relationship between the width of the Trip-Box and speed of the vehicle and we analyze how the sampling rate affects the width of the Trip-Box. Further we measure the minimum Guard distance required to prevent repeated triggering of events.

4.4.1 Effects of Speed on the Width of the Trip-Box

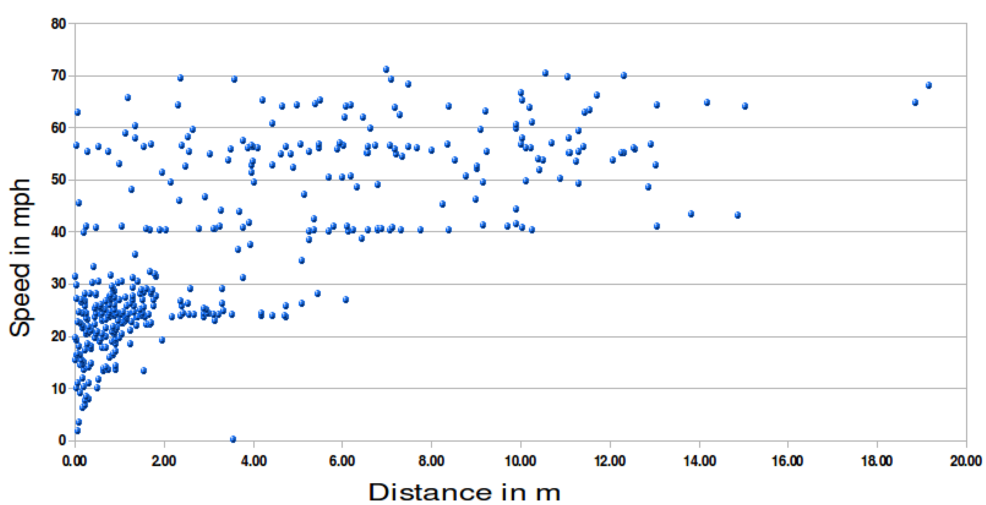

Figure 4.5: Width of Trip-Box

For evaluating the width of the Trip-Box, we calculate the perpendicular distance between the ideal position located on the edge of the Trip-Box and the actual GPS coordinates that causes the initiation of event triggers. The figure shows the Displacement of GPS coordinate from the ideal position in m for different speeds(mph). From the plot shown in figure 4.5 we are able to observe that as the speed increases from 0 to 80mph the distance between the GPS sample from the traces that triggers the events and the ideal place where the triggering of event should occur also increases from 1m to 14m. Thus from the above result we observe that with the increase in the speed, the width of the Trip-Box needs to be increases to prevent the missing of Trip-Box. The reason behind this behavior of increase in width is because for the given interval of time, the distance covered by the object becomes large with the increase in speed. Thus, when the speed increases the distance between the two consecutive GPS readings becomes large within the given interval of time. Nevertheless, when the object moves slowly, this difference in distance will be small and hence the minimum width required is also small. However, there are few samples with lesser displacement from the edge of the Trip-Box, this is due to the possibility of getting few GPS coordinates closer to the edge of the Trip-Box.

