ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Master's Thesis

Estimating and Learning the Trajectory of Mobile Phones

Mahdad Hosseini Kamal

NOKIA Research Center-Lausanne
Signal Processing Laboratory 2 (LTS2)
Ecole Polytechnique Fédérale de Lausanne

Supervisors
Dr. Olivier Dousse
Prof. Pierre Vandergheynst

March 2010

Abstract

This project is based on the ongoing data collection campaign by Nokia Research Center-Lausanne. We use location data sampled everyday by mobile phones in the campaign to estimate position of the participants. It is with emerging mobile systems that combines different sensors in a mobile phone so, we can merge different information sources to improve our estimations.

Positioning is a problem encountered frequently in many applications. GPS is widely used for positioning but its output is noisy and it does not work in every location. Considering the embedded sensors and processing capacity of mobile phone, we can improve positioning of clients by other data like visited GSM cell and Wireless LANs. In addition, for situations where GPS does not work such as indoors, we can lay on these information to locate the user.

Although most mobile phones are equipped with GPS receivers, users prefer to keep them turned off because of their considerable battery consumption. On the other hand, people usually take determined paths when they want to travel among places that they frequently visit. Learning these trajectories helps to keep GPS receiver turned off and localize users by other sources of information.

We describe a new positioning algorithm with two modes of operations: one for cases that we have GPS signals and the other for times that there is not any GPS signal. This work also outlines a new algorithm, based on undergoing Nokia data collection campaign, for positioning and navigating of participants among their recurrent locations.

Acknowledgments

I would like to express my deep and sincere gratitude to my supervisor, Doctor Olivier Dousse. His wide knowledge and his persistence help have been of great value for me and provided me a good basis for my thesis.

I owe my most deep and sincere appreciation to Professor Pierre Vandergheynst, who gave me the opportunity to work in LTS2, for his constructive comments and important support during this project.

It is a pleasure to show my warm gratitude to all those who gave me the possibility to complete this thesis. I want to specially thank Nokia Research Center-Lausanne for giving me permission to commence this thesis.

Last, but not least, I thank my family, for giving me life in the first place, and educating me with its different aspects, for their unconditional support and love and encouragement to pursue my interests. It is a pleasure to dedicate this thesis to my family.

Contents

List of Figures

Chapter 1

Introduction

Position discovering and sharing have been identified as an interesting application in emerging mobile applications [3]. Location Based Services depend on users location to deliver context aware functionality for which industry anticipates huge market growth [7].

The Global Positioning System is the most popular positioning system. GPS is satellite-based navigation system, which provides geographical positioning framework. GPS system with its global satellite constellation has been designed in a way to offer a reliable and accurate coverage [4].

Despite these efforts, GPS positioning still has errors which is more severe for mobile phones with their low cost GPS chip; furthermore, it does not work where people spend most of their time. Coverage today is either constraint to outdoor environment or limited to particular buildings. As the result, location based services are not available in places that people really need.

Existing indoor positioning technologies are costly for both developers and consumers. Many positioning systems need expensive infrastructure, and sensors [3]. These obstacles have caused LBS in an unfortunate circle. Thus, while we can offer compelling expression of LBS, few LBS application can be useful in places that we spend our social life.

Moreover, development of mobile phones with the wide range of capabilities helps us to have different information sources of users in all places. If we could somehow reach to these sources, then we would be able to find new ways for positioning that works everywhere. Fortunately, with the help of ongoing campaign of Nokia Research Center-Lausanne, we could utilize different kinds of collected data and make benefit of them to have a better positioning algorithm.

The ongoing data collection campaign helps to address both the lack of ubiquity and accuracy of the current approaches to location. We will suggest a method that provides lower battery consumption and better coverage. In addition, we not only take advantage of the GPS sensor, which is included in the distributed mobile phones of the campaign, but also we benefit of having other collected users information like GSM cell towers and Wireless LANs.

There are places that we spend most of our day like our office and home. We usually take the same path, when we want to travel among these locations. It would be interesting if we could find the most common path that we usually take to travel from or to these locations, then we could turn off our GPS receivers and would be able to save our mobile phone battery life for other usage. This can be addressed by collecting an archive of the user's common trajectories, which looks doable by the ongoing campaign. As the result, we can estimate the user most common trajectories for such region. Then the mobile phone can predict the user position and navigate him to his destination based on an archive of his most probable paths and help of visiting GSM cell towers.

In this work we will find statistical model for GPS points and also use other information that we can retrieve from the current multi-sensor smart phones to increase the positioning coverage and precision. We will also explain future path estimation based on the history of the user trajectory and we will specify an algorithm that use navigation methods other than GPS to guide the user to his region of interest.

This report is organized as follows: in chapter 2 we give an introduction about the Nokia data collection campaign. We also briefly address about the campaign client and its different states machine.

In Chapter 3, we mention different source of GPS errors and we try to find a statical model for GPS point, based on that we estimate the position of the mobile user. In addition, we suggest an algorithm to find WLANs position. Then we use WLANs to locate the users in places that we do not have GPS signal. At last, we will suggest a positioning algorithm with better coverage comparing to existing ones.

In Chapter 4, we will discuss about two smoothing algorithms and we will apply them to GPS trajectories to have a smooth paths. In addition, we will explain about common paths that users take among their regions of interest and how we can turn off GPS receiver and take advantageous of other information to navigate the users to their most popular destinations.

Finally in chapter 5, the summary and conclusion are given and some future work is proposed.

Chapter 2

Data Collection Campaign

The model that we will develop during this report is based on the campaign that we are conducting right now. All the simulations are based on the data gathered during this campaign. Thus, it is useful to have a general knowledge of the ongoing campaign [5].

The campaign is supposed to run for at least 9 months and up to 200 participants. The campaign officially has started 21^{th} of September 2009 [5]. Planing of data collection was suggested by NRC[1] Lausanne for mathematical modeling, potential user applications, recommendation systems, and privacy issues.

2.1 Software

The campaign client has some special features that could have direct effect on our models and experiments so, we will explain it briefly.

The original version of the software was developed by NRC Palo Alto known as Nokia SimpleContext service and they developed Client SW. Our goal is to capture data and location tracking as much as possible. As a result, we changed timer based operation of their client to fully state machine driven mode [5]. Now we discuss about the operating states of the client.

[1]Nokia Research Center

2.1.1 Known WLAN

This mode can result in a higher power saving. When a WLAN MAC address has been seen enough, the client will enter to this mode. We divide WLANs into three different groups.

- Geo-coded know WLAN APs
  These WLANs are frequently seen and users are in the exposer of these WLANs long enough. They are in locations where GPS signals are available, in consequent, they can be localized.
- Non-geo-coded known WLAN APs
  These WLANs are like the previous ones but with the only difference that they cannot be localized, because they are in places that GPS receivers do not work.
- Other WLAN APs
  These WLANs are rarely seen. These kinds cannot change the client mode of operation.

