FEASIBILITY OF DUTY CYCLING GPS RECEIVER FOR TRAJECTORY-BASED SERVICES

BY XIAOHAN LI

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

Prof. Janne Lindqvist

and approved by

New Brunswick, New Jersey

May, 2014

ABSTRACT OF THE THESIS

Feasibility of Duty Cycling GPS Receiver for Trajectory-based Services

by XIAOHAN LI

Thesis Director: Prof. Janne Lindqvist

Energy efficient localization is important for lots of smartphone applications. The research community has argued that fixed duty cycling of GPS is not a good choice for trajectory-based services concerning route accuracy. In this note, we show that duty cycling of a smartphone GPS receiver achieves considerable energy efficiency without sacrificing much route accuracy. When increasing sampling period to 120 seconds, it saves at least 78% energy in comparison to continuous GPS sampling, while the loss of route accuracy tends to be stable at 0.23 to 0.25.

Acknowledgements

First and foremost, I would like to thank my advisor Professor Janne Lindqvist, who gave me a lot of advices and instructions during the project, especially when I was stuck with a method. This project is an important research project to me. I did benefit a lot from it, not only academically but also mentally. I could not finish it without all the courage and confidence my advisor gave to me.

I would also like to thank all the other students in our lab. During the project, I have encountered many technical difficulties that could not be solved by myself alone. By talking with other students in the lab, I always found useful and valuable advices to my project.

Last but not the least, I would like to express my gratitude to the program. During the two years in the program, I have met many wise, respected and innovative professors. I appreciate and value all the lectures they have given and all the knowledge they have taught.

This material is based upon work supported by the National Science Foundation under Grant Number 1211079. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Table of Contents

List of Tables

List of Figures

Chapter 1
INTRODUCTION

1.1 Motivation

Location service dominates in mobile applications nowadays. Location information serves as an important context to many mobile applications, thus almost all the smart phones today have integrated GPS chips. Some of the location based applications are satisfied with a single position fix. However, some of them require continuous location traces. For example, applications like navigation cannot navigate based on a single position fix. It requires continuous location traces to construct route and navigate based on the route. Those services are called trajectory-based services. Compared to single location services, trajectory-based services require continuous GPS sensing. GPS sensing is known for its high energy consumption. Usually continuous GPS sensing can drain the battery of a smart phone within 10 hours. Thus, energy efficiency has increasing importance to trajectory-based services for modern mobile device. In order to realize energy efficient localization, the most obvious way is duty cycling GPS receiver instead of performing continuous GPS sampling. However, several works have cast doubt on duty cycling GPS concerning route accuracy or energy efficiency [1–7].

1.2 Background

1.2.1 Location-based Services

Location-based service allows location-based apps to use information on the geographical position of the mobile device. There are mainly three options for a smart phone to determine your approximate location: cellular, Wi-Fi, and Global Positioning System (GPS) networks. Depending on which option your mobile device is using, the precision is different.

- Cellular

  In a cellular radio system, based on radio signal delay of closest cell-phone towers and cell ID, a smart phone can obtain its raw location information.

- Wi-Fi

  Crowdsourced Wi-Fi data can also be used to identify a smart phone's location. Wi-Fi based locating is widely used in the indoor environment when GPS is not available due to satellites invisibility.

- GPS

  Location obtained by GPS is significantly more precise that it plays a critical role in military, civil and commercial users around the world. Trajectory-based services often need to activate continuous GPS sampling to obtain location traces, in order to satisfy its high precision requirement.

1.2.2 Basic Concept of GPS

A GPS receiver calculates its location by satellite position and the precise time that signals transmitted from satellites. It can obtain a position fix only when there is an unobstructed line of sight to at least four GPS satellites. Each satellite continuously transmits messages including the time that the message was transmitted and its satellite position at the time of message transmission. When a GPS receiver receives the messages, it will determine the transit time of each message and computes the distance to each satellite using speed of light. The paper Energy Efficient GPS Sensing with Cloud Offloading [7] explains how a GPS receiver works.

- GPS System

  Currently, the GPS system has 31 active satellites in orbits and one satellite for redundancy. And each orbit is about 20,000 km from the Earth's surface. A satellite can orbit the Earth two cycles a day. A set of ground management stations are used to monitor satellites' orbit and status, and send the satellite data to the satellites. Two types of data included are almanac and the ephemeris.

- Almanac and Ephemeris

  In order to determine the location of the satellites, two types of data are required by the GPS receiver: the almanac and the ephemeris. The satellites continuously transmit these data and the GPS receiver collects and stores these data.

  The almanac contains coarse information about status and orbit of the satellites. The almanac is used by the GPS receiver to calculate which satellites are currently visible. However, it is not precise enough for the GPS receiver to calculate a position fix. Almanac will be stored in the GPS receiver and is considered valid for up to 180 days. If a GPS receiver has not been used for some time, almanac data is not valid anymore, and it may take 15 minutes or so to receive a current almanac. In some old models of GPS receiver, almanac is required to acquire the satellites. But many newer models of GPS receiver are able to acquire the satellites without waiting for the almanac.

  Except for almanac data, a GPS receiver also requires ephemeris data for each satellite, which contains very precise information about the orbit of each satellite. The location of a satellite can be calculated with accuracy of a meter or two by GPS receiver using ephemeris data. The ephemeris is broadcasted by the satellite every 30 seconds and is usually valid up to four hours, according to Navstar GPS User Equipment [8]. If a GPS receiver has been off for a while, it may spend up to several minutes to receive the ephemeris data from each satellite before obtaining a position fix. A GPS packet frame is shown in Figure 1.1. There are five subframes in the packet. Each subframe contains 10 words that each needs 0.6 second to process. Thus it takes 6 seconds to process one subframe, and 30 seconds to decode the whole packet.

- Time to First Fix

  An important concept to understand GPS is Time To First Fix (TTFF). TTFF is the time that a GPS needs to spend to receive satellite signals and data in order to calculate a position fix. According to GPS Receiver Testing [9], TTFF is commonly broken down into three scenarios depending on GPS start up mode:

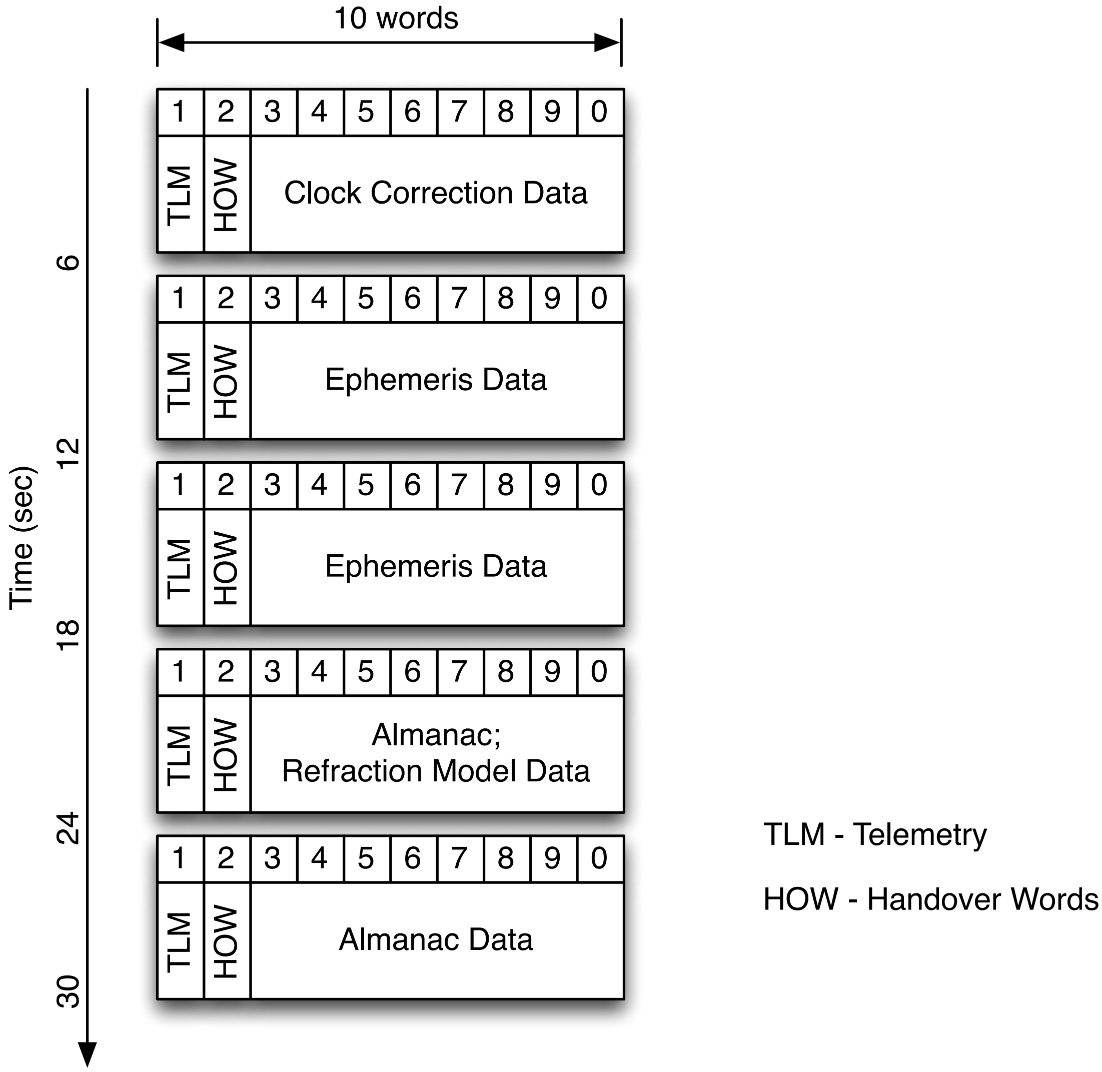


Figure 1.1: GPS packet frame

- Cold Start

  If a GPS receiver has no prior knowledge of its position, velocity, the time, or the visibility of any of the GPS satellites, it has to systematically search the entire space for all possible satellites. After acquiring a satellite signal, the receiver can begin to obtain almanac on all the other satellites. In this scenario, GPS receiver may take a few second to acquire one satellite. And it also needs to decode the whole packet shown in Figure 1.1, which takes 30 seconds. This is one of the main reasons why it takes longer time to obtain initial position fix and consumes higher energy.

- Warm Start

  If a GPS receiver knows the time within 20 seconds, the current position within 100 kilometers, its velocity within 25 m/s, and it has valid almanac data, then it can skip almanac acquisition and start from acquiring ephemeris data.

- Hot Start

  If a GPS receiver has valid time, position, almanac and ephemeris data, it then can enable a hot start. In this scenario, the GPS receiver can skip the acquisition and only need to obtain timing information from each satellite. The time spent to calculate a position fix in this scenario may also be termed as Time to Subsequent Fix (TTSF). For most modern GPS receivers, TTSF is usually within 0.5 to 20 seconds.

1.2.3 Assisted GPS

Assisted GPS (A-GPS) is one of multiple ways to improve TTFF. For example, in the Mobile-Station Based A-GPS mode, the infrastructure can supply ephemeris data so that the GPS receiver does not have to decode them from the satellites signals.

1.2.4 Duty Cycling GPS

For smart phones, continuous GPS sensing means to sample at 1 Hz, which is very energy consuming. The most obvious way for energy efficient localization is to reduce the time that GPS receiver is on. Usually for single location service, it activates GPS receiver only when location information is required and then turn off the GPS receiver right after the position fix has been obtained. For trajectory-based service, it requires continuous location traces, which means the GPS receiver should be on and sampling all the time. Thus, trajectory-based services are much more energy consuming. However, by observing GPS traces, we found that sampling at 1 Hz was unnecessary even for trajectory-based services. With the knowledge of road information, we are still able to construct the route from a GPS trace even when the GPS sample rate is inadequate. Thus, duty cycling GPS is a feasible approach for efficient trajectory-based services. The problem is to find out the trade-off between the accuracy of routes constructed from GPS traces with different sample rates and energy consumption. Thus after balancing the pros and cons, we can formulate the most energy efficient period of duty cycling GPS, meanwhile guarantee high accuracy of routes constructed from the traces.