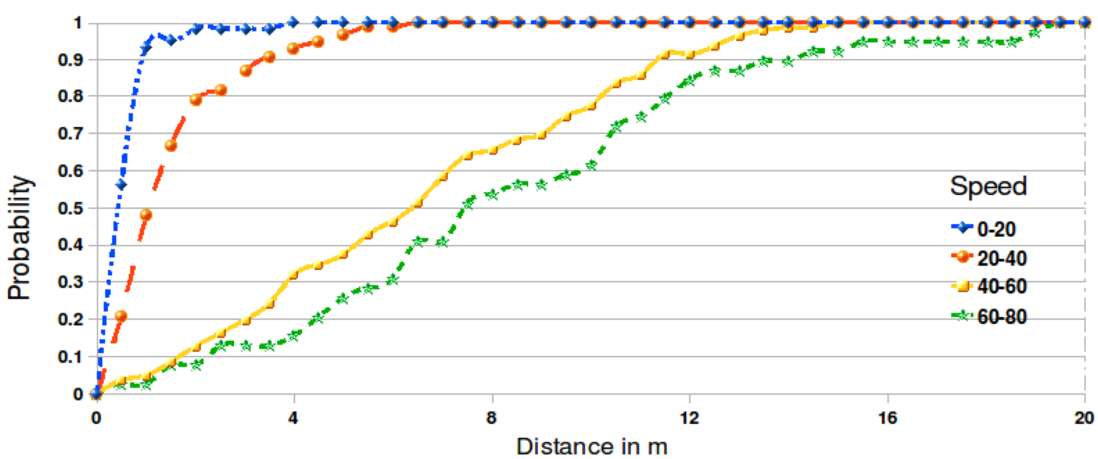

Figure 4.6: For given width, the probability of traversing a Trip-Box

Using the above results we estimate the probability of finding the Trip-Box at different speeds. Figure 4.6 plots the Probability of finding the Trip-Box against the width of the Trip-Box for different range of speeds. From the figure 4.6 we are able to observe that, a vehicle traveling with the speed less than 20mph has a 95 percent probability of crossing a Trip-Box of width 1m. Similarly, a vehicle traveling at a speed of 60mph has a 95 percent probability of not missing the Trip-Box if its width is 13m. The curves for speeds between 40-60mph and 60-80mph looks similar because the number of sample points between 60-80mph is very less compared to 40-60mph as shown in the figure 4.5. But as we can observe from the figure 4.5 the displacement of GPS samples from the entry edge of Trip-Box for 60-80mph is higher than the 40-60mph.

4.4.2 Effects of Sampling rate of GPS on the Width of the Trip-Box

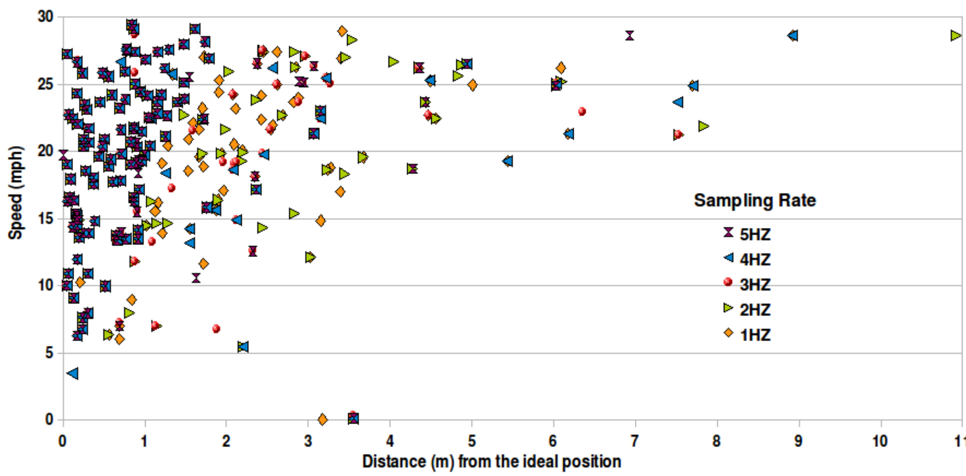

Figure 4.7: Effects of Sampling rate on the Width of Trip-Box

In order to understand the effects of sampling rate on the width of the Trip-Box, we perform experiments by varying the sampling rate and calculate the perpendicular distance from the one side of the Trip-Box, through which the car enters, to the first GPS coordinate that falls inside the Trip-Box. This experiment is similar to the experiment that we performed in section 4.4.1. In order to evaluate the effects of sampling rate, we vary the sampling interval for a given interval of speed. The figure 4.7 shows the scattered plot of the displacement of the GPS coordinate from the ideal position for different drives at various speeds between 0 - 30mph, in a typical city driving. This variation in the distance is observed because as the sampling rate decreases, the number of GPS samples per second also decreases, therefore, on traveling with the same speed the time taken between two consecutive GPS samples increases. With this increased time interval, the object covers a larger distance and hence with the lower sampling rate, this displacement becomes higher. However, this distance can also be small, if we receive the GPS information closer to the edge of the Trip-Box.