The client stores a list of recent WLANs. If a WLAN has seen enough (2h or 50 times [5]) it will be kept in the list for 10 days. Mode overview is as follow:

- To enter: Being in the range of any know WLAN
- To exit: Movement detected, known WLAN is seen, or battery level goes below 15%
- States that may precede this state: *Outdoor Mobile with fix, Mobile Choked, Choke Rebound, Indoor Mobile, or Know WLAN Lost*
- States that may follow:*Outdoor Mobile with fix, Indoor Mobile, or Low Battery* (in case of battery level goes below the threshold)

2.1.2 Mobile Chocked and Choke Rebound

This state is for situation that movement can be sensed from changing cell towers nevertheless GPS fix is not available. Without this state the mobile phone would have tried to get the GPS fix which would quickly draining the mobile phone battery. The state can be exit by *Choke Rebound* or being in range of constant cell tower for interval of 30 minutes. Mode outline is as follow:

- To enter: No GPS fix or successive changing cell towers for 30 minutes.
- To exit: Successful GPS fix, or battery level goes below the threshold.
- Preceding states: *Indoor Mobile, or Outdoor Mobile without fix*
- Following states: *Urban Stationary, outdoor Mobile with fix, Outdoor Mobile without fix, or Low Battery*

2.1.3 Low Battery

This state is used to turn off the major battery consumption sensors like GPS receiver but still some data are collected. State summary is as follow:

- To enter: Battery level below 15%.
- To exit: Plugging to electricity.
- Preceding states: Any state.
- Succeeding states: *Plugged in*

2.1.4 Plugged in & Plugged in Long

The original idea was to turn on all the sensors, when the mobile phone is connected to the mains. But this caused an unpleasant effect when mobile phone was in *Known WLAN*. The unreliable scanning caused the client SW to exit from the *Known WLAN* state and turns on all the data collection sensors. Once again if the known WLAN is found, then the mobile phone will go to that mode again. Over night this mode transition leads to download massive GPS data assistance. To prevent this kind of problems, there are two separate modes for charging. If the mobile phone being plugged to electricity and the previous mode be *Known WLAN*, the GPS sensor is not turned on. If a known WLAN is not detected during 3 minutes, the client goes to *Plugged in Long* and turns on its GPS receiver. We abridge this state in the following:

- To enter: Charger is connected to electricity.
- To exit: Disconnected from electricity.
- Preceding states: Any one.
- Following states: Indoor Mobile.

2.1.5 Paused

This mode helps participants to manually turn off the client. This model helps applicants turn off the client for privacy issues or to do something critical. Sate notes are as follow:

- To enter: Manually enabled by the user
- to exit: Timer is elapsed, or manually by the user.
- Preceding states: Any state.
- Subsequent states: *Outdoor Mobile without fix.*

2.1.6 Priority order of states

There is a predefined priority order for the client mode as follow:

1. *Paused*
2. *Known WLAN*
3. *Plugged In*
4. *Plugged in Long*
5. All other modes are in equal priority

2.2 Client SW Performance

The client is optimized to have lasting battery as much as possible. It can last with about 16 hours of operating time [5]. Participants are supposed to plug their phone to a charger during nights to have a fully functional mobile phone during days. The operational performance of the client with N95 8GB depends on its mode of operation and is shown in Table 2.1.

State	Power Consumption	Operating Time
All On	1.4 [W]	2.8 [h]
Outdoor	0.4 [W]	8.5 [h]
Known WLAN	0.2 [W]	23 [h]

Table 2.1 – Power consumption and operation time for different modes of operation [5]

2.3 Types of Collected Data

The collected data are listed as follow:

- Localization data like: GPS coordinates, WLAN information and Cell towers details.
- Social data information like: Address book, SMS, BT[2] details, and call logs.
- Media information as pictures, and videos.
- Application usage.
- Control data like: current active profile, Battery level, Ring type and etc.

2.4 Participants

We wanted to facilitate the social dimension of the campaign as much as possible, thus the campaign information was distributed from viral channel to participants. The enrollment tendency toward participants leads them to be well socially connected before the start of campaign.

Fig. 2.1 shows how we enrolled participants inside our campaign in principal. In practice the campaign populated with some starting nodes and then branched accordingly with motivated people. Many of the participants are between 23 and 33 years old and about 40% of participants are females.

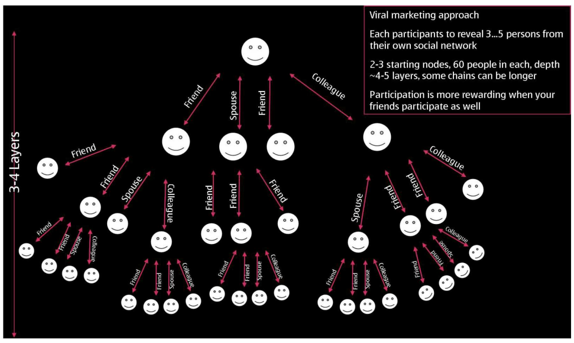


Figure 2.1 – Relation among participants of the campaign

[2]Bluetooth

Chapter 3

Location Estimation

As we discussed previously location based services are emerging mobile application, but the lack of well covered and accurate positing method have made a difficult cycle for these kind of user applications. In addition, users always wish to have a precise estimation of their location in order to monitor the places that themselves or their children have been visited. With emerging smart phones market it seems needly to step forward and consider old location estimation methods again. We all know that new mobile phones are well equipped with different types of sensors for different usage that can also give us valuable information sources to improve having more meticulous estimation of the users locations.

In the previous chapter we elucidated about campaign client state machine and we said, for example, in *Known WLAN* state even thought the mobile phone could have received GPS signal, it would turn off its GPS receiver. As a result it will lose its track. Thus we need a model for GPS points to be able to fill the gaps based on neighbor GPS data. On the other hand, we know that GPS is not accurate and it has errors.

In this chapter we aim to describe GPS and some of its error sources. We also explain the probability distribution that seems to be well describing these source of errors. We use this distribution model to estimate the position of the users between times that we have GPS position to fill the missing GPS points. Moreover, we will extend a technique for cases that we do not have any GPS signals and we lean on other capability of our smart phones.

All the simulations and schemes are based on real data that we collect from our participants of the campaign. In this way, we are quite sure that the suggested method will be useful for future mobile phone applications.

This chapter is started with Global Positioning System and its error sources then we describe the probability model that we found for each GPS point. Furthermore, we will speak about cell towers and their usage in localization which is followed by strong potential of WLANs in location estimation and how to approximate the position of WLANs. We utilize all the sources of information to have a more reliable location estimation algorithm.

3.1 Global Positioning System

Global positioning system consists of 24 to 32 satellites in Medium Earth Orbit[1] and developed by the United States Department of Defense (DoD).

This Satellite positioning method is independent of weather condition and time and also it is accessible both inshore and offshore. These distinguished features resulted in its densely usage for military purposes at its time of invent, and later it used for civilian 3D positioning and scientific applications. Nowadays, GPS has two frame works one for civil, Standard Positioning Service (SPS), and the other for military users, Precise Positioning Service (PPS), the main difference of these two services is their different accuracy [2].

In this section we will shortly review how GPS positing works and its error sources. Then we explain the probability distribution model of each GPS point.

3.1.1 Positioning Method

A GPS receiver works with messages from at least 4 visible satellites. It can determine the time and its corresponding satellite position at the time that a message is sent. The satellite position and the message transmission time for satellite i is $[x_i, y_i, z_i, ti]$. If the message is received at t_r, then the GPS receiver can calculate the transmission time as $(t_r - t_i)$. The distance that the message is traveled is calculated by $p_i = (t_r - t_i)c$ in which c is the speed of light.

A satellite position and its distance form the GPS receiver makes a spherical surface with the satellite positioned at the center. The GPS receiver

[1] A region around the Earth with altitude of 1200 miles [4].

position is somewhere on this surface. As Fig. 3.1 shows, the intersection of the four satellites spherical surface indicates the GPS receiver position.