1.2.5 Map Matching

Map matching is the procedure of aligning a sequence of observed user positions with the road network on a digital map. The most obvious algorithm is that simply matching each location sample of GPS trace with the nearest node of map data. Each road in the map provided by OpenStreetMap is represented by a series of nodes. Due to measurement noise and the density of the map data, this procedure is prone to error. The paper Hidden Markov Map Matching Through Noise and Sparseness [10] describes a novel, principled map matching algorithm that uses a Hidden Markov Model (HMM) to match location samples from GPS traces onto map data. In this project, I propose another algorithm for map matching.

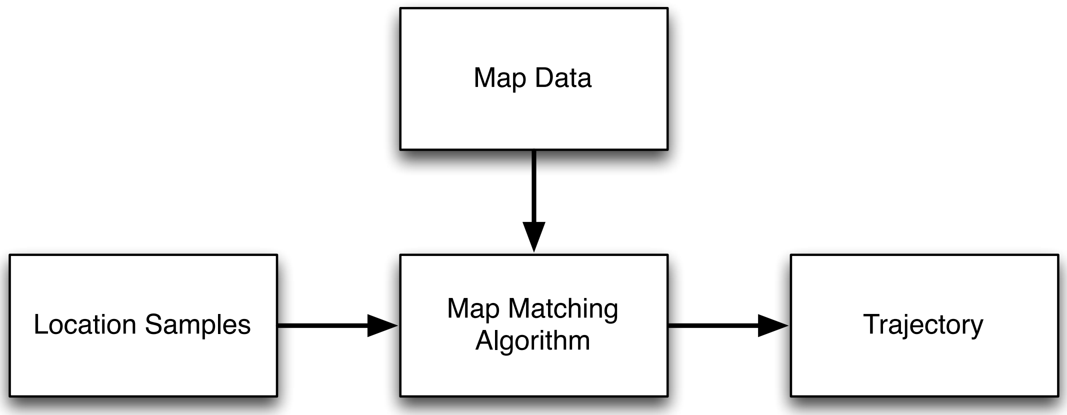


Figure 1.2: Map matching process

1.2.6 Path Constructing

For trajectory-based services, the most important part is to construct the path from the GPS traces. Path constructing is the procedure that determines the roads vehicle has been on according to the data collected by GPS device. The accuracy of path constructed depends on three elements. First, the accuracy of map matching provides the foundation for path constructing. Second, the sample rate of GPS trace determines the density of trace. Trace with high sample rate provides more information to construct the path. Third, the algorithm that we use to construct the path can also determine the accuracy. This paper demonstrates how to construct the path of GPS traces by applying road network information provided by OpenStreetMap in order to find trade-off between path constructing accuracy and sample rate of GPS trace. By analyzing energy consumption of GPS device, we can therefore be able to find the trade-off between accuracy and energy consumption. To construct a path, there are two steps. First, map matching procedure matches location samples of GPS traces onto map data provided by OpenStreetMap. Second, path exploration will be based on the result of map matching. If samples from the GPS trace provide adequate information, meaning only one legal path between every successive two samples, then the path can be obtained by simply connecting these two samples. Otherwise, path exploration can be done by querying the road network information provided by OpenStreetMap and by applying some restrictions to find the most likely path. The restrictions included are no U turn or right/left turn on certain road, one way, impossibility to make a turn under high speed limit, etc. In this case of path exploration, additional road network information provided by OpenStreetMap will make up for the inadequate GPS samples to construct the path. This paper focuses on the path exploration method.

Chapter 2

RELATED WORK

Previous researches have made many approaches on energy efficient localization. Those approaches include trade-off between positioning accuracy and adaptive duty cycling GPS; using low power sensors to aid GPS sampling; making up GPS sampling insufficiency by using history location information etc. Most past research papers have made a point that fixed duty cycling GPS receiver is not a good choice for trajectory-based services.

LEAP: a low energy assisted GPS for trajectory-based services [6] explains why it is hard to realize duty cycling GPS receiver for trajectory-based services. The GPS processing contains four stages: acquisition, tracking, decoding and position calculation. The GPS receiver is in charge of the first three stages. And the position calculation is processed in main processor of the device. When the GPS receiver starts up, it acquires and receives the data transmitted in by GPS satellites. Once the signals are acquired, the receiver enters the second stage, i.e. tracking. In this stage, it runs continuously to keep feedback loops with satellites. To calculate a position fix, the receiver must track time to microsecond level. The millisecond part is decoded in the decoding stage. The sub-millisecond part called code phase is computed by using correction in the tracking stage. Thus the receiver needs to maintain code phase sync with satellites. Then the receiver will go to decoding stage with correct tracking. In this stage, it decodes the packets sent by the satellites. During the decoding, the tracking stage still keeps running. If the tracking component of GPS shuts down, it takes time to search and reacquire connection with satellites, forming a time gap that can cause severe code phase error. In order to be energy efficient, LEAP off-loads packet decoding and location calculation to the cloud. And further more, by introducing a mechanism for fast re-acquisition based on previous tracking results, LEAP realizes duty-cycle tracking loops of GPS receiver. Hence, LEAP has made a good performance for energy consumption on trajectory-based services.

Energy-efficient localization: GPS duty cycling with radio ranging [3] makes a point that duty cycling the GPS module can prolong the device's battery life at the cost of increased position uncertainty while the GPS is off. The paper analyzes the relationship among energy, GPS uncertainty, GPS off-time and speed. It then proposes three speed models for duty cycling strategies for maintaining position uncertainty within specified bounds. Static model is based on a constant assumed speed. Dynamic model is based on setting the assumed speed as the last observed speed of mobile node. Probabilistic model is based on last observed speed and a state model of the mobile node. According to the required uncertainty bound of certain application and speed, the system can decide how to schedule duty cycling.

Improving energy efficiency of location sensing on smartphones [5] talks about RAPS, rate-adaptive positioning system, based on the approach of duty-cycling GPS. It uses a collection of techniques to cleverly determine when to turn on GPS. It takes three elements into account when deciding whether to turn off GPS. First, it uses location-time history to estimate user velocity and adaptively turn on GPS according to the uncertainty. Second, it estimates user movement using a duty-cycled accelerometer and Bluetooth to reduce position uncertainty among neighboring devices. Third, cell tower-RSS blacklisting is used to detect GPS unavailability. RAPS achieves much of its energy saving by avoiding GPS activation in the places not available. However, when it comes to driving on the roads that GPS service is always available and necessary, it does not save much energy.

Energy-accuracy trade-off for continuous mobile device location [11] proposes an approach that is similar to the one above. Their goal was to develop location as a system service that automatically manages location sensor availability, accuracy and energy. The approach is based on two observations. First, location applications do not always need highest available accuracy. Second, a phone has multiple modalities to sense location aside from GPS. The paper saves energy by determining the most energy efficient sensor to be used, such that the required location accuracy can be achieved. In addition, before spending energy on sensing at the current time step, the paper uses Hidden Markov Model to provide a probability distribution of predicted location.

Energy-efficient positioning for smartphones using Cell-ID sequence matching [1] comes up an approach to save energy by using a cell-ID sequence matching technique to estimate current position based on the history of cell-ID and GPS position sequences that match the current cell-ID sequence. Obtaining current location information by cell tower uses much less power than by GPS. However because the uncertainty of using cell tower is much higher, it is barely used in obtaining precise location. By adding additional information to the location information obtained by cell tower, it is possible to gain current position with relatively high accuracy while saving energy. When locating current position, the system designed has three salient features: Spatial and temporal mobility history, cell-ID sequence matching and opportunistic learning. The paper reveals a fact that people often take similar routes in daily life. The cell-ID sequence and GPS coordinates are stored in the database as history route. If the current cell-ID matches a sequence or a sub-sequence in the database, it estimates the user position within the route traveled in the past. If it is not, the system will turn on the GPS to opportunistically learn and build the history of route for future usage. The system has two main limitations. First, it requires storage for route of history. Second, if the user always explores new route instead of having a travel pattern, the system cannot save much energy as it always has to turn on GPS every time to learn a new route.

SensLoc: sensing everyday places and paths using less energy [12] has made an attempt to efficiently provide contextual information about locations as places and paths instead of simply tracking a user's raw coordinates. The paper proposes a new abstraction of continuous location: places and paths. It also presents a framework that provides location context as places and paths using less energy. The system architecture contains three parts: place detector, movement detector and path tracker. The place detector can learn a new place by saving its place signature, recognize the place by regularly scanning neighboring radio beacon and consulting database, triggering the movement detector to find an opportunity to sleep. Path tracker is only enabled when the place detector senses a place departure. The system saves most of energy by context detection, detecting places and only enabling GPS on these paths.

EnTracked: energy-efficient robust position tracking for mobile devices [4] proposes EnTrack, a system that is based on the estimation and prediction of system conditions and mobility. It can schedule position updates to both minimize energy consumption and optimize robustness. First, the paper analyzes the power consumption model, giving us the relationship of power consumption and five power parameters instead of simply assuming that power consumption for position sensing and sending is instantaneous. Second, by detecting movement and estimating speed, the paper proposes an error model with two parameters: the estimated uncertainty of the last GPS position delivered to the application, the time since the last GPS position and the estimated speed. Then based on the application-defined error limit, the current error and the estimated speed, the system calculates the time limit for the next GPS position. Taking the power consumption and the time limit into account, the paper formulates an equation to minimize power consumption. The main limitation of EnTrack is that the experiment is conducted by tracking the pedestrian target with maximum speed of 10m/s, which is not suitable when it comes to driving scenario.

Energy-efficient trajectory tracking for mobile devices [13] proposes an on-device sensor management strategy and a set of trajectory updating protocols which cleverly determine when to sample different sensor (accelerometer, compass and GPS) and when data should be simplified and sent to a remote server. The framework the paper proposes is an extension of EnTrack. The sensor management strategy is compass-based change-of-direction sensing and adaptive duty cycling accelerometer and compass sensors. Trajectory simplification algorithm is designed for energy-efficient trajectory update protocols. The paper has made great progress after EnTrack that it can be used for different transportation modes other than pedestrian.

Exploiting temporal stability and low-rank structure for localization in mobile networks [14] focuses on localization in mobile networks instead of using GPS for location determination in order to be energy efficient. It proposes three schemes to accurately determine locations in mobile networks: Low Rank based Localization, Temporal Stability based Localization and Temporal Stability and Low Rank based Localization. According to the paper, GPS for localization is not the only option to accurately determine locations. However, it does not analyze energy consumption for those schemes.

Chapter 3

EXPERIMENTAL METHODOLOGY

3.1 Experimental Goal

This study was performed with two main goals in mind. The first goal is to formulate the relationship between the accuracy of path constructed by GPS traces and the sample period of the GPS traces. To do so, we need to collect a large amount of GPS traces and reduce the sample rate of the traces gradually. And then by applying our algorithms, construct the routes from those GPS traces and quantify the accuracy of the routes constructed from traces with different sample rates. After a large amount of experiments, we can formulate a general result. The second goal is to analyze energy consumption under different sample rates. The energy analysis in combination with the result of route accuracy gives us a clue of whether duty cycling GPS is a feasible approach for energy efficient trajectory-based services.

3.2 Experimental Design

To realize the study goal, we designed our experiment into 5 stages.

- Data Collection: Collecting real-life driving GPS traces is the first step of our project, since it uses a lot of GPS traces. In order to do so, each participant carries an Android phone with an application that can activate GPS location service and store traces into local database during driving. The traces collected are all under normal driving circumstances, with no traffic violation. And the routes we chose are distributed in different area of Central Jersey and contain both highways and local roads.