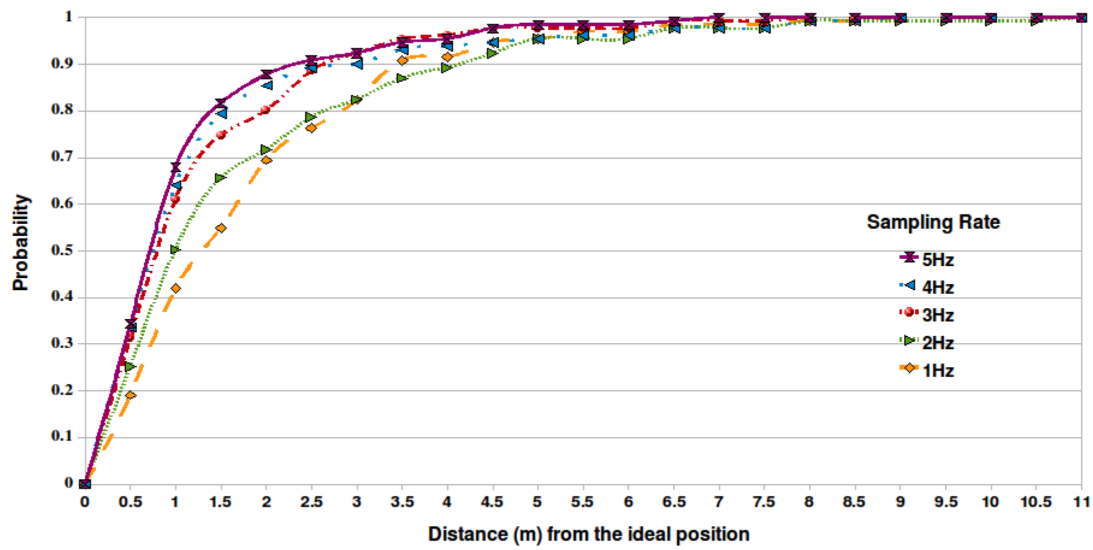

Figure 4.8: Probability of traversing a Trip-Box for a given width for different Sampling rates

From the figure 4.8 we can observe that as the sampling rate increases, the accuracy of GPS also increases and hence the distance from the ideal position becomes small whereas for a lower sampling rate, the refresh rate of GPS device is slow and hence affects the accuracy of GPS. When the accuracy of GPS readings is low then the perpendicular distance from the ideal position is increased. Hence, in order to assure 95 percent accuracy of GPS with 5Hz sampling rate on a road with 30mph speed or less, the width of Trip-Box has to be 3.5m, however for the same road with the sampling rate of 2Hz, the width of the trip-box needs to be 5.5m for achieving a 90 percent accuracy. Thus, if the sampling rate is less then width of the Trip-Box needs to be increased as they are inversely proportional to each other.

4.4.3 Estimation of Length of the Trip-Box

For estimating the length of the Trip-Box we used the data that was collected over a month along the 0.4 miles road. Length is computed as the displacement of the GPS coordinates on either side of the road, from the center of the road as explained in the