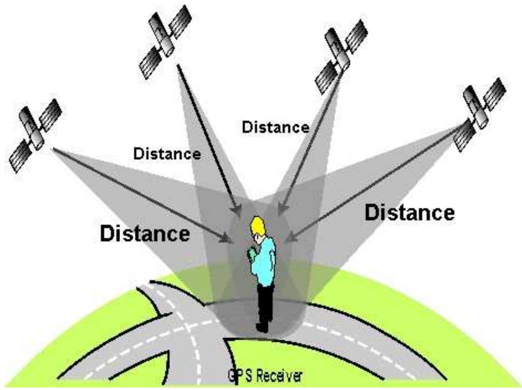


Figure 3.1 – A GPS receiver finds its position based on at least four visible satellites

3.1.2 GPS Error Sources

The above mention GPS positioning method that we explained is for no error situation. Although the GPS system has designed in a way to be as accurate as possible but still there are error sources. We try to give a short description of the major error bases to have a better overview of our GPS modeling. The major source of errors for a GPS message are described in the following parts.

Multipath Effect

Multipath is result of the reflection of the satellite signal to the receiver. This type of error usually happens in cities near high buildings. The reflected signal needs more time to reach the GPS receiver than the direct one so the calculated time by the GPS receiver is erroneous, which leads to wrong distance calculation by GPS receivers. Fig. 3.2 shows the multipath effect for GPS message.

Atmospheric effects

Another source of error is none equality of message propagation speed in its way to GPS receivers. GPS message travels with the speed of light in outer space but when it goes through troposphere and ionosphere its speed is reduced.

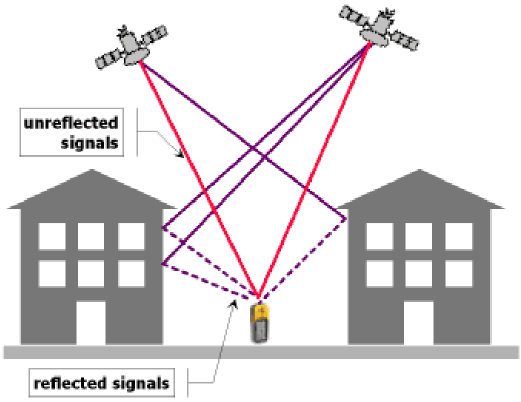


Figure 3.2 – Multipath effect for GPS message [6]

However this is a quite known error source, the typical speed variation of GPS message going through ionosphere is know for low and high frequencies. These speed variations are considered in GPS receivers.

The tropospheric error decreases traveling time of the message by refracting the electromagnetic waves. The reason of tropospheric error is of different concentration of water vapor in the troposphere, which is also dependent on weather condition. Tropospheric error is usually estimated by general calculation model.

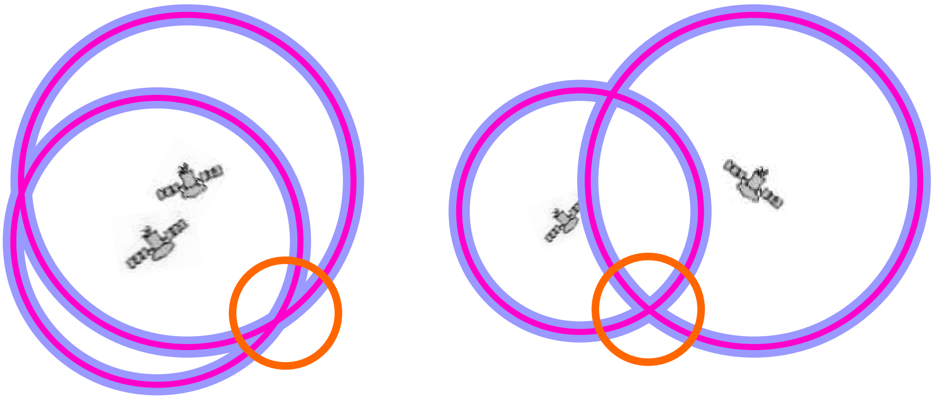
Satellite position

The position of satellites with respect to GPS receivers is another factor that affects the accuracy of the position. Dilution of Precision (DOS) is a term that measures the configuration of satellites with respect to the users.

If the satellites are far apart which is the desirable case for GPS receivers, the intersection of satellites spherical surface on the Earth leads to a more accurate positioning. When satellites are close to each other their intersection at the receiver position is obscure. Fig. 3.3 shows satellite relative positions for the two different cases.

3.1.3 GPS Data

The client collects GPS points every 10 seconds. As we previously discussed in chapter 2 considering the battery consumption of GPS receivers and the



(a) (b)

Figure 3.3 – 3.3(a) shows a bad position GPS satellites 3.3(b) shows a well positioned GPS satellites.

battery deficiency of mobile phones, we cannot keep GPS receiver in active mode all the time. Thus we should keep the GPS sensor in inactive mode as much as possible. One characteristic that we use is known WLAN. It means that if the mobile phone is in the area covered by an already-seen WLAN, it turns off its GPS receiver. This method helps us to extend the usage time of mobile phones increasingly. In addition, because the WLANs positions are available to mobile phones, they can send their location to the database and extend their operation time.

Although this method helps to extend the battery life but it also affects accuracy of localization. As we all know WLANs position are not available, thus mobile phones should try to guess a location for each discovered WLAN.

Client SW has a predefined method, when it is in the area covered by a WLAN, it looks at the last GPS tag that has been received and records that as the WLAN position. Now assume that the user just has gone through a tunnel and after the tunnel is a WLAN, the last GPS point is a point before entering the tunnel. Thus the estimated position of the WLAN has a large error. Now every time that the mobile phone is in its covered area, the phone turns off its GPS receiver and declares its position as the WLAN position and never tries to find a better estimation of its location.

Whenever the GPS receiver is active, it sends the user location and speed every 10 seconds. The location is always available but sometimes GPS receiver cannot find the speed so it will not send it.

3.1.4 GPS Error Variance

As we explained, there are many sources of error for a GPS message. In order to model the error distribution of a GPS point, we need to have the variance of the error. We tried to have an experiment to collect GPS records during a time and based on the collected data, we would try to find a distribution model for GPS error.

In our experiment we arbitrarily chose a N95 Nokia mobile phone, and put it in the office near a closed window. We tried to locate the position of the mobile phone with Google Map, unfortunately, the map was not accurate enough so, we could not locate the position of the mobile phone with the Google Map. As the result, The average of all collected data would be the best estimated position of the phone and the error variance was measured with respect to this point.

We put the phone in a fixed location and plugged it to electricity and its GPS sensor was active for two weeks. We can see in Fig. 3.4 the distribution of the GPS points are like a Gaussian distribution and its variance is equal to $\sigma_e^2 = 13.66845$ $[m]$. We should mention that this model is for a not moving mobile phone. We can deduct that the model for a moving user should be similar to this model but we should also consider speed and direction of movement.

3.1.5 GPS Modeling

Obviously our participants are moving so, we cannot apply the same model to GPS points. Moreover, we know any GPS point has error of about 13 meters.

We can also retrieve time of GPS records from the database. We want to find a distribution model for GPS point i for time t. The first model is a Gaussian distribution with GPS coordinates as its mean and variance equal to $\sigma^2 = V_i \Delta t_i + \sigma_e^2$, V_i is its speed, t_i is the time that i is recorded by the phone, $\Delta t_i = |t_i - t|$, and σ_e^2 is error variance that we previously found.