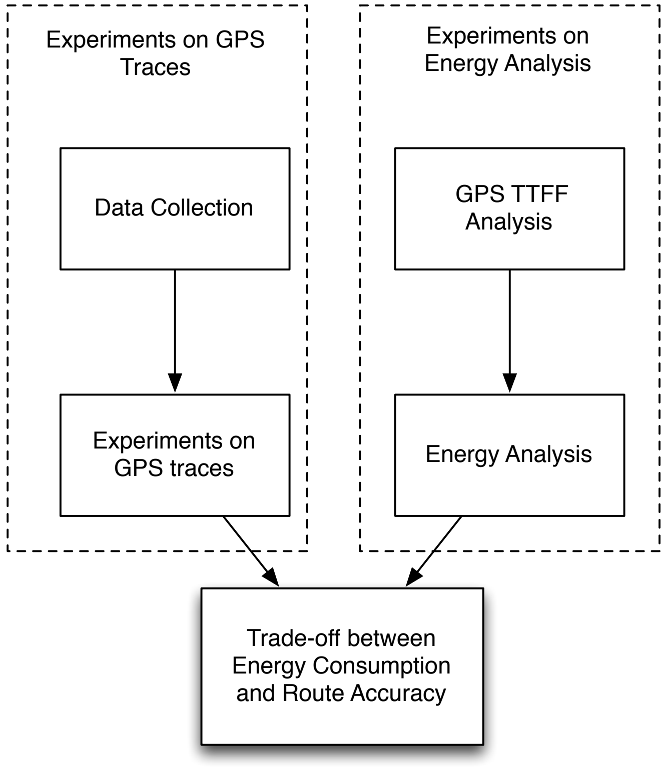


Figure 3.1: We designed our experiment into 5 stages

- Experiments on GPS Traces: The data we collected are time stamped latitude/longitude coordinates. And the original sample rate of those traces is 1 Hz. To simulate reducing sample rate, we removed samples in those traces gradually. For example, to simulate 0.1 Hz sample rate, all we need to do is to select samples with interval of 10 seconds and remove all the other samples between them. After simulating reducing sample rate of each trace, we have traces with different sample rates. We then construct routes from these traces, compare them with original route and quantify their accuracy. After a large amount of experiments on those traces, we can then formulate our result, as indicated in Figure 3.2.

- GPS Time To First Fix (TTFF) Analysis: Real duty-cycling GPS working process is not exactly like our simulation. Because it takes time for GPS to get a position fix after acquiring satellite signals, as introduced in Chapter 1, we need to consider GPS TTFF into the result we get from the last step. However, GPS TTFF is not a fixed value. It depends on the weather condition, the number of visible

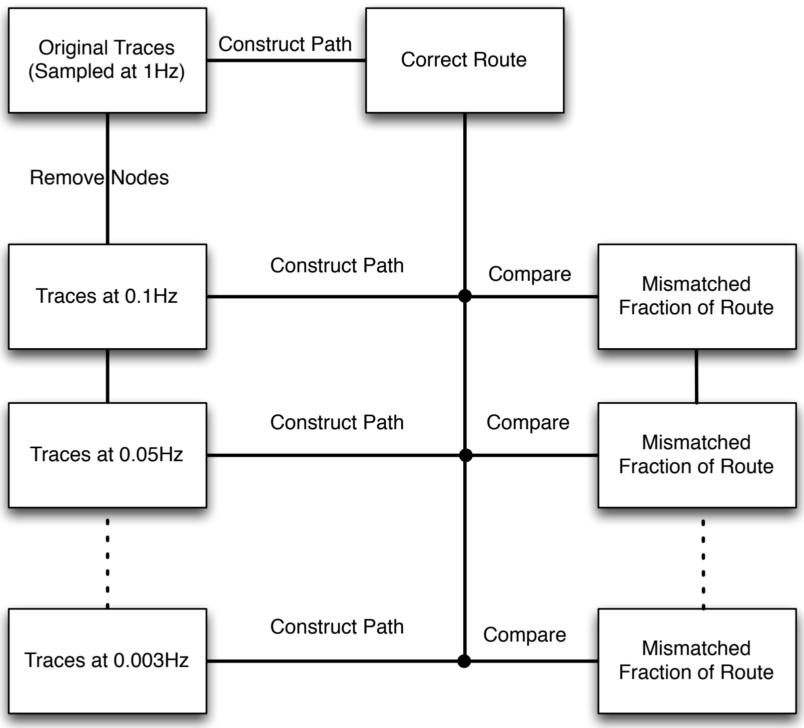


Figure 3.2: Experiments on GPS traces

satellites, signal strength, driving speed, etc. In order to have a general clue of TTFF, we test it by activating GPS receiver and timing its TTFF on the real driving circumstances under different weathers and different roads.

- GPS Energy Analysis: To do energy analysis, we activate GPS component of the smartphone under the condition that satellites are visible and then we monitor the electric current going through the smartphone's battery using an oscilloscope. We then can get a clue of how much energy GPS receiver consumes on different working stages. The data we get from this part can help us understand the trade-off between energy consumption and accuracy in our approach.

Chapter 4

ALGORITHMS AND IMPLEMENTATIONS

4.1 Parse the Road Information

Trajectory-based services are usually associated with road information or map. GPS receiver alone can only obtain its latitude/longitude coordinates but not be able to use that information. Only associated with road information or map, the latitude/longitude can be used for applications like navigation, traffic and advertising, etc. Thus, it is inevitable to use map when requiring trajectory-based services. To construct path of the GPS trace, we need to know the road information. The road information we used for this project is provided by OpenStreetMap. OpenStreetMap (OSM) is a collaborative project to create a free editable map of the whole world. Due to the emphasis of local knowledge and ground truth in the process of data collection, the project has a geographically diverse user-base. The density of map data of OpenStreetMap varies from area to area, as indicated in Figure 4.1. The map database downloaded from OpenStreetMap contains the following tables.

- Bounds: This table contains four elements: minimum latitude/longitude and maximum latitude/longitude. It sets up the boundaries of the map we want to use. OpenStreetMap contains map all around the world. We need to set the boundaries of the map more specifically according to the area that the GPS trace covers.

- Nodes: The map data provided by OpenStreetMap consists of many nodes. Each node in this table has its unique node ID and latitude/longitude coordinates.

- Ways: All the road information within the area is stored in this table. Each way has its unique way ID, road name, its type and node IDs of all the nodes on it.

Figure 4.1: OpenStreetMap GPS trace density

The types of way are primary, residential, secondary, motorway, etc. Each way type has its own speed limit.

- Speed Type: Depending on its type, each road has its speed limit which is stored in the speed type table. There are three columns in this table: road type, minimum speed for road type, maximum speed for road type. The unit of speed used here is miles per hour. For example, the minimum speed for motorway is 55 mi/hr and the maximum speed is 65 mi/hr.

Way ID	Name	Type	Node ID
5670031	Staff Street	residential	42425780
5670031	Staff Street	residential	60916236
5670031	Staff Street	residential	60916237
5670031	Staff Street	residential	42431666
5670088	West 167th Street	residential	42432218
5670088	West 167th Street	residential	42432220
5670088	West 167th Street	residential	42432223
5670131	Dyckman Street	secondary	42427859
5670131	Dyckman Street	secondary	42432789

Table 4.1: An example of Ways table

Type	min speed (mi/hr)	max speed (mi/hr)
motorway	55	65
motorway_junction	55	65
motorway_link	25	35
trunk	45	55
trunk_link	25	35
primary	40	50
primary_link	25	35
secondary	35	45
secondary_link	25	35
tertiary	30	40
tertiary_link	25	25
residential	25	25
service	5	25
construction	25	45

Table 4.2: Speed Type

- Lanes: Each way may contain more than one lane. For example, a highway usually contains at least two lanes, one lane to the north and one lane to the south or one lane to the east and one lane to the west. Driver cannot change directly to the opposite direction. Lanes table contains 2 columns: way ID and the number of lanes of this road.

- Adjacencies: Road is represented by a series of connected nodes. Adjacencies table stores the adjacency relationship between nodes on a road. There are three columns in this table: way ID, from_node ID and to_node ID. When we explore ways from one node to the other node, we need to consider the adjacency relationship among the nodes.

4.2 Map Matching

4.2.1 Features of Map Data

The features of map data provided by OpenStreetMap cause the difficulty to match the samples from GPS trace onto the right road. In the database, each road is represented by a series of connected nodes. Figure 4.2 below shows all the nodes along the route in the database. As Figure 4.2 indicates, the distances between two connected nodes in

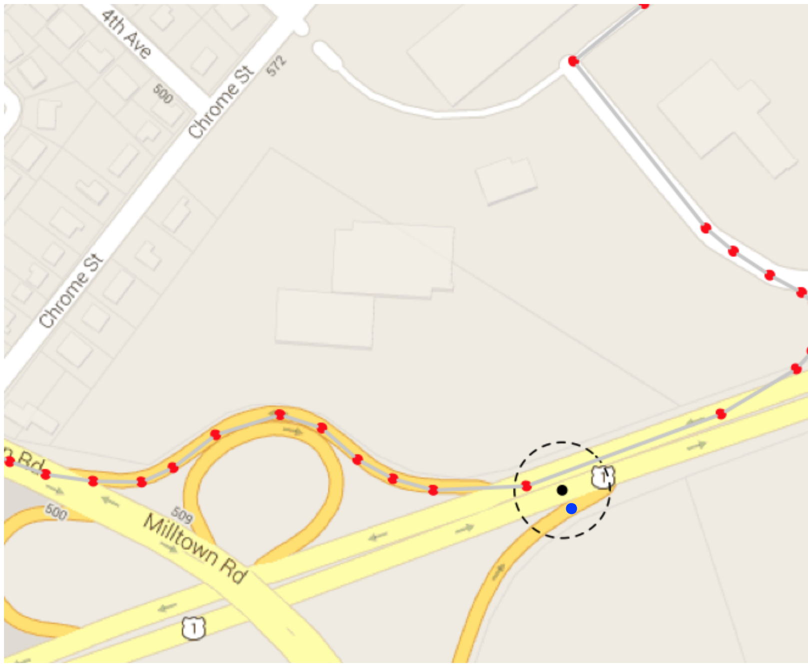


Figure 4.2: Difficulty of mapping caused by two elements: noise of GPS device and uneven distribution of nodes in the database

the map data are different. The database represents straight road by relatively small density of nodes. However, for a curvy road, there are relatively more nodes to represent it in the database.

4.2.2 Difficulty of Map Matching

The difficulty of map matching is caused by the noise of GPS device and the uneven distribution of nodes in the database. As Figure 4.2 indicates, the red colored nodes are all the nodes along the route in the database. The correct route is represented by simply connecting those red nodes with straight line. The black node is one of samples from the GPS trace. Our task is to match the black node onto one of the nodes stored in the database. The blue node is a node in the database that is close to the back node but not on the correct route. The obvious algorithm for map matching is to match the GPS trace to its nearest node in the database. However, as we can see, due to measurement noise and uneven distribution of the database, the nearest node in the database to the black node is the blue one, which obviously does not belong to the correct route.

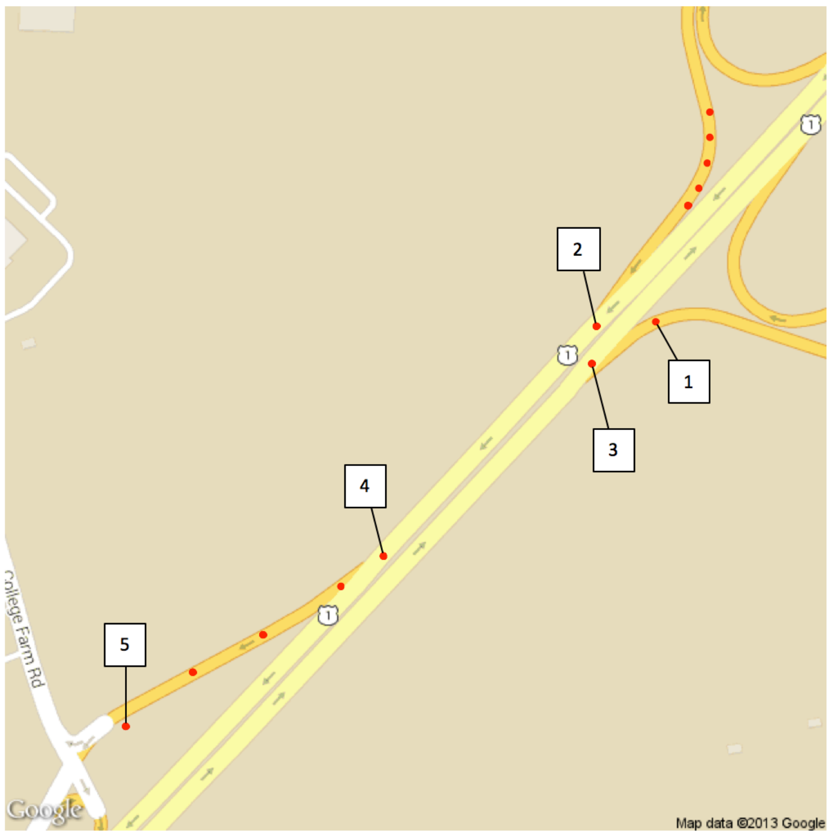


Figure 4.3: Example of bad mapping

4.2.3 Map Matching Error

There are two scenarios of map matching error as indicated in Figure 4.3. Either of them can cause severe problems when it comes to constructing path.