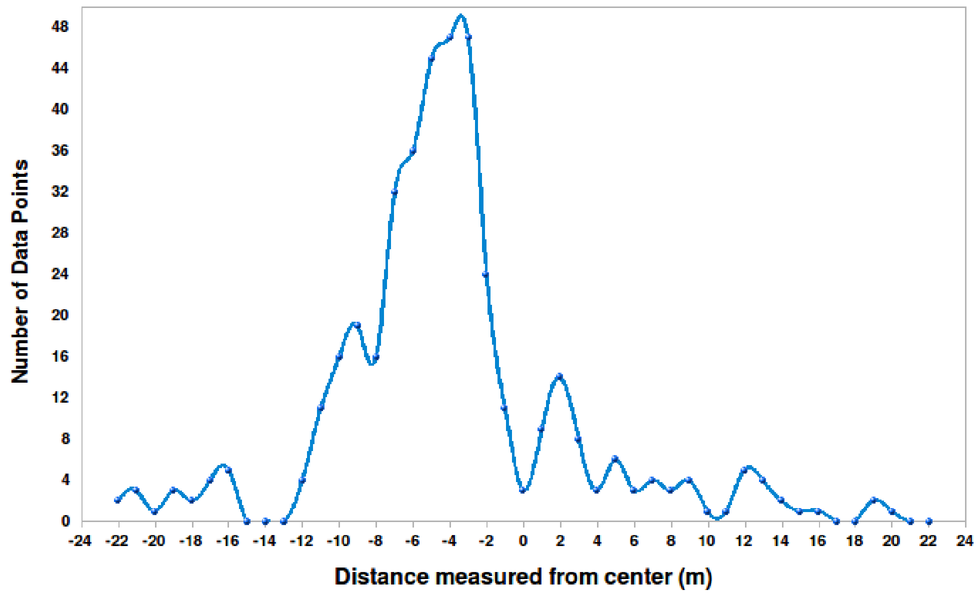

Figure 4.9: Evaluation of length of Trip-Box

experimental details section. Figure 4.9 plots a distribution where the negative sign refers to the distance from the center line along geographic north and positive sign denotes the distance from the center line alone geographic south direction. The car traveled in both the directions and on right and left lanes and with the assumption of average width of road with four lanes as 16m. The median of -4m denotes that the car has traveled most of the times towards the south direction and in the left lane of the road.

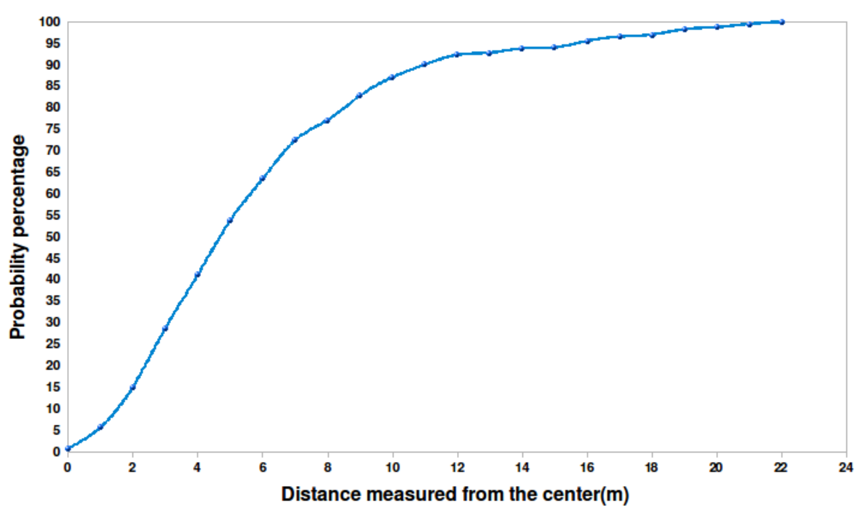

Figure 4.10: Probability of crossing the Trip-Box of given Length measured from the center

Using the above results we calculate the cumulative distribution function for finding the extension length of the Trip-Box on either side of the road. The distance in x axis denotes the extension required from the center of the road and the y axis shows the probability of crossing the Trip-Box. From the Figure 4.10 we can infer that there is a 95 percent probability of crossing a Trip-Box if the extension length of the trip-box from the center is equal to 16m. Hence in order to ensure a 95 percent chances of encountering a Trip-Box or Virtual Trip-Line, the minimum length needs to be 32m. The -4m and +4m are the distance of the left lanes from the center of the road. With the vehicle traveling in the left lane, there occurs a large increase in the value around 4m.