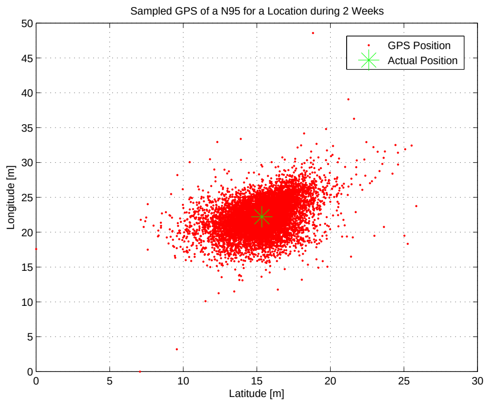
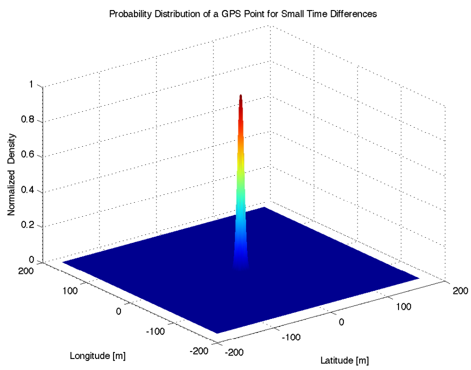
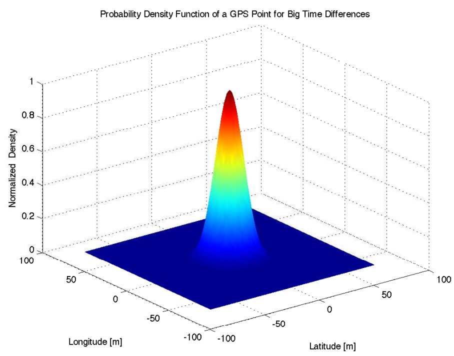


Figure 3.4 – Sample of N95 GPS sensor output for a specific position during 2 weeks.

Fig. 3.5[2] shows the Probability Density Function of this model for different time durations. If Δt_i is increased, then the slope of the GPS point probability density function will be decreased. As a consequence, comparing to the other GPS points with less time different, this GPS point has less effect for determining the positing of the user at the time t.



(a)



(b)

Figure 3.5 – GPS probability distribution using first model $V = 2.27$ $[m/s]$ 3.5(a) $\Delta t = 5$ $[sec]$ and $\sigma^2 = 25.0184$ $[m]$ 3.5(b) $\Delta t = 40$ $[sec]$ and $\sigma^2 = 104.4648$ $[m]$.

We want to estimate the probability density function of a point based on

[2]In all the GPS figures in this report we changed latitude and longitude from degree to meter.

the probability density of the known GPS points. From Bayes' rule we know:

$$f_X(x|Y=y) = \frac{f_Y(y|X=x)f_X(x)}{f_Y(y)} \tag{3.1}$$

Where X is the true location of the user at time t, Y is GPS point that we use for our estimation, $f_Y(y|X=x)$ is posterior probability function of GPS point for time t, which we modeled by Gaussian distribution, $f_X(x)$ probability density of the estimated GPS point, $f_Y(y)$ probability density of the GPS point that we want to estimated based on it. If we have more than one GPS point and we would like to estimate based on them then we have:

$$f_X(x|Y_1,\cdots,Y_n) = f_X(x|Y_1)f_{X,Y_1}(x,y_1|Y_2)\cdots f_{X,Y_1,\cdots,Y_{n-1}}(x,y_1,\cdots,y_{n-1}|Y_n) \tag{3.2}$$

We also know that probability of GPS points are independent of each other so:

$$f_X(x|Y_1,\cdots,Y_n) = f_X(X|Y_1)f_X(x|Y_2)\cdots f_X(x|Y_n) \tag{3.3}$$

Which means that the probability density of the estimated GPS point for time t is equal to multiplication of posterior probabilities that we found from individual GPS points. Estimated GPS point is a point that maximize $f_X(x|Y_1,\cdots,Y_n)$. Because we do not have any prior knowledge of the user location $f_X(x)$ has flat distribution. Moreover, $f_Y(y)$ is a constant term for all posterior probabilities so, it does not have any effect on our calculation. As a result, we can normalize the probability density function of the estimated GPS point in each step and estimate the user position at time t based on the joint probability density function of GPS points.

We should check the consistency and convergence of our PDF model. Suppose that we have chosen two GPS points $G_1 = (x_1, y_1), G_2 = (x_2, y_2)$ and we want to estimate the user position ($G' = (x', y')$) based on these points. Fig. 3.6 presents this procedure in which d_1, d_2 are distances form the first and second point from our desired GPS point and D is the distance between the two selected GPS points. Time T' is arbitrary selected between the time of the two chosen GPS points T_1, T_2.

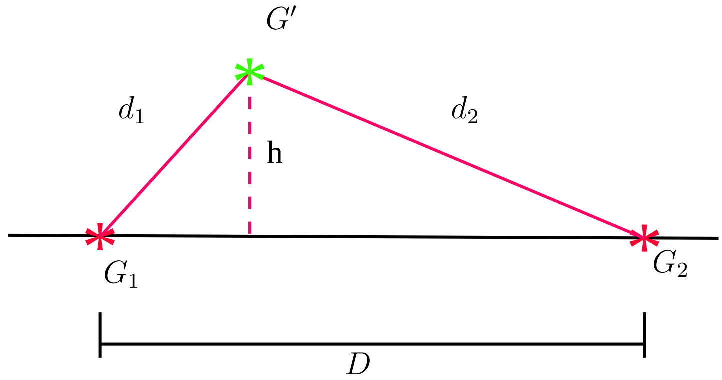


Figure 3.6 – Estimating a GPS point based on two GPS points using Gaussian model

$$
\begin{aligned}
\sigma_1^2 &= V_1 \Delta t_1 + \sigma_e^2 \qquad \Delta t_1 = (T' - T_1) \\
\sigma_1^2 &= V_2 \Delta t_2 + \sigma_e^2 \qquad \Delta t_2 = (T_2 - T') \\
f_1 &= \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x-x_1)^2 + (y-y_1)^2}{2\sigma_1^2} \right) \\
f_2 &= \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{(x-x_2)^2 + (y-y_2)^2}{2\sigma_2^2} \right)
\end{aligned}
\tag{3.4}
$$

In (3.4) f_1 and f_2 are Gaussian distribution function of the GPS points respectively. We know these probability densities are independent of each other, as the result, the PDF of the estimated GPS point, the joint PDF of G_1 and G_2, is simply multiplication of the two probability density functions. In order to find the optimum GPS point, we should find the point that maximizes this probability density.

$$
f = \frac{1}{2\pi\sigma_1^2\sigma_2^2} \exp\left(-\frac{(x-x_2)^2 + (y-y_2)^2}{2\sigma_2^2} - \frac{(x-x_1)^2 + (y-y_1)^2}{2\sigma_1^2} \right) \tag{3.5}
$$

$$
\max_{x,y} f \approx \min_{x,y} \left\{ \frac{(x-x_2)^2 + (y-y_2)^2}{2\sigma_2^2} + \frac{(x-x_1)^2 + (y-y_1)^2}{2\sigma_1^2} \right\} \tag{3.6}
$$

If we solve Eq. (3.6) for all possible values of x and y, then the estimated

GPS point is as follows:

$$x' = \frac{x_2 + Kx_1}{1+K} \qquad \text{where } K = \frac{\sigma_1^2}{\sigma_1^2}$$
$$y' = \frac{y_2 + Ky_1}{1+K} \tag{3.7}$$