- Map Matching Onto a Wrong Road

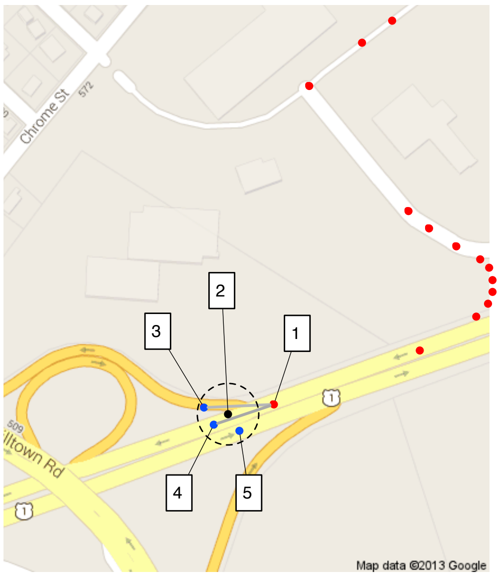
  Map matching onto a wrong road could cause path constructing error. For example, in Figure 4.3, node 1 and node 5 are matched onto a wrong road. For this scenario, we can apply the relationship among samples in the GPS trace as a restriction to do a better map matching. To map match each sample in the trace, we not only get the nearest node on the road but get a range of possible nodes near it and sort them by the distance to the sample that we want to match. Then we check its succeeding sample. We also get a range of candidates for the succeeding sample and sort them. Check whether there is one among the former candidates that shares same way ID with the one among candidates of the succeeding sample. If so, we match the former sample on the road that the way ID represents. If there is no such same way ID, meaning the node is not on the same road with its succeeding node, then it must share the same way ID with the node ahead of it. Apply with the former way ID, and then select the candidate on the road that the former way ID represents. This map matching algorithm proves to be effective in some cases. However, when it comes to the scenario below, it is prone to error.

- Map Matching Onto a Right Road But a Wrong Lane

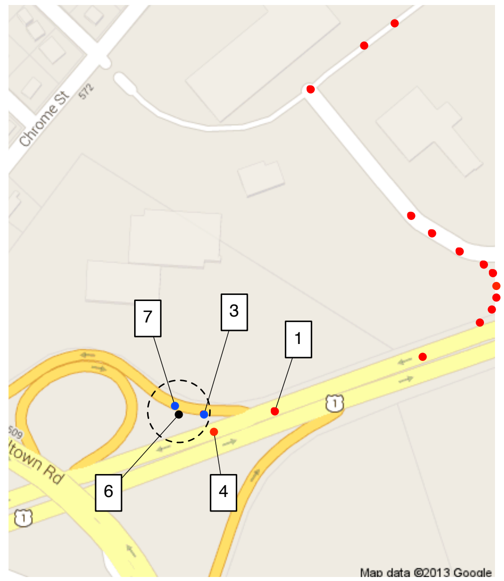
  When a road has lanes with different directions, for example, one lane to the north, the other to the south, we cannot distinguish the lanes since there is no lane ID to distinguish different lanes of the road in the database. If we match the sample onto the right road but a wrong lane, it will explore error when constructing the path. Because it is illegal to drive from one lane and directly turn to the lane with opposite direction if U turn is not allowed, we cannot connect them with a legal path. As illustrated in Figure 4.3, node 3 is matched onto the right road but a wrong lane. When constructing the path, we will find node 3 can neither be connected from node 2 nor connected to node 4. For this scenario, there is no way but checking the connectivity from the former node to the node that we are matching. First, we query the database to get a range of candidates connected to the former node. And then we get a range of candidates for the node that we are matching and sort them according to the distance. Check whether each candidate is connected to the former node according to the order of its distance to the sample in GPS trace. If it is connected, match it, otherwise, check next candidate. Due to lacking way ID information, this algorithm sometimes can also cause map matching error.

4.2.4 Backtracking and Connectivity Check Map Matching

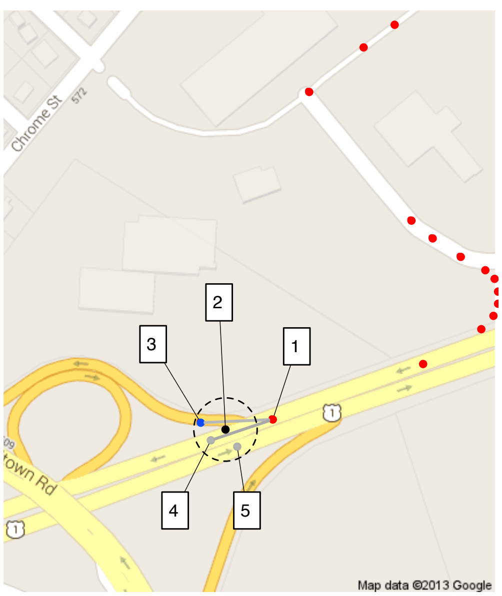
For this project, in order to construct the path and compare the path we construct with the correct route, we need to improve the accuracy of map matching algorithm. As explained in Figure 4.4a, the red nodes are the nodes we have already matched. The black node is a sample from the GPS trace that we want to match onto the map data. There are two conditions that need to be satisfied when matching the black node onto a certain node in the database: the candidates we select must be within a certain range of the black node; the candidates must be reachable by the nodes that have been already matched which are colored by red. So first, we get a range of candidates in the database that are close to the black node and color them by blue. And then we check the connectivity from node 1 to each of the blue nodes. Both node 3 and node 4 are connected to node 1. Because node 4 is closer, we match the black node onto node 4. For this step, even though node 4 is closer to node 2, we know from Figure 4.2 that node 3 is the right choice. We can modify that by backtracking. Since black node 2 has already been matched, we are going to match next sample from the trace which is node 6, as indicated in Figure 4.4b. We get a range of candidates for it and check the connectivity from node 4 to them. Then an error arises that none of the candidates can be reached from node 4. Thus we know it is a wrong decision to match node 2 onto node 4. We must go back to rematch node 2, as indicated in Figure 4.4c. Because we have failed with matching node 2 onto node 4, as indicated in Figure 4.4c, unqualified



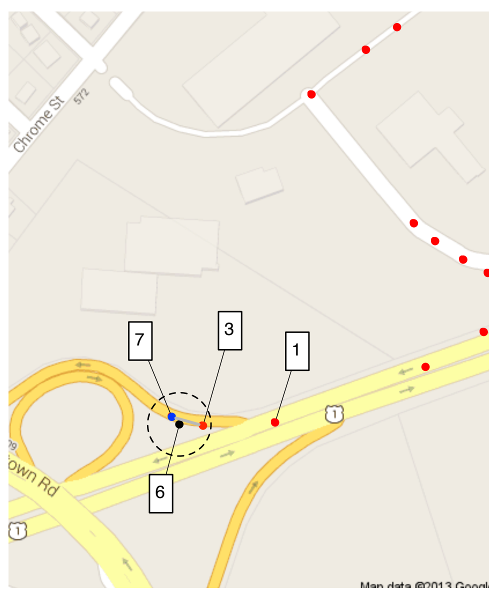
(a) Get a range of candidates and check connectivity



(b) An error happens: none of the candidates has connectivity from node 4



(c) Backtracking and remap node 2



(d) Proceed to map node 6

Figure 4.4: Map Matching process

candidates being marked by grey, node 3 is the only candidate left. Therefore we match node 2 onto node 3. Then we can continue to match node 6 in Figure 4.4d. This time we can tell that candidate node 7 satisfies the two conditions, so match node 6 onto node 7. Because of the backtracking and connectivity check mechanism, if a map matching error happens, we can always detect it and backtrack to correct it. This map matching algorithm achieves much better accuracy.

4.2.5 Limitations of Map Matching

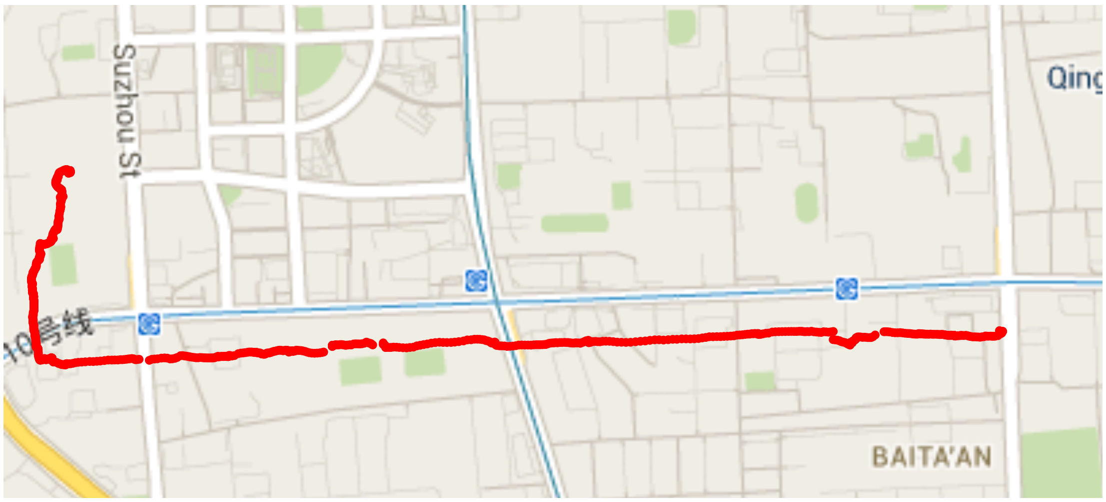
The map matching algorithm is dependent on the map data provided by OpenStreetMap, which is uploaded by users around the world. The density of map data provided by OpenStreetMap varies from area to area. Thus, in certain area, the density of nodes in the database is inadequate due to infrequency of GPS trace uploading. Figure 4.5a below is a GPS trace collected in Beijing, China. There are 1,270 nodes in the original trace. However, when map matching it onto the map data provided by OpenStreetMap as indicated in Figure 4.5b, the route is represented by only 15 sample nodes in the database. For this area, the density of map data is inadequate compared to the complexity and density of road network.

4.3 The Path Constructing Problem

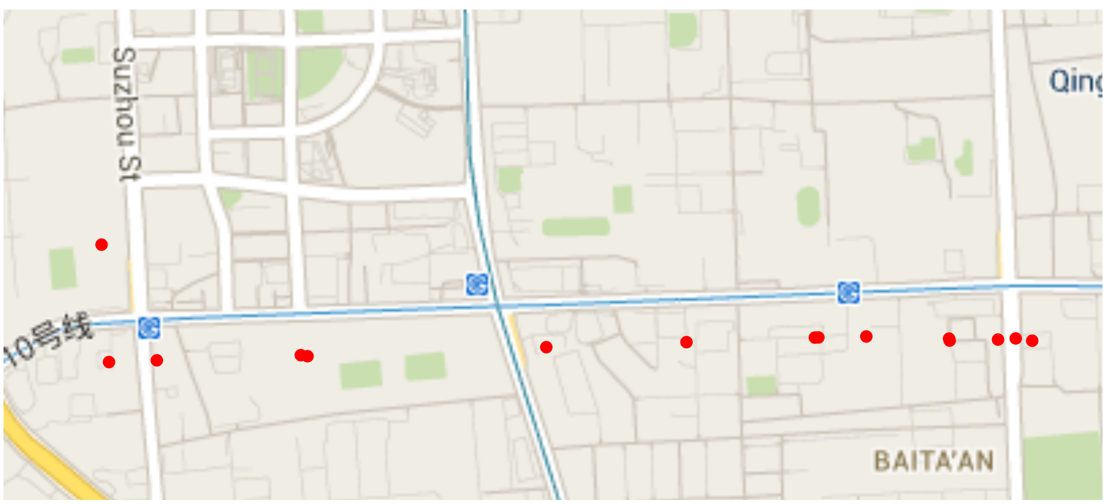
Each node stored in the database provided by OpenStreetMap has attributes such as, a unique node ID, latitude/longitude coordinates and way ID indicating which road the node belongs to. A node may have several way IDs.