4.4.4 Estimation of Guard Parameters

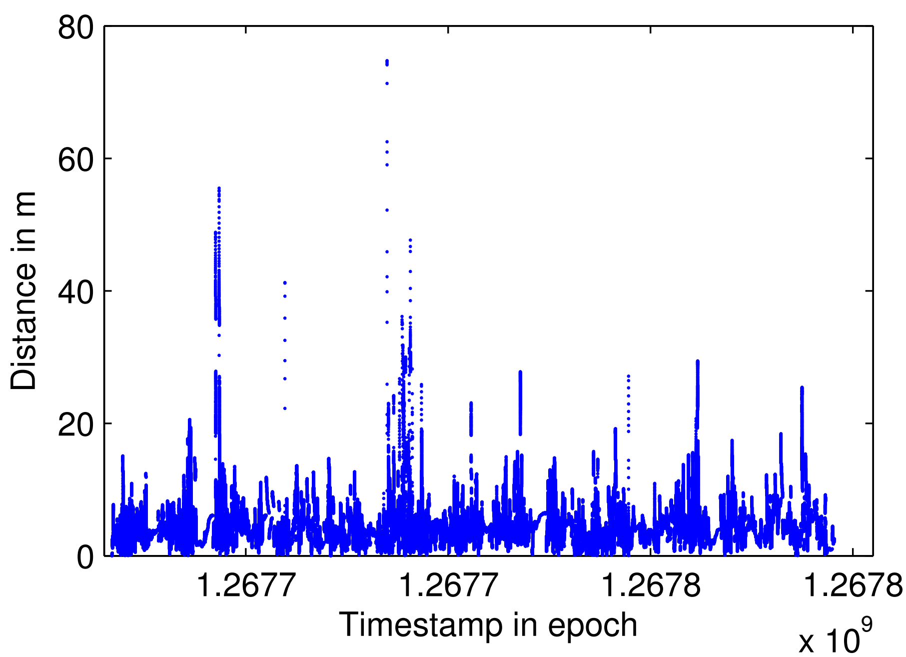

Figure 4.11: GPS Accuracy evaluation

The Guard Distance was evaluated by calculating the standard deviation of GPS samples that were collected continuously for 2 days. We assumed a fixed GPS coordinate for the location as the reference point and calculated the distance of all the GPS coordinates from this reference point. As we are concerned about the short waiting time near the signals and stop signs, we varied the reference point every 1s, 10s, 60s, 48 hours and repeated the experiments again. For example, in case of 1s time interval, with an assumption of receiving 5 samples per second, the first sample is initialized to be the reference location and this is updated by 6th sample that was received during the start of the next second. With time interval as 10s we change the reference location with the newly updated position every 10s however for the fixed point experiment, we used the first fixed location as the reference location. The variation in the distance from the fixed reference point over the time period of 48 hours is shown in the figure 4.11.

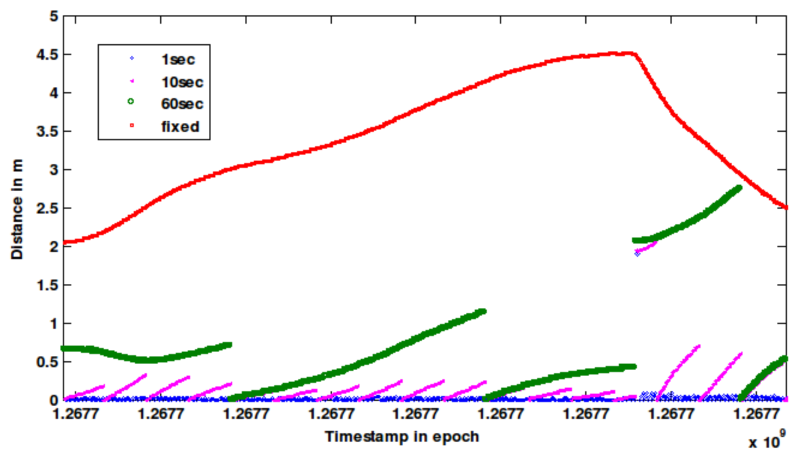

Figure 4.12: Deviation of GPS samples calculated at different time intervals for 5minutes