If we have estimation based on n GPS points, by induction the coordinates are as follow:

$$x' = \frac{\sum_{i=1}^{n} \frac{x_i}{\sigma_i^2}}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}} \qquad y' = \frac{\sum_{i=1}^{n} \frac{y_i}{\sigma_i^2}}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}} \tag{3.8}$$

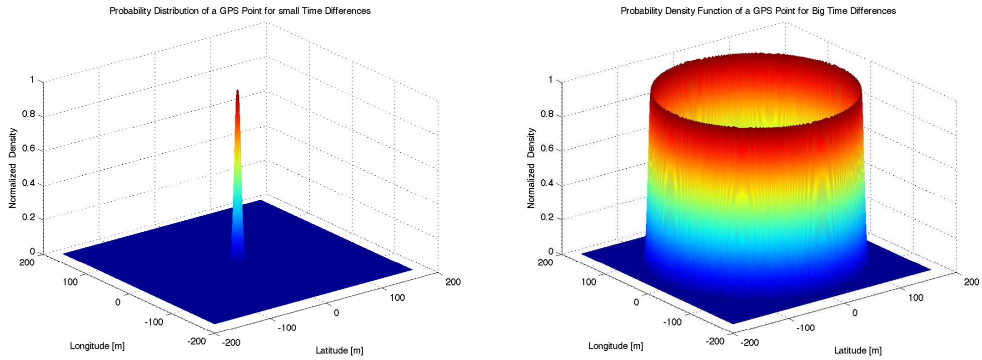
To check the convergence of the model, we will use a simple example. We try to guess a GPS position for a given time based on infinite number of GPS points having the following characteristics; their speed are the same and equal to $V_i = 1$ $[m/s]$ and they are positioned on a line with 1 meter distance from each other. We proved in Eq. (3.8) that $x' = \sum_{i=1}^{\infty} \frac{x_i}{\sigma_i^2} / \sum_{i=1}^{\infty} \frac{1}{\sigma_i^2}$ and $\sigma_i^2 = |i - t'| + \sigma_e^2$. In addition, by using integral test theorem[3] from series convergence methods, we can deduct that $\sum_{i=1}^{\infty} \frac{x_i}{\sigma_i^2}$, is a divergent series, thus this model is not an appropriate model for our problem.

So the first model, which was a quite natural model, did not work properly. The second model that we used to express a GPS point probability distribution was also a Gaussian distribution. There are some differences between the new model and the previous one. In this model we use the same error variance for all the GPS points, which is equal to the measured variance. The first change is in mean of the Gaussian distributions comparing to the previous $\eta = V_i \Delta t_i$. In Fig. 3.7 we can see that as Δt_i increases the effect of the GPS point on the estimation of the new point will decreased. In fact, σ^2 determines the dilution of this GPS point based on GPS sensor error and η decides over the covered area by the point.

[3]If a_n is monotone decreasing function we have the following characteristic:

$$f(n) = a_n$$
$$\int_1^{\infty} f(x)\,\mathrm{d}x = \lim_{t\to\infty} \int_1^{t} f(x)\,\mathrm{d}x < \infty$$

then the series converges, otherwise, the series diverges [16].



(a) (b)

Figure 3.7 – GPS probability distribution for different time durations $\sigma^2 = 13.66845\ [m]$ 3.7(a) $\Delta t_i = 5\ [sec]$ and $\eta = 1.2\ [m]$ Recording time of this GPS point is near the selected time 3.7(b) $\Delta t_i = 105\ [sec]$ and $\eta = 149.612\ [m]$. We can see as Δt_i increases the effect of the GPS point will be decreased.

We can see that in this model every direction around GPS point has the same worth and its probability value will be decreased, as we go farther away from the GPS point. So all the points on a circle around the GPS point has the same probability value. On the other hand, we know that when you take a path and follow it toward your destination, all the points aligned with your direction are more probable comparing to other points on the same circle but on a wrong direction.

Fig. 3.7 shows that this restriction has not taken into account in our model. So we should improve our model. To find a direction for each GPS point, we will take into account the effect of the previous and the next GPS points. Assume that you want to go from your home to your office and you have chosen your way to your office and started your GPS receiver. Each recorded GPS point is in your path toward your destination and also affected by your previous position and has influence on your next points.

Therefore, the direction of each GPS point is influenced by its neighbors. As shown in Fig. 3.8 we connect the former and later GPS points to the current GPS point to find their slope, their bisector will be the direction of the current GPS point. Direction estimation is an erroneous method that its error probability distribution can be best modeled by Gaussian distribution of average error about 20°. Fig. 3.9 shows the Gaussian distribution around

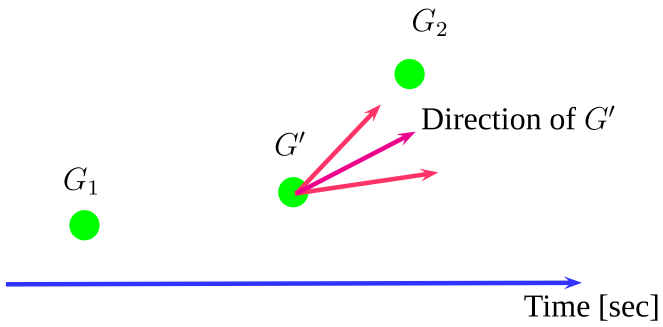


Figure 3.8 – Direction estimation of GPS points

angle axis (θ) with $\sigma_{ed}^2 = 0.35$ $[r]$. GPS points near to the estimated direction are more probable.

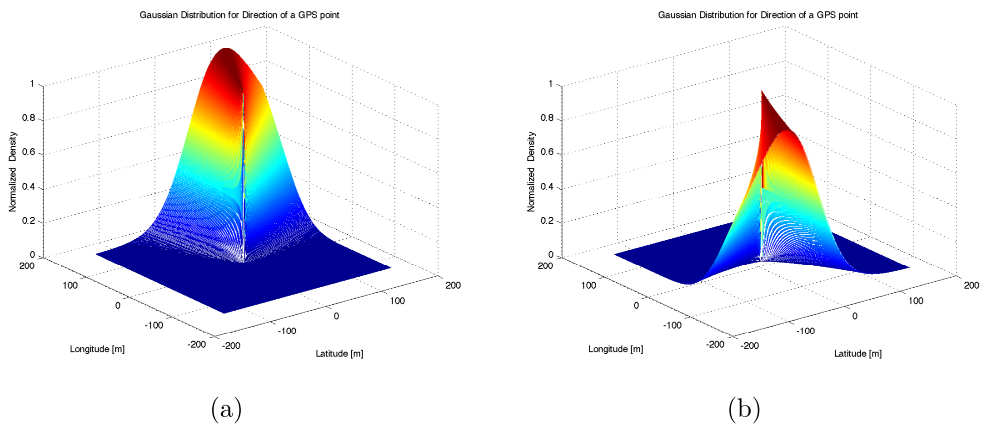
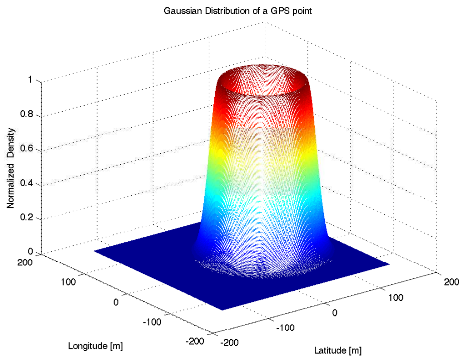


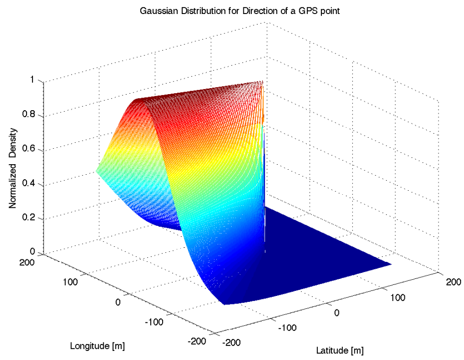
Figure 3.9 – 3.9(a) Back view 3.9(b) Front view of probability density function for direction estimation error. We can see that Gaussian distribution has the highest value in the estimated direction and as passing the estimated direction, the directions become less probable.