4.3.1 Path Constructing Scenarios

Path constructing problem contains two scenarios. The first one, as indicated in Figure 4.6a, is when the two nodes have the same way ID. To this scenario, we query the database and then simply connect the two nodes by adding points between them along the road as illustrated in Figure 4.6b. The number of points we add between the two nodes depends on the density of nodes along the road in the database. The

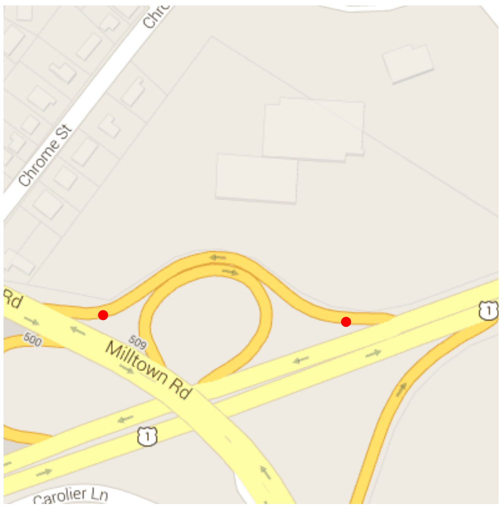


(a) A GPS trace collected in Beijing, China

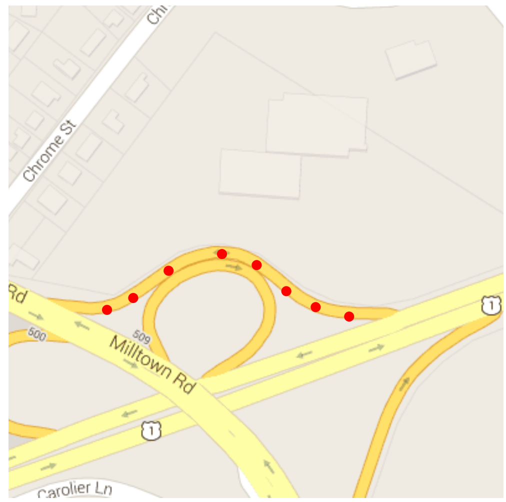


(b) Same route represented by the sample nodes in the database

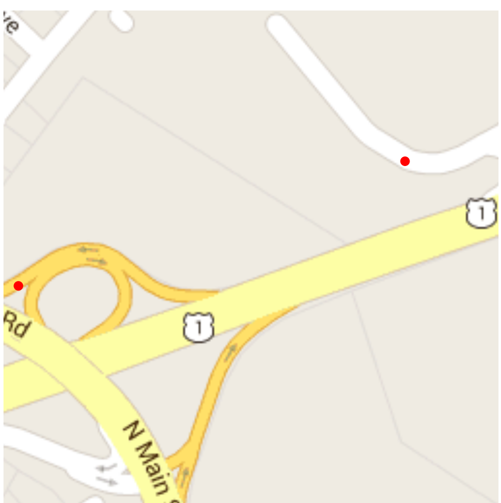
Figure 4.5: Inadequate sample nodes in Open Street Maps



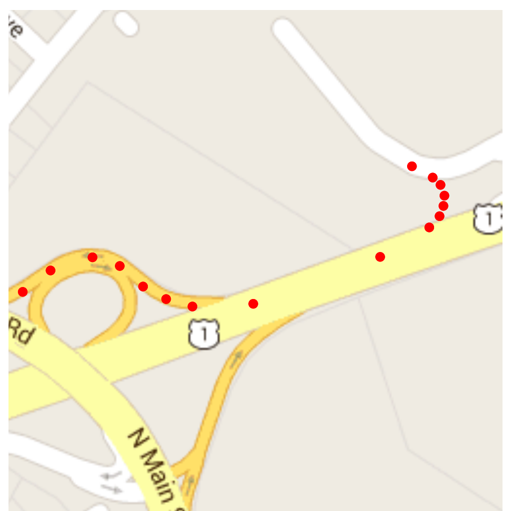
(a) Two nodes have same way ID, meaning they are the same road



(b) Connect the two nodes by adding points between them along the road



(c) Two nodes are on the different roads



(d) Explore path by inquiry database to find intersection of the two roads

Figure 4.6: Map matching process

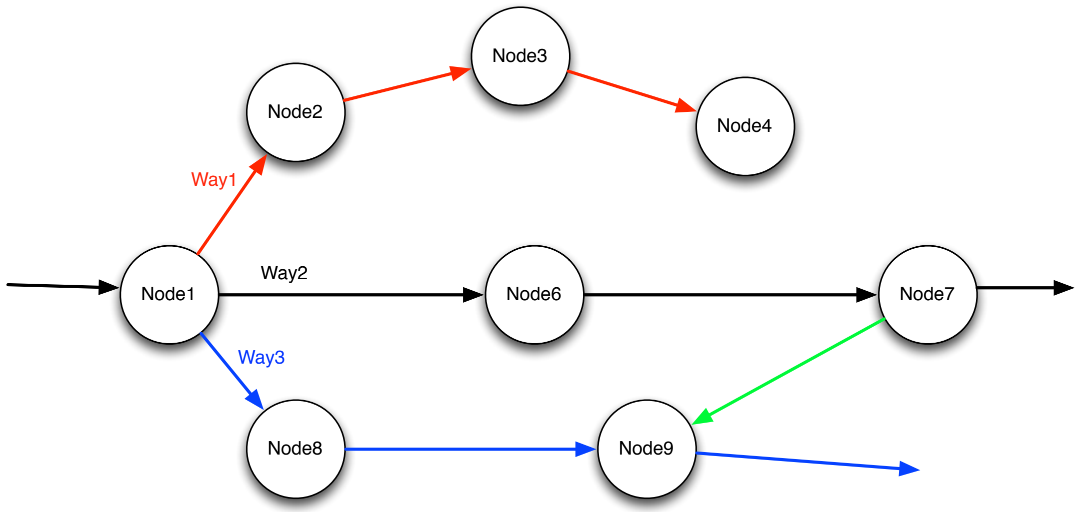


Figure 4.7: Breadth First Search to explore path between nodes that are not on the same road

second scenario is when the two nodes do not have a common way ID, as indicated in Figure 4.6c. To construct the path under this scenario, we query the database to find all the nodes connected to the first node and use restrictions to eliminate the impossible nodes. Then use modified Breadth First Search algorithm to explore all the possible nodes until we find the path connected to the second node.

4.3.2 Exploring Possible Nodes

To construct the path under the second scenario, we want to find the possible path between two nodes with different way IDs. To explore the path between the two nodes, we use modified Breadth First Search algorithm. From the starting node, push all the possible next nodes adjacent to it into a queue. Explore each node in the queue. Check whether the node popped out from the queue matches the destination node. If it matches, stop exploring and construct the path from the starting node to the destination node. Otherwise keep exploring until find the destination node in the queue. For example, in Figure 4.7, we want to explore the path from node 1 to node 9. We can see there are two paths form node 1 to node 9: node1-node 6-node7-node 9 and node 1-node 8-node 9. Clearly node 1 is the intercourse of three different roads: way 1, way 2 and way 3. Each way is colored differently. Starting from node 1, we query the database and find its adjacent nodes, node 2, node 6 and node 8. Because we have not encountered our destination node, i.e. node 9, we need to keep exploring from these nodes. Starting from node 2, we find its successor node 3. Node 9 is still not met, so we keep exploring from node 6 and then node 8. We find node 8's successor is node 9. We then stop and way 3 is our choice. This algorithm tends to find straight and shortest way between two nodes, which is similar to our driving habit. When driving from one location to the other location, people tend to choose the shortest way.

Chapter 5

EXPERIMENTS

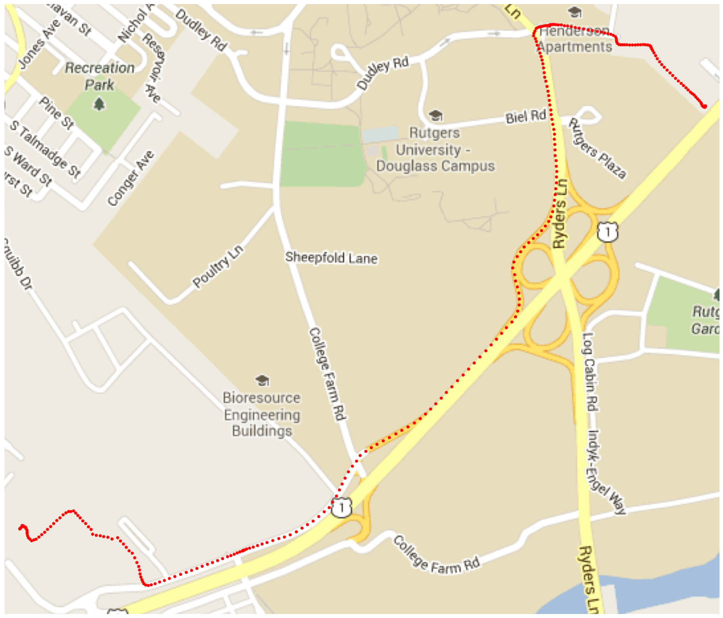
5.1 Experiments on GPS Traces

5.1.1 GPS Traces

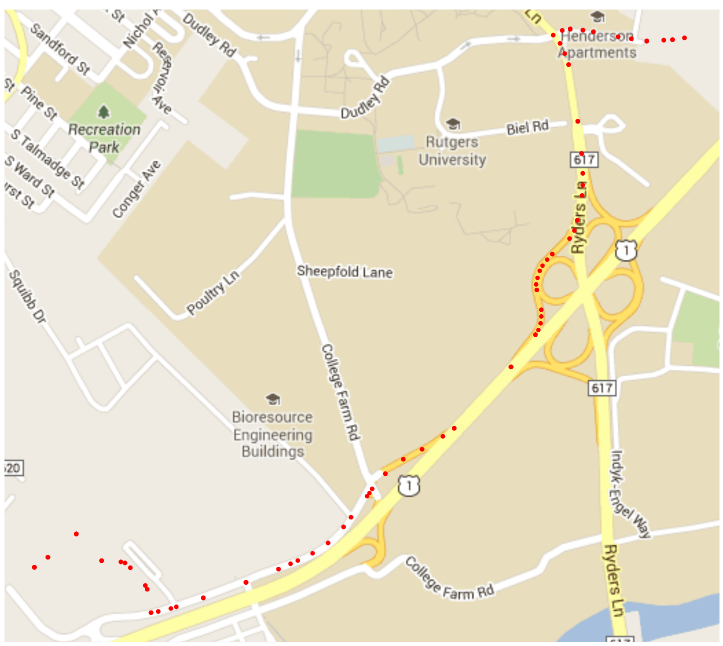
We tested our theory using the data collected in driving cars. We have collected 123 traces with different route complexity and road network density in Central Jersey. The total distance of the traces is 1188.88 miles. The longest trace is 36.55 miles long. The original sample rate is 1 Hz. The shortest trace is 1.78 miles. An example of the traces we used is as shown in Figure 5.1a. This route is 2.886 miles long. The number of latitude/longitude coordinates is 327. After map matching the trace onto the map data provided by OpenStreetMap and filtering unqualified nodes, only 66 nodes have been left, as shown in Figure 5.1b. We simulated reducing sample rate of the GPS data by removing points. The sample period was gradually increased to 10, 20, 30, 40, 50, 60, 90, 120, 240, 300 seconds.

5.1.2 Accuracy Evaluation

First, we ran our program on the trace with sample period of 10 seconds. The path constructed from it would be considered as the correct route when evaluating the route accuracy. After running our program on traces with longer sample periods, we qualified the accuracy of the routes constructed from those traces by comparing with the correct route. Figure 5.2 explains how we estimated the accuracy by the percentage of mismatched fraction of route.