As we can observe from the plot that there could be a maximum deviation of 75m from this fixed location, however on varying this reference location every 1s, we observe a maximum deviation of 40m. Figure 4.12 shows the deviation in the GPS coordinates for 5 minutes, calculated by changing the reference point at different intervals. As we can see for 1s variation, the displacement of consecutive GPS coordinates was almost zero. However for the 10s interval, we were able to see a mean of about 1m. For a fixed reference point, the mean is approximately 4.314m. So with the increase in the time interval, the deviation from the reference coordinate becomes large.

We calculate the percentile rank of the distances of all the GPS coordinates from the reference point in order to get the reliable measure of guard distance for different time intervals. The plot in the figure 4.13 shows the percentile rank along y axis against the distance from reference position along the x axis. From this, we are able to infer that about 95 percent of the data are below 8m for a fixed reference location, but on varying reference location every 10s we observe that the 95 percent of the data are below 2.5m.

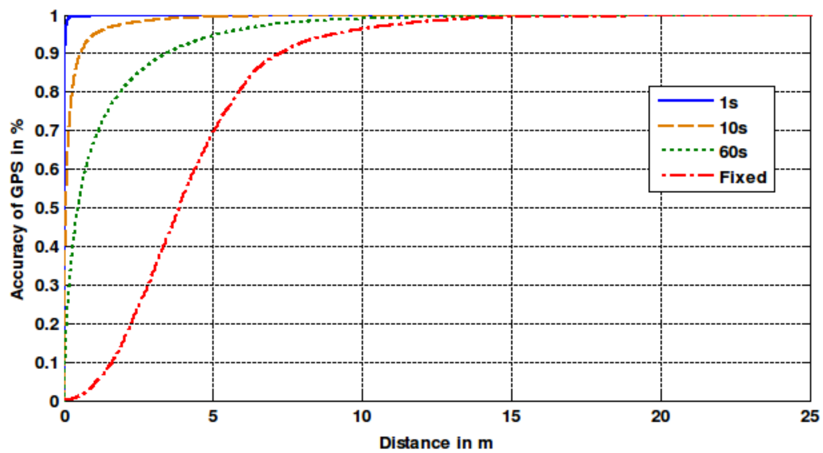

Figure 4.13: CDF of deviation of GPS on changing the reference point at different intervals

If the signal coincides with the boundary line of the Trip-Box, the waiting time might 60seconds, hence for a 95 percent chance of not initiating the same event, we will require a guard distance of approximately 6m. Therefore a guard distance of 3m and a guard time of 10s can ensure 90 percent reliability of not triggering the same event of the Trip-Box. In other words, after the vehicle exists a Trip-Box, it has to travel at least for 10 seconds with a minimum distance of 3m before entering the same trip-box.

Chapter 5

Conclusion and Future Work

In this work, we proposed the idea of the Virtual Trip-Box for automating events. Specifically, we demonstrated the effectiveness of Virtual Trip-Box for automating data collection for vehicular experiments upon reaching a specific geographic location. Through long term experiments, we found out that using a GPS with 5Hz sampling interval for data collection on highways with four lanes and speed limit between 40-60mph the length and the width of the trip-box needs to be 28m and 13m respectively in order to achieve 95 percent accuracy. The width of the Tripbox is approximately 2m for local roads with speed limits between 0-20mph. In addition to the size of the trip-box, we also showed the need for guard parameters in order to compensate for the GPS errors. We found that the typical guard distance to minimize the effects of GPS oscillations on the vehicular data collection experiment, when the vehicle is halting for 60s closer to the boundary is 6m for achieving 99 percent accuracy. Finally, we also experimentally estimated our GPS errors and found that the median GPS error was 9m.