The final distribution model that we use for each GPS point is combination of the probabilities of the 2^{nd} model. Since the two probability distributions are independent of each other, the probability distribution of each GPS point is multiplication of this two probabilities. With this model we could consider all the available parameter for a GPS point and consider errors that are more important for a GPS point.

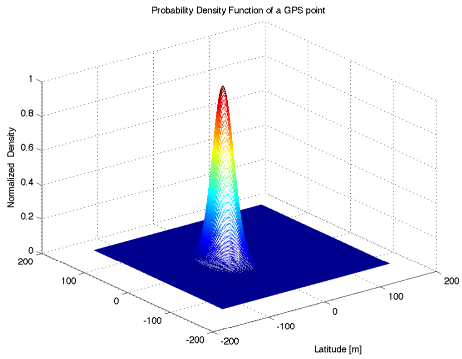
Fig. 3.10 shows the actual distribution of a GPS point. There is a Gaussian distribution with respect to the r axis and another Gaussian distribution



(a)



(b)



(c)

Figure 3.10 – 3.10(a) Gaussian distribution of the GPS point based on its speed and time 3.10(b) Gaussian distribution of the GPS point for its estimated direction 3.10(c) The probability distribution of the GPS point. The probability distribution is multiplication of its probability distribution around angle axis and r axis.

with respect to the θ axis. The final distribution of the GPS point is multiplication of the above mentioned Gaussian distribution and is showed in Fig. 3.10(c). We summarize GPS probability distribution in algorithm 1.

We could find a model for estimating the user location, merely when our GPS receiver works properly. But do we always have a GPS track and do we spend more of our time out in the streets? Surely not, even when you are out, it sometimes happens that you do not have GPS signal, for example, when you are in a tunnel. Moreover, we spend most of our day time in our office or at home, which there is no GPS signal.

Algorithm 1 GPS point distribution

$\sigma_e^2 \longleftarrow 13.66845$
$\eta_r \longleftarrow V_i \Delta t_i$
$\sigma_{ed}^2 \longleftarrow 0.35$
$\eta_\theta \longleftarrow$ estimated direction
$f_r(r) \longleftarrow \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp \frac{-(r-\eta_r)^2}{2\sigma_e^2}$
$f_\theta(\theta) \longleftarrow \frac{1}{\sqrt{2\pi\sigma_{ed}^2}} \exp \frac{-(\theta-\eta_\theta)^2}{2\sigma_{ed}^2}$
$f_{gps} \longleftarrow f_r(r) \cdot f_\theta(\theta)$

In all remarked situation with the current model we cannot locate the user. On the other hand there are some other sources of information that we can rely on to estimate the person location like: WLANs, and cell towers. In the next section we will extend our localization model for cases that there is no GPS signal.

3.2 Global System for Mobile Communication (GSM)

GSM is the most common cellular telephony in the world and is utilized within more that 200 countries. From September 2005, GSM cellular phone has 1.5 billion users [11]. A GSM cell tower consists of a number of directional antennas that covers a specific area so-called cell. Depending on the covered area and traffic load each cell is composed of a number of physical wireless channels. Each base station is identified by combination of “Country code”, “area code”, and “cell ID”.

Many applications using cellular networks to benefit the cell tower IDs and their locations. Examples of such applications like location-sharing and disclosure systems [1]. Considering densely populated cosmopolitan areas with cell towers, thus cellular networks are an accessible localization source of information. As a consequence, we would try to rely on cellular networks as information to locate the mobile phones.

In addition, data regarding to position of GSM cell towers is not available in many countries and in those which is available is not easy to find [1]. there are some companies that they have an estimated position for each cell tower for their internal usage. Fortunately, we could find a location for most of the cell tower seen in our campaign from one of this companies[4]. Physical location of GSM cell towers helps us to have an estimation of the mobile location.

3.2.1 Localization Using GSM Cell Towers

In each stamp of time a mobile phone is connected to a GSM cell tower, beside that there are usually other cell towers that could also cover the phone. If the mobile phone could record all the cell towers that it can see, by having the range and position of the cell towers, then we could locate the mobile phone. If we assume that the mobile phone could pick three or more cell towers, we can compute its approximate location in relation to these cell towers.

Our campaign software is just able to find the cell tower ID that the mobile is connected, and cannot discover other cell towers that are in range. On the other hand, GSM cell towers usually cover a large area so, localization based on a GSM cell towers cannot be accurate.

Although GSM cell towers are a source of information that is always available, we just use it in crucial situations that we do not have any other source of information except the GSM cell tower ID that the mobile phone is connected. Mostly, WLANs are more reliable for user positioning.

3.2.2 GSM Cell Tower Application

An interesting usage of GSM cell towers is to find places that the user spends most of his time during a day, like office, and home. When we are in a place for a long duration of time, the cell tower ID will remain permanent. In the other hand, during this period we do not change out GSM cell tower. Thus, if we look at cell towers that are seen by a mobile phone, there are some cell towers that the user has spent considerably longer time connecting with them comparing to other cell towers. Those cell towers are ones that cover the places that the user spends most of his time.

Another application of GSM towers can be movement detection. We already said if users do not move, they are connected to a certain cell tower as they start moving the cell towers that they are connecting begins to change.

[4]The company that we used to find location of cell towers is: Open Cellid

By deciding over a time that we can consider the user in standing position, we would be able to detect the movement of participants [8].

3.3 Wireless LANs

Wireless LANs (WLAN) were built to give the possibility to access a fixed network architectures. Because of their flexibility, connectivity and their low price, their market is growing rapidly. A group of them are classified with IEEE 802.11 working group and amongst 802.11b has become the most successful one. They work up to 11 Mbps and operate in 2.4 GHz band which is the ISM band[5] [7].

3.3.1 Indoor Positioning

GPS is the main positioning system; however, it does not work in indoor places. Knowing the position of a user is of great importance, for this reason there are some systems that are specifically built for indoor positioning such as Active Badge, Cricket, and the Bat that can be used for indoor positioning [11].

These systems are costly and not everybody can utilize them. Some existing infrastructures that can be used are like television signals and Wireless LANs [10]. WLAN positioning is the easiest among those methods [7].

3.3.2 Wireless LAN Positioning

As we previously mentioned GPS is not available in indoor places and during this time we can make profit from WLANs or cell towers information as a source of information to locate the mobile user. In addition, we know GSM cell towers are not an accurate source of information for position estimation.

The only problem for WLAN positioning is that we do not know the position of WLANs. If we could somehow find their position then we would be able to utilize them to find the user location. Thus we firstly explain the method that we use to find the WLANs' position then we will describe how we use their position to estimate the position of the user.