(a) An example of the traces we use



(b) Same route represented by map data

Figure 5.1: An example of GPS traces

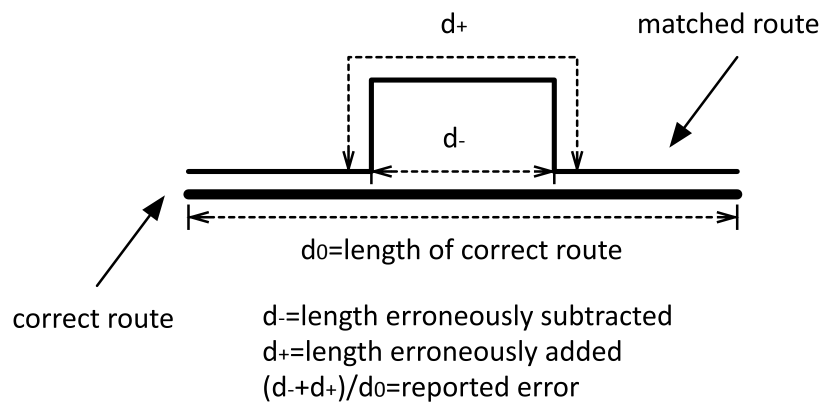


Figure 5.2: This illustrates how we measured the error between the correct route and the route constructed from GPS traces

5.1.3 Results

After running our program on all 123 traces, we have all the results. For GPS traces, because of different route complexity, the results vary from each others. Table 5.1 is a part of the result of the trace shown in Figure 5.1.

Sample Period (second)	d0(mile)	d+(mile)	d-(mile)	Mismatch Fraction
20	2.886	0	0	0
30	2.886	0.876	0.280	0.40
40	2.886	0.876	0.280	0.40
50	2.886	0	0	0
60	2.886	0.876	0.280	0.40

Table 5.1: Result of a single trace

After having tested on all 123 traces that we have collected, we got a chart of how the mismatched fraction of route changed when constructing route from GPS traces with different sample periods. The total distance of the traces is 1188.88 miles. And the result is as shown in Figure 5.3 and Table 5.2.

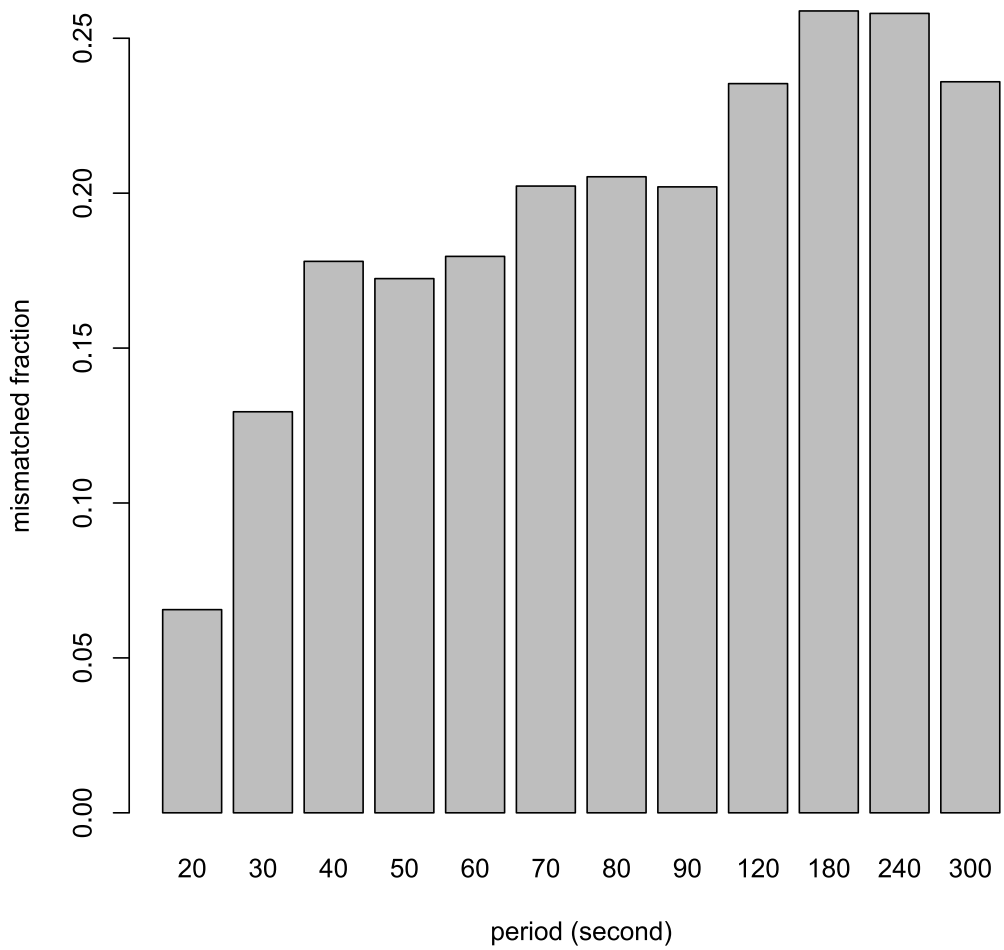


Figure 5.3: The result of how the mismatched fraction of route changes as sample period increases

Sample Period (second)	Mismatched Fraction
20	0.065
30	0.129
40	0.178
50	0.172
60	0.180
70	0.202
80	0.205
90	0.202
120	0.235
180	0.256
240	0.258
300	0.236

Table 5.2: Result of general cases

5.2 Experiments on Energy Analysis

5.2.1 Duty Cycling GPS

After conducting experiments on the GPS traces, we have the trade-off between GPS sample rate and route accuracy. However, in order to analyze the trade-off between energy consumption and route accuracy, we also need to conduct experiments on energy analysis. Then we would be aware of how much energy we can save from duty cycling GPS receivers instead of continuous GPS sampling. The energy consumption is related to the status of the GPS receiver. Thus before energy analysis, we need to recall how GPS works.

- GPS Working Process

  GPS signal processing contains three stages: acquisition, tracking and decoding. **Acquisition**: It is the first stage when the GPS receiver starts up. During the stage, the GPS receiver searches for visible satellites in order to start receiving data transmitted from the satellites. **Tracking**: After satellites signals are acquired, the GPS receiver enters tracking stage, during which stage the GPS receiver keeps a lock to the satellites. **Decoding**: With correct tracking, the GPS receiver can decode the packets sent from the satellites. Then the location calculation is finished by the main processor. Acquisition is more expensive than the other stages.

There are three start up modes for GPS receiver: cold start, warm start and hot start. **Cold Start**: When GPS receiver has no prior knowledge of its last position and time, a cold start takes place and the GPS receiver has to search the entire space for satellites and at least one GPS frame must be downloaded from each of the satellites. It has the longest GPS Time To First Fix (TTFF) among the three start up modes. Most modern GPS receivers achieve position fixes from a cold start condition in 30 to 60 seconds. **Warm Start**: A warm start occurs when the receiver has some almanac information that is less than one week old but does not have valid ephemeris information. From a warm start condition, a modern GPS receiver can achieve a position fix in much less than 60 seconds for it only needs to decode ephemeris data from the satellite's packet. **Hot Start**: If a receiver has up-to-date almanac and ephemeris information, it will perform hot start mode, namely skip the acquisition process and start directly. It takes 0.5 to 20 seconds to get its position fix for modern GPS receivers under this mode, because it only needs to obtain timing information from each satellite.

When it comes to duty cycling GPS receiver, if the receiver has valid almanac and ephemeris information, it is more likely to perform a hot start each time. In our case, because the time that the GPS receiver has been off during each duty cycle period is much less than almanac and ephemeris valid time, it is more likely to perform hot start every time after being activated during the duty cycle. From result of the experiments on constructing route from GPS traces with reducing sample rates, we can see duty cycling GPS is still a possible way for energy efficient trajectory-based services of smart phones.

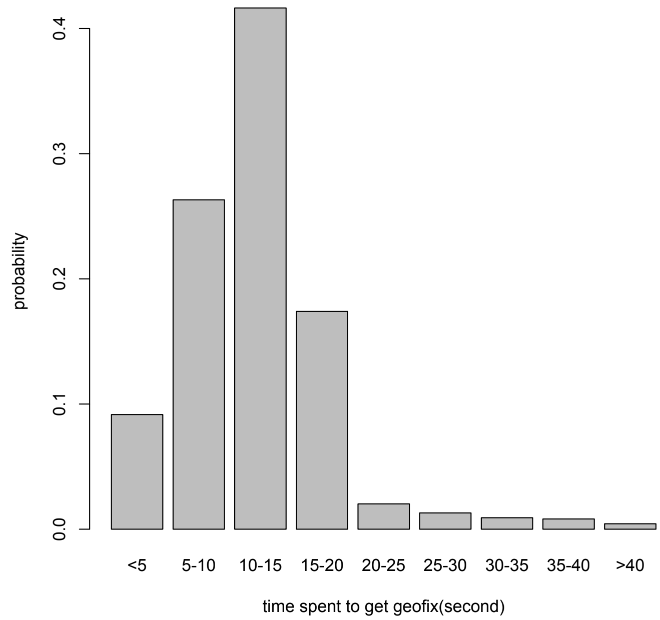
- GPS Time to First Fix (TTFF)

  In order to get trade-off between accuracy of the route constructed from GPS traces with different sample rates and the power consumption, an important task is to decide how to duty cycle GPS. From the explanation of GPS working process, we know that it takes time for a GPS receiver to get a position fix after activating it. In hot start mode, GPS TTFF is still dependent on several elements: the number of satellites visible to GPS device, moving speed of GPS device, signal strength, etc. Under normal driving condition, we tested how long it took to get a position fix each time after activating the GPS receiver on average. Because we want the GPS receiver to perform hot start each cycle, we set the duty cycle period as 10 seconds or 20 seconds. We have collected 2150 samples. The distribution of GPS TTFF is as shown in Figure 5.4.

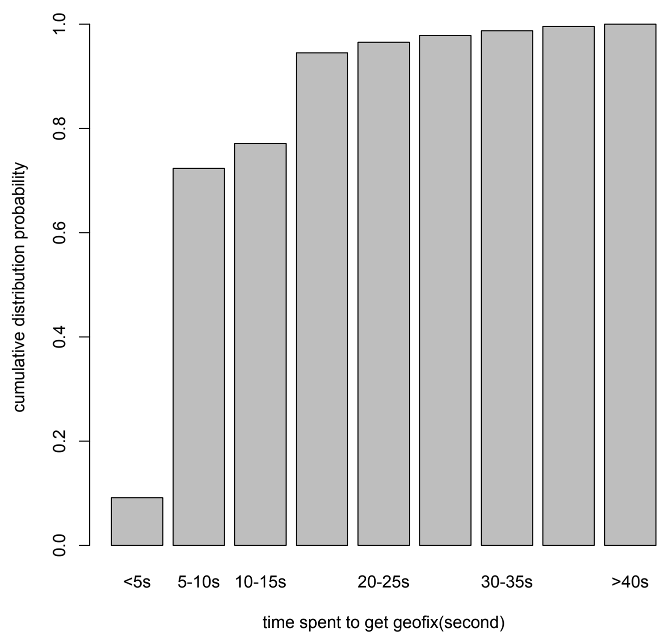
We can see from the result shown in Figure 5.4, the probability that the GPS receiver takes less than 20 seconds to get a position fix is more than 94%, which satisfies the data provided by [9]. And the average GPS TTFF from our experiments is 11.549 seconds. Based on these data together with power consumption of GPS receiving process, we can then calculate how much energy we can save from duty cycling GPS. We can suppose that the lowest energy efficiency occurs when the GPS TTFF is 20 seconds, because in order to obtain one single position fix, the receiver keeps on for 20 seconds, which means that it has to decrease the time that GPS receiver is off during the duty cycle to maintain relatively high route accuracy. And the average case for battery consumption is when we consider GPS TTFF as the average value 11.549 seconds.

5.2.2 Power Consumption Measurement

Our measurements of power consumption of GPS receiver were performed on Samsung Galaxy Player 4.0. It is an mp3 player with GPS component. We used an oscilloscope to measure the current consumption of the GPS component. Because the Android device cannot boot without battery in the device, we left the the battery connected to the device and measured continuous power transferred from the battery to the device. To avoid interference from battery charging circuitry, we took the measurements with no external charger connected to the device. We placed a 0.5 ohm resistor in series with ground and measured the voltage drop on the resistor from which we can calculate the current of the circuit. The power consumption we measured here was under hot start condition. The device ran a background service to get locations by activating its GPS receiver periodically. During the idling, the current is 8 mA and the power consumption



(a) The probability distribution of GPS TTFF



(b) The cumulative distribution of GPS TTFF

Figure 5.4: GPS sensing duration

Battery Capacity (mAh)	Power Consumption on idling (mW)	Power Consumption on Sampling (mW)
1200	29.6	325.6

Table 5.3: Battery consumption

is 29.6 mW. And the power burst after activating the GPS receiver is as high as 260 mA. However it can be ignored because its duration is much shorter than the total sensing duration. During the GPS sensing, the current is 88 mA and the power consumption is 325.6 mW. Thus for a smart phone having battery capacity of 1200 mAh, continuous GPS sampling can drain the battery in 13.6 hours, assuming that the smart phone is not used for anything else.