We are currently extending the concept of Virtual Trip-Box to Virtual Trip-Polygon where the number of edges used to define the boundary for event triggers is unrestricted. The advantage of doing this is that we could define the boundaries with much higher precision. We are also currently integrating the Virtual Trip-Polygon with the ORBIT[15] OMF interface in order to be able to configure the ORBIT mobile nodes for experimental data collection through the ORBIT OMF interface.

References

[1] L. Bauer, L. F. Cranor, M. K. Reiter, and K. Vaniea. Lessons learned from the deployment of a smartphone-based access-control system. In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pages 64–75, New York, NY, USA, 2007. ACM.

[2] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Comput. Commun.*, 31(12):2838–2849, 2008.

[3] G. Chandrasekaran, M. Ergin, J. Yang, S. Liu, Y. Chen, M. Gruteser, and R. Martin. Empirical evaluation of the limits on localization using signal strength. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on*, pages 1–9, June 2009.

[4] M. Y. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky. Practical metropolitan-scale positioning for gsm phones. In *Proceedings of the Eighth International Conference on Ubiquitous Computing (Ubicomp 2006)*, Lecture Notes in Computer Science, pages 225–242. Springer-Verlag, September 2006.

[5] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 29–39, New York, NY, USA, 2008. ACM.

[6] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 15–28, New York, NY, USA, 2008. ACM.

[7] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: a distributed mobile sensor computing system. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 125–138, New York, NY, USA, 2006. ACM.

[8] H. Laitinen, J. Lahteenmaki, and T. Nordstrom. Database correlation method for gsm location. In *Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd*, volume 4, pages 2504–2508 vol.4, 2001.

[9] T.-H. Lin, P. Huang, H.-H. Chu, and C.-W. You. Energy-efficient boundary detection for rf-based localization systems. *IEEE Transactions on Mobile Computing*, 8:29–40, 2008.

[10] K. Lorincz and M. Welsh. Motetrack: A robust, decentralized location tracking system for disaster response, 2004.

[11] S. Mathur, S. Kaul, M. Gruteser, and W. Trappe. Parknet: a mobile sensor network for harvesting real time vehicular parking information. In *MobiHoc S3 '09: Proceedings of the 2009 MobiHoc S3 workshop on MobiHoc S3*, pages 25–28, New York, NY, USA, 2009. ACM.

[12] G. Ostojic, S. Stankovski, M. Lazarevic, and V. Jovanovic. Implementation of rfid technology in parking lot access control system. In *RFID Eurasia, 2007 1st Annual*, pages 1–5, Sept. 2007.

[13] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara. Accurate gsm indoor localization. In *7th International Conference on Ubiquitous Computing (UbiComp)*, Tokyo, Japan, September 2005.

[14] Z. Pala and N. Inanc. Smart parking applications using rfid technology. In *RFID Eurasia, 2007 1st Annual*, pages 1–3, 2007.

[15] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar. Omf: a control and management framework for networking testbeds. *SIGOPS Oper. Syst. Rev.*, 43(4):54–59, 2009.

[16] I. Smith, J. Tabert, T. Wild, A. Lamarca, A. Lamarca, Y. Chawathe, Y. Chawathe, S. Consolvo, S. Consolvo, J. Hightower, J. Hightower, J. Scott, J. Scott, T. Sohn, T. Sohn, J. Howard, J. Howard, J. Hughes, J. Hughes, F. Potter, F. Potter, P. Powledge, P. Powledge, G. Borriello, G. Borriello, B. Schilit, and B. Schilit. Place lab: Device positioning using radio beacons in the wild. In *In Proceedings of the Third International Conference on Pervasive Computing*, pages 116–133. Springer, 2005.

[17] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 85–98, New York, NY, USA, 2009. ACM.

Vita

Janani Chandrasekaran

2010	M.S. in Electrical and Computer Engineering, Rutgers University, New Jersey, USA
2004-08	B.S. in Computer Science and Engineering, B.S.A. Crescent Engineering College, Vandalur, India
2004	Graduated from Modern Senior Secondary School, Nanganallur, India

2008-2009	Part Time Lecturer, Department of Mathematics, Rutgers University