[5] industrial, scientific and medical (ISM) radio bands

Ascertaining WLANs Position

The client SW has the possibility to search for all WLANs every 60 seconds and records their MAC address. Now assume that at the time that our mobile phone sees a WLAN it also has GPS signal so, it can impute the GPS position (P_{GPS}) of the WLAN. On the other hand, the mobile phone is not exactly at the position of Access Point (AP), so there is an error of about 50 meters for estimating of the WLANs position (P_{AP}) depending on the covering range of WLAN. To decrease this error we would check the GPS every time that the mobile phone is in the range of that WLAN and take the average position as the position of that WLAN. Moreover, considering number of participants in our campaign, we are supposed to have a good estimation for popular[6] WLANs.

The challenging issue is the indoor WLANs. For this type of WLANs, we do not have GPS signal at their AP location. Only for a small portion of these WLANs, when the mobile phone is covered by them, the phone can receive GPS signal.

The method that we use for positioning of this series is as follows: when a mobile phone is in range of a WLAN from this kind, it will look at the last 50 second of its GPS track. We assumed that a person cannot go too far in 50 seconds.

If there is any GPS position available, it will take the one with least time difference, between the time that the GPS point is recorded and that of WLAN search, as the WLAN position. We iterate this method for each WLAN in every time that they are seen. At last we take the average of the all estimation as of the WLAN position. We summarize WLAN positioning in the algorithm 2.

3.4 Position Estimation

Our position estimation method consists of two separate modes based on availability of GPS signal:

1. GPS Mode
2. WLAN-GSM Mode

[6] WLANs that are seen with many participants of the campaign.

Algorithm 2 WLAN localization algorithm using mobile phone GPS receiver

A WLAN is detected by the mobile phone
if Its MAC address is new in the database **then**
    Add this WLAN to the database
end if

while GPS is available **do**
    $P_{temp} \leftarrow P_{AP} \cdot Counter + P_{GPS}$
    $Counter \leftarrow Counter + 1$
    $P_{AP} \leftarrow P_{temp}/Counter$
end while

if GPS is not available **then**
    Look at GPS data of 50 sec. before

    if There is any GPS entry **then**
        Take the nearest GPS
    end if

    $P_{temp} \leftarrow P_{AP} \cdot Counter + P_{GPS}$
    $Counter \leftarrow Counter + 1$
    $P_{AP} \leftarrow P_{temp}/Counter$
end if

3.4.1 GPS Mode

In this mode of operation our GPS receiver is connected to GPS satellites so, we can locate the user based on GPS data. We described in section 3.1.5 how we can find the probability distribution of each GPS point.

To estimate the position (P') of the user for a given time, we could consider the influence of all the GPS points. Not only it is not necessary, but also in many cases it can produce huge errors: for example, Fig. 3.11 shows that although G_1 and G_2 are near to each other in spacial domain, they are far apart in temporal domain. If we wanted to estimated the user position in a time between G_1 and the next GPS point, then G_2 can have a disturbing effect on our estimation method; because of the difference in estimated direction of G_1 and G_2, we will estimated an incorrect position.

Therefore, the number of GPS points that we combine to estimated a

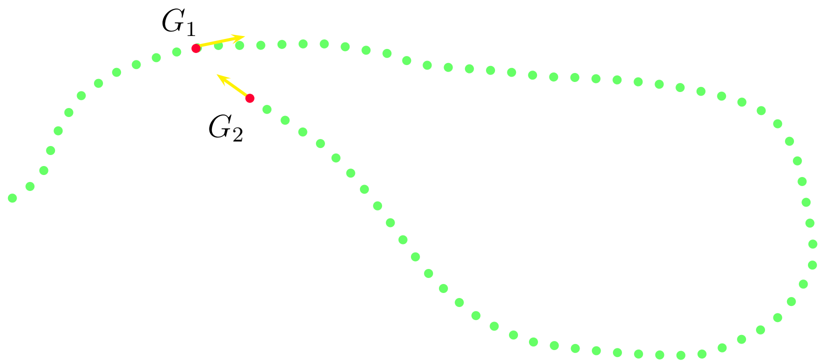


Figure 3.11 – This figure shows that we should not take into account all GPS points to estimated a position.

position is a critical issue. This number also depends on the speed of the user. If the user has high speed then two consecutive GPS points are also far apart and there is no need to combine more than two points.

As a result, we consider both special and temporal domain. The time difference of each GPS point to the time that we want to estimated the position is also an important issue. As we saw in Fig. 3.4 if the device does not move, it collects different GPS coordinates for a point; on the other hand, it has noisy output that we should consider. Although they may have big time differences but because mobile phone is static, they are all should be considered as one GPS point. We know that our client records GPS every 10 seconds and the calculated GPS error is $\sigma^2 = 13.66845\ [m]$ so, even if you move but if your speed is less than $1.3\ [m/s]$ for a GPS receiver, you are the same as a stagnant user. As a consequence, we consider all the GPS point less than σ^2 of distance as one point.

Fig. 3.12 shows the probability distribution of 6 different GPS points that are considered noisy outputs based on GPS error. It also shows the predicted probability density and the estimated position for a given time.

In addition, we mentioned that if GPS points have large time difference, though they are near in special domain, it is wrong to consider them together. Since the GPS point is far from the time that we want to estimate the position of the user, it cannot have any effect on that time except just producing estimation error. We should consider these matters in estimation method and select those points that fulfill the restrictions.