Chapter 6

DISCUSSION

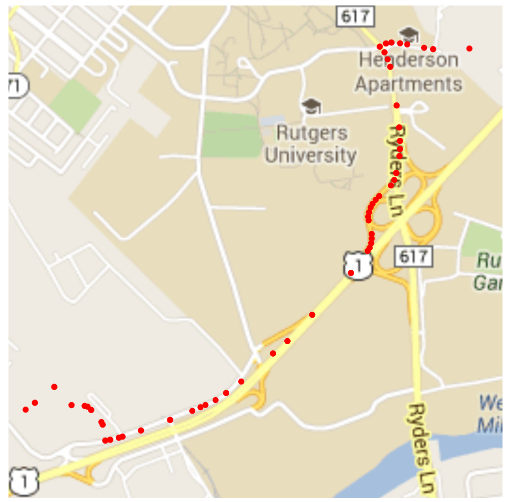
6.1 Result of Experiments on GPS Traces

6.1.1 Single Trace

It is interesting to find that for a single trace sometimes the mismatched fraction of route does not increase steadily as the sample rate decreases. This situation is caused by the randomness when removing samples in the GPS trace. Figure 6.1 are the 30-second-sample-period trace and 50-second-sample-period trace and their results.

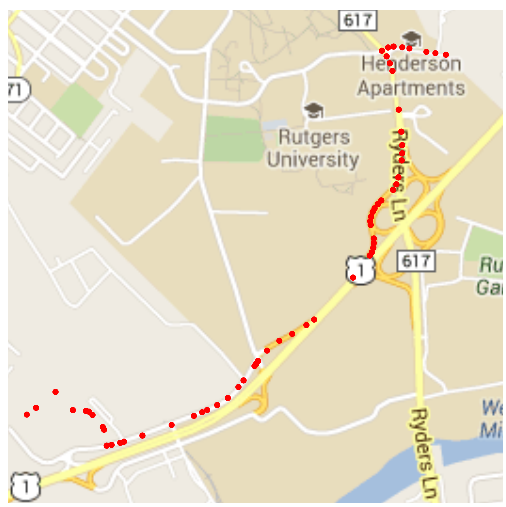
After comparing the results to the correct route, it is obvious that the path constructed by 50-second-sample-period trace is 100% accurate. On the other hand, the path constructed by 30-second-sample-period trace mismatched a fraction of the route. Figure 6.2 illustrates how it happens. As we can see, though 30-second-sample-period trace has denser nodes, it does not contain the key node which is on the curve street instead of the straight main road (marked as blue). When it comes to explore the path between the two nodes, our algorithm tends to find the shortest path (marked as blue) between the two nodes rather than the other path (marked as red). Thus it erroneously constructs a wrong path using the 30-second-sample-period trace. When factorizing the long sample period, if the short sample period is not a factor of it, then this situation may occur. On the other hand, if the short period is a factor of the long sample period, for example, 30 seconds and 60 seconds, there is no way that path constructed by the longer sample period trace is better than the one constructed by the shorter sample period trace, because the 30-second-sample-period trace contains all the nodes in the 60-second-period-trace and is twice denser. If the 30-second-sample-period trace misses a key node, the 60-second-sample-period-trace will definitely end up missing a key node

(a) 30-second-sample-period GPS trace



(b) path constructed by 30-second-sample-period GPS trace

(c) 50-second-sample-period GPS trace



(d) path constructed by 50-second-sample-period GPS trace

Figure 6.1: Single trace result

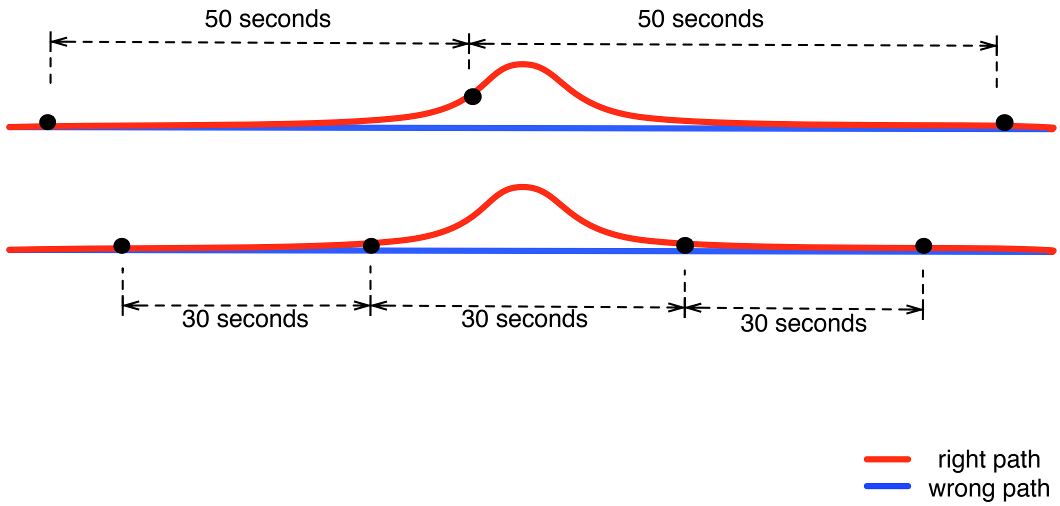


Figure 6.2: compare the 50-second trace with 30-second trace

for the same part of route as well. Thus the general trend of the mismatched fraction of route is going upward as the sample period increases.

6.1.2 General Case

We made an observation based on the result of general case in Figure 5.3. The maximum slope happens from 20 seconds to 40 seconds. By visualizing the trace on the map as shown in Figure 6.1, we know usually the mismatched fraction of route happens on the local part of the route. For highway part, it does not change much as sample period increases. Take residential road for example. The speed limit for residential road is 25 mi/hr, which means a driver could travel 0.2 miles within 30 seconds on it. Considering high density of roads in local area, 0.2 miles between two nodes means that there could be more than two streets or roads connecting them. Because more than one path are legal, the chance of choosing a wrong path by our path constructing algorithm thus increases. However, on the freeways, even though the driving speed is from 55 mi/hr to 65 mi/hr meaning a driver could travel 0.5 miles within 30 seconds, there is big chance that only one legal path between two nodes due to much less density and complexity of roads.

6.2 Energy Analysis Results

From the GPS working process, we know that in our case duty cycling GPS performs hot start each time after being activated. And even though TTFF is not a fixed value, through experiments on TTFF, we can get a clue that under normal driving condition, TTFF is not likely to exceed 20 seconds and the average value of it is 11.549 seconds. Based on that, we may consider the trade-off between route accuracy and energy consumption with average case and worst case.

We measured how much power it consumed during GPS sensing process after a hot start. And then we can calculate how much energy saved. In the combination with the result from the path constructing experiments on GPS traces, we can get the trade-off between energy consumption and route accuracy. Then we can decide whether duty cycling GPS is energy efficient. The relationship can be summarized as in Equation 6.1, 6.2.

$$T_{\text{total duration}} = T_{\text{off}} + T_{\text{on}} \tag{6.1}$$

$$P_{\text{duty-cycle GPS}} = \frac{T_{\text{on}}}{T_{\text{on}} + T_{\text{off}}} \times P_{\text{continuous GPS sampling}} + \frac{T_{\text{off}}}{T_{\text{on}} + T_{\text{off}}} \times P_{\text{idling}} \tag{6.2}$$

Where:

- $T_{\text{total duration}}$: time interval between GPS samples
- T_{off}: time that GPS has been off each time during duty cycling GPS
- T_{on}: time that GPS has been on each time during duty cycling GPS
- $P_{\text{duty-cycle GPS}}$: power consumption when duty cycling GPS
- $P_{\text{continuous GPS sampling}}$: power consumption when continuous GPS sampling
- P_{idling}: power consumption when the smart phone is on idling

6.3 Trade-off between Route Accuracy and Energy Consumption When Duty Cycling GPS Receivers

To conclude the trade-off between route accuracy and energy consumption when duty cycling GPS receivers, we took results from three parts into account: path constructing experiments on the GPS traces, GPS TTFF experiments, and battery consumption measurement. When it comes to the real world duty cycling GPS receivers, GPS TTFF is always changing. We can only set the time that GPS has been off each time during duty cycling GPS (T_{off}) fixed. The duty cycling strategy would be activating the GPS receiver at the beginning of each cycle and keeping it on until it obtains a position fix. And turn it off for T_{off}. Then next cycle begins. When analyzing the trade-off between route accuracy and energy consumption, we cannot have both GPS TTFF and T_{off} changing. We must set one of them as a fixed value. The experiments on GPS TTFF shows that for hot start condition, TTFF is most likely to be within a range. Thus based on the experiments on GPS TTFF, we divided our conclusion into two cases: worst case and average case.

- Worst Case

  As explained in Introduction, the GPS TTFF in hot start condition is usually within 0.5 to 20 seconds for modern GPS receivers. And our experiments on GPS TTFF show that the value has 94% probability that it does not exceed 20 seconds. Thus we used 20 seconds to analyze the trade-off as worst case, as indicated in Table 6.1.

- Average Case

  The GPS TTFF we measured has an average value of 11.55 seconds. We used this value to analyze the trade-off as average case, as indicated in Table 6.2.

$T_{\text{total duration}}$(sec)	T_{on}(sec)	T_{off} (sec)	Mismatched Fraction	Average Power Consumption(mW)
30	20	10	0.129	224.66
40	20	20	0.178	177.6
50	20	30	0.172	148
60	20	40	0.180	126.98
70	20	50	0.202	114.17
80	20	60	0.205	103.6
90	20	70	0.202	95.37
120	20	100	0.235	78.93
180	20	160	0.256	62.49
240	20	220	0.258	54.27
300	20	280	0.236	49.33

Table 6.1: Worst case of Trade-off between Route Accuracy and Energy Consumption When Duty Cycling GPS Receiver

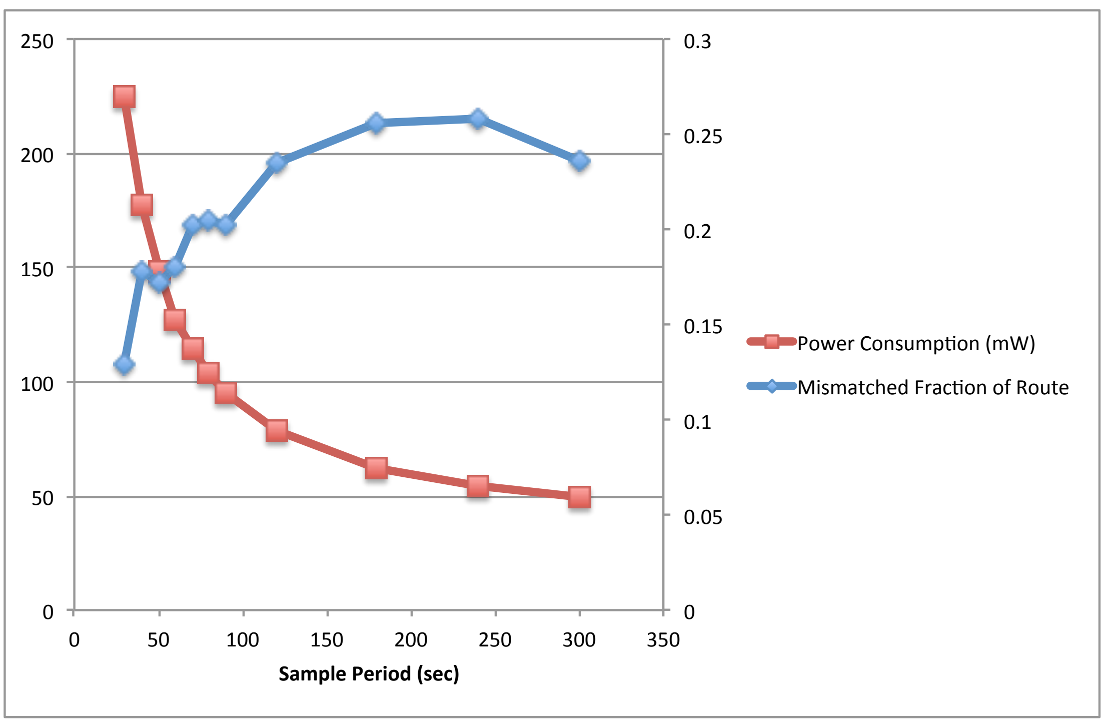


Figure 6.3: Worst case of Trade-off between Route Accuracy and Energy Consumption When Duty Cycling GPS Receiver

$T_{\text{total duration}}$(sec)	T_{on} (sec)	T_{off} (sec)	Mismatched Fraction	Average Power Consumption(mW)
20	11.55	8.45	0.065	200.54
30	11.55	18.45	0.129	143.56
40	11.55	28.45	0.178	115.07
50	11.55	38.45	0.172	97.98
60	11.55	48.45	0.180	86.58
70	11.55	58.45	0.202	78.44
80	11.55	68.45	0.205	72.34
90	11.55	78.45	0.202	67.59
120	11.55	108.45	0.235	58.09
180	11.55	168.45	0.256	48.59
240	11.55	228.45	0.258	43.85
300	11.55	288.45	0.236	41.00

Table 6.2: Average case of Trade-off between Route Accuracy and Energy Consumption When Duty Cycling GPS Receiver

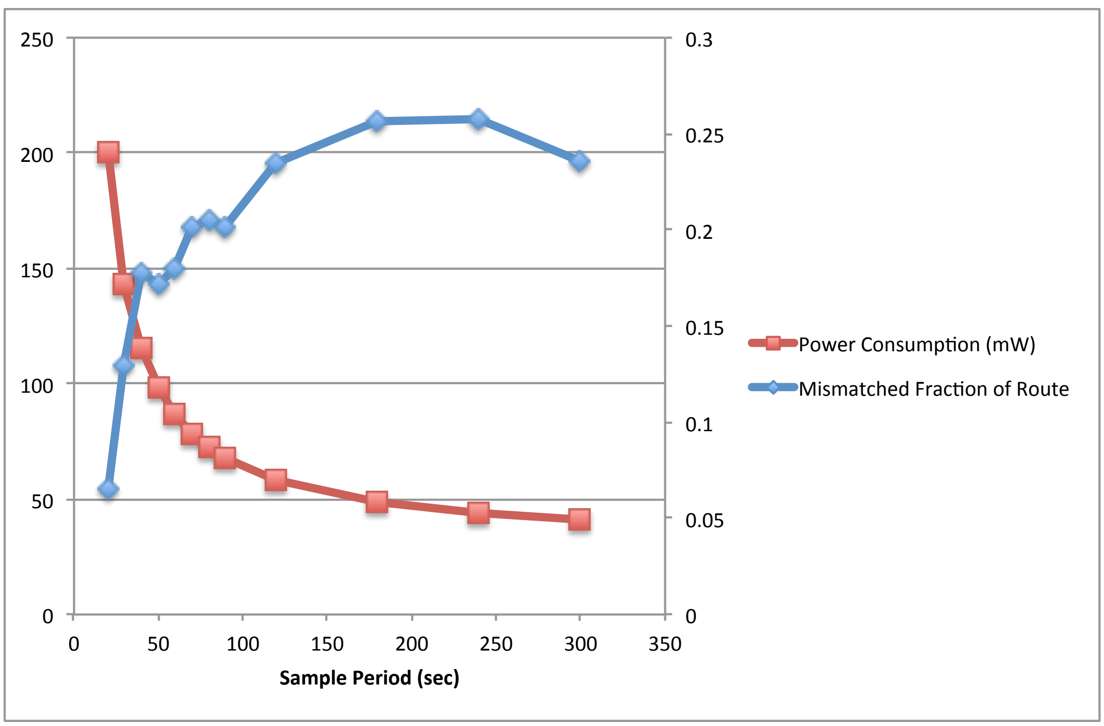


Figure 6.4: Average case of Trade-off between Route Accuracy and Energy Consumption When Duty Cycling GPS Receiver

Chapter 7

CONCLUSIONS

7.1 General Conclusion

Duty cycling GPS receiver achieves energy efficiency by sacrificing route accuracy. As the sample period increases, both energy consumption and route accuracy are generally decreasing. When the sample period exceeds 120 seconds, the mismatched fraction of route tends to be stable at 0.23 to 0.25. By visualizing each route on the map, we can make a bold assumption that this value is proportional to the composition of the route. When reducing sample rate of GPS receivers, the mismatched fraction of the route is more likely to occur on local roads rather than on highways. Thus duty cycling GPS receiver mainly saves energy from the loss of accuracy on local part of the route. If a mobile app can bear the loss of accuracy on local roads, fixed period of duty cycling GPS receiver is still a feasible approach for energy efficient trajectory-based services.

7.2 Schedule Duty Cycling GPS Based on Trade-off Result

To schedule a duty cycle based on our results, we also need to establish some criteria based on the specification that a certain app wants to achieve. We can analyze the trade-off result and make a decision by 2 decision-making methods: single criterion choice and pros/cons trade study.

- Single Criterion Choice

  For this decision-making method, only one criterion dominates. For example, for trajectory-based services, the criterion for route accuracy usually dominates over energy consumption. Thus how to schedule a duty cycle is totally based on the requirement of route accuracy. Because both of route accuracy and energy consumption decrease as sample period increases, the most efficient duty cycle period can be chosen based on the maximum loss of accuracy an app can bear.

- Pros/Cons Trade Study

  Pros/cons trade study is used when more than one criterion is known. When making decision based on this method, we need to consider all the criteria, quantify those criteria by inventing a scoring system and make decision based on the score. An example is illustrated as follow.

  An app with trajectory-based service expects to consume less than 250 mW power on GPS sampling. And the maximum mismatched fraction of route it can bear is 0.2. Both of route accuracy and energy consumption are equally weighted to the app. Associated with the result in 6.3, a scoring system can be invented as Equation 7.1, 7.2, 7.3. After calculating the score based on it, sample period with lowest score is the most efficient one under those criteria.

$$P_{\text{unified}} = \frac{P_{\text{duty-cycle GPS}}}{P_{\max}} \tag{7.1}$$

$$F_{\text{unified}} = \frac{F_{\text{duty-cycle GPS}}}{F_{\max}} \tag{7.2}$$

$$S = P_{\text{unified}} \times W_{\text{power}} + F_{\text{unified}} \times W_{\text{accuracy}} \tag{7.3}$$

  Where:

  - P_{unified}: unified power consumption on the scale of maximum power consumption on GPS sampling the app can bear
  - $P_{\max}$: maximum power consumption on GPS sampling the app can bear
  - $P_{\text{duty-cycle GPS}}$: power consumption when duty cycling GPS with certain sample period
  - F_{unified}: unified mismatched fraction of route on the scale of maximum mismatched fraction of route the app can bear
  - $F_{\max}$: maximum mismatched fraction of route the app can bear
  - $F_{\text{duty-cycle GPS}}$: mismatched fraction of route when duty cycling GPS with certain sample period

– W_{power}: weight of power consumption for the app

– W_{accuracy}: weight of route accuracy for the app

Applying the scoring system on the result in 6.3, we have scored each sample period as in Table 7.3. The sample period with lowest score is 20 seconds. Thus for this app, we can schedule duty cycling GPS receiver with sample period of 20 seconds.

Sample Period (sec)	$P_{\text{duty-cycle GPS}}$ (mW)	P_{max} (mW)	P_{unified}
20	200.54	250	0.802
30	143.56	250	0.574
40	115.07	250	0.460
50	97.98	250	0.392
60	86.58	250	0.346

Table 7.1: Unified power consumption

Sample Period (sec)	$F_{\text{duty-cycle GPS}}$ (mW)	F_{max} (mW)	F_{unified}
20	0.065	0.2	0.325
30	0.129	0.2	0.645
40	0.178	0.2	0.890
50	0.172	0.2	0.860
60	0.180	0.2	0.900

Table 7.2: Unified mismatched fraction of route

Sample Period (sec)	P_{unified}	W_{power}	F_{unified}	W_{accuracy}	Score
20	0.802	0.5	0.325	0.5	0.564
30	0.574	0.5	0.645	0.5	0.610
40	0.460	0.5	0.890	0.5	0.675
50	0.392	0.5	0.860	0.5	0.626
60	0.346	0.5	0.900	0.5	0.623

Table 7.3: Score each sample period

7.3 Future Works

To improve the route accuracy of duty cycling GPS receiver, we may consider some future works as below.

7.3.1 Using Other Sensors to Detect Turns

In 6.1.1, it shows that it is critical to detect key nodes on the route to improve route accuracy. Our path constructing algorithm tends to choose the shortest legal path between two nodes, thus it usually tends to choose a straight road over a curvy road. Yet sometimes the correct path is the curvy road. To prevent this situation, a future work can be focused on using other sensors to help detect turns during duty cycling GPS receivers. By comparing turning points and curvature of the roads, it is possible to avoid missing key nodes and improve route accuracy.

7.3.2 Adaptive Duty Cycling GPS Receiver

The loss of route accuracy tends to occur on the local roads rather than highways. Therefore, the tolerance of reducing sample rate of the GPS receiver is different to local roads and highways. To improve route accuracy on local roads, instead of using fixed sample rate of duty cycling GPS receiver, a better approach would be using adaptive duty cycling GPS receiver based on the density of road network. If the density of roads is high, then increase the GPS sample rate, otherwise, reduce it. Future works for this approach include: analyze the features of different road types, qualify the parameters that affect route accuracy, etc.

References

[1] Jeongyeup Paek, Kyu-Han Kim, Jatinder P. Singh, and Ramesh Govindan. Energy-efficient positioning for smartphones using cell-id sequence matching. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys '11, pages 293–306, New York, NY, USA, 2011. ACM.

[2] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 299–314, New York, NY, USA, 2010. ACM.

[3] Raja Jurdak, Peter Corke, Alban Cotillon, Dhinesh Dharman, Chris Crossman, and Guillaume Salagnac. Energy-efficient localization: Gps duty cycling with radio ranging. *ACM Trans. Sen. Netw.*, 9(2):23:1–23:33, April 2013.

[4] Mikkel Baun Kjaergaard, Jakob Langdal, Torben Godsk, and Thomas Toftkjaer. Entracked: energy-efficient robust position tracking for mobile devices. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys '09, pages 221–234, New York, NY, USA, 2009. ACM.

[5] Zhenyun Zhuang, Kyu-Han Kim, and Jatinder Pal Singh. Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 315–330, New York, NY, USA, 2010. ACM.

[6] Heitor S. Ramos, Tao Zhang, Jie Liu, Nissanka B. Priyantha, and Aman Kansal. Leap: a low energy assisted gps for trajectory-based services. In *Proceedings of the 13th international conference on Ubiquitous computing*, UbiComp '11, pages 335–344, New York, NY, USA, 2011. ACM.

[7] Jie Liu, Bodhi Priyantha, Ted Hart, Heitor S. Ramos, Antonio A. F. Loureiro, and Qiang Wang. Energy efficient gps sensing with cloud offloading. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 85–98, New York, NY, USA, 2012. ACM.

[8] Navstar gps user equipment introduction. 1996.

[9] National Instruments. Gps receiver testing, Mar 2013.

[10] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 336–343, New York, NY, USA, 2009. ACM.

[11] Kaisen Lin, Aman Kansal, Dimitrios Lymberopoulos, and Feng Zhao. Energy-accuracy trade-off for continuous mobile device location. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 285–298, New York, NY, USA, 2010. ACM.

[12] Donnie H. Kim, Younghun Kim, Deborah Estrin, and Mani B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 43–56, New York, NY, USA, 2010. ACM.

[13] Mikkel Baun Kjaergaard, Sourav Bhattacharya, Henrik Blunck, and Petteri Nurmi. Energy-efficient trajectory tracking for mobile devices. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, MobiSys '11, pages 307–320, New York, NY, USA, 2011. ACM.

[14] Swati Rallapalli, Lili Qiu, Yin Zhang, and Yi-Chao Chen. Exploiting temporal stability and low-rank structure for localization in mobile networks. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, pages 161–172, New York, NY, USA, 2010. ACM.