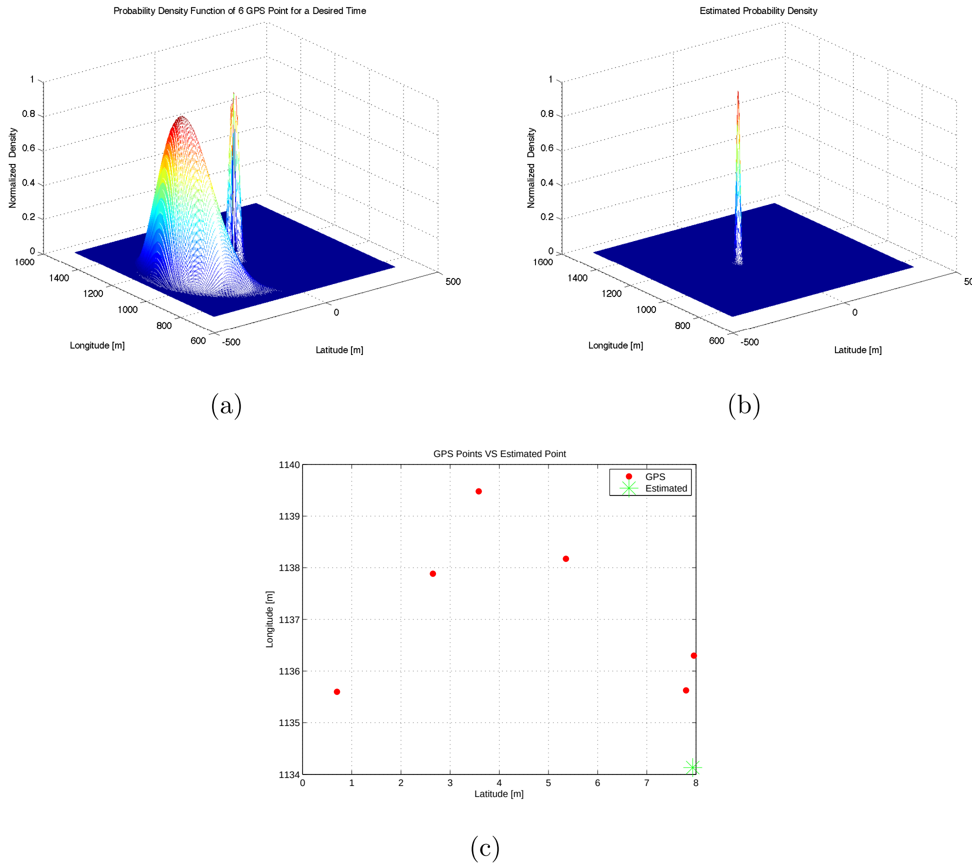


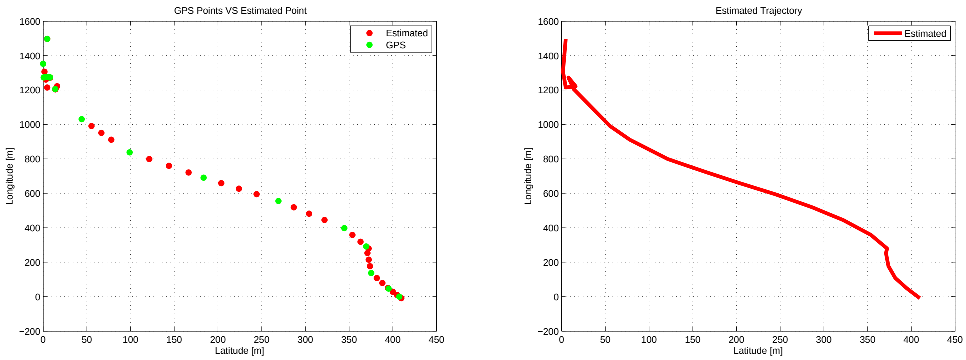
Figure 3.12 – 3.12(c) Probability density of 6 GPS points 3.12(b) Predicted probability density 3.12(c) Estimated GPS point base on noisy inputs

Among the remaining GPS points we calculate the probability density for each of them (f_{GPS}), then we should calculate the distribution of their joint probability. We also know that GPS points are independent of each other, so their joint probability is equal to multiplication of each probability density (f'_{GPS}). The estimated position is equal to:

$$P' = \quad argmax_{(x,y)} \quad f'_{GPS}(x, y) \tag{3.9}$$

The estimation error (e') is equal to the position that the probability distribution of estimated point is decreased to 0.1 of its highest value. This value shows the error radius that the user can be inside.

An important usage of our model is to find the trajectory that the user has taken during a time. For this we estimated position of the participant for different time stamps during the desired time with the help of the GPS points. Now if we connect estimated points together the user path is constructed. Fig. 3.13 illustrates estimated position versus GPS points, we also used estimated position to find the user trajectory. Moreover, in Fig. 3.13 the estimated trajectory needs smoothing, we can use some smoothing functions like: Bézier which will be discussed in details in the next chapter.



(a) (b)

Figure 3.13 – 3.13(a) Shows estimated position versus GPS points. We can also see that in some parts of figure there are some errors that with we used the denoising method to combine them. 3.13(b) If we estimated enough position for a given time duration then we can find the trajectory that the user has taken.

3.4.2 WLAN-GSM Mode

Fig. 3.14 is an example of the situations that we do not have GPS for a certain duration of time, consequently, we cannot use GPS Mode. When the time difference between two consecutive GPS points is more than 300 seconds, the positioning method enters its second operating mode. GPS points are recorded about every 10 seconds. If there is not any record during 300 seconds, we have lost 30 GPS points. There are two possibilities of the user state. First if we assume that the user is moving, considering average speed of 40 $[Km/h]$, then he has traveled for about 3 $[Km]$, second if we assume he is not moving and he is inside a closed area. In the both situations the GPS point that we have are old and we cannot just predict the user position based on them. As a result, we should take into account other information sources.

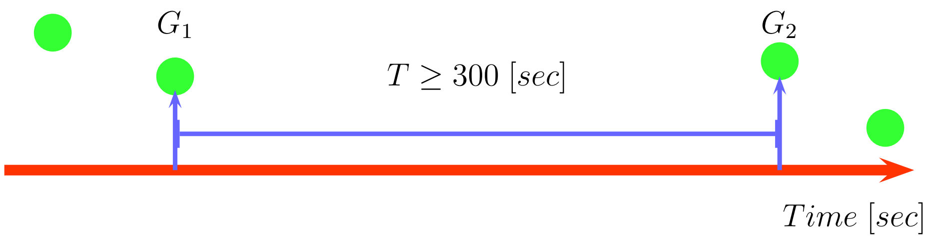


Figure 3.14 – This figure shows WLAN-GSM mode and its criterion is the time difference between two consecutive GPS ($T \geq 300\ [sec]$).

We assume that we have found position of WLANs (P_{AP}) and GSM cell towers P_{GSM} beforehand. We estimate the location of the user based on the WLANs and GSM cell towers covering him. Firstly, we should retrieve bordering GPS points ($G_1,\ G_2$) relative to the time that the algorithm goes into this mode. Hence, These bordering GPS points facilitate finding the region containing the user, though it could be large. In addition, we should get all the visited GSM cell towers and WLANs by the user during this period.

WLANs are the most accurate estimation of location, because they have covering range of about 50 meters, which is negligible comparing to the range of cell towers and the distance between G_1, G_2. In order to improve our estimation we make an average of all WLANs and declare that as the user position. By induction the upper bound of the estimation error is equal to half of the maximum distance between the estimated position and the adjacent GPS points. Distance between bordering GPS points is roughly the diagonal of the circle that the user is during the time that we have lost GPS signal so, the upper bound of the error is equal to the distance that he could traverse to see the WLAN. In addition, because the user could have started from the WLAN or from bordering GPS points thus, the average error is equal to half of this distance.

Fig. 3.15 is an example of location estimation based on observed WLANs position. Bordering GPS points are $G_1,\ G_2$, the estimated user position (G') is one hour after the first GPS point (G_1). This figure shows that the user parked his car and then went to his office which reveals a perfect estimation of the user position.

There are situations that we cannot find any WLAN. In this circumstances we surely have GSM cell towers that the user is connected. Like before, we find the bordering GPS points and calculated their distance to all seen GSM cell towers (D).

Figure 3.15 – $G_1,\ G_2$ are bordering GPS points with time difference of about 5 hours. G' is the estimated position of the user after approximately 1 hour. As expected the user is in the office for the expected time which completely matches to the reality.

We choose the longest and shortest distance and the center of the shortest distance as the estimated position of the user. The error upper bound with the same reason as WLAN case is equal to half of the longest distance.

The method that we use to estimate the path that the user has traveled during a time is composed of estimating some points during this time based on our estimation method. Then we connect the points to make the desired user trajectory. We summarize position estimation in algorithm 3.

3.5 Control Experiment

In all the presented figures we used real data but in order to verify our algorithm, we asked from a participant to point out the places and the route that he has taken during six hours. The GPS track of the user was available just during the first hour of the experiment and after that we did not have any GPS data of the user. Fig. 3.16 shows our algorithm results and also what the

Algorithm 3 Position estimation using GPS, WLAN, and GSM cell towers

if $T_{G_2} - T_{G_1} \leq 300\ [sec]$ **then**
  Retrieve relevant GPS points based on distance & time

  for $i = 1$ to number of GPS points **do**
    $f_{GPS}(i) \leftarrow f_r(r) \cdot f_\theta$
    $f'_{GPS} \leftarrow f_{GPS}(i) \cdot f'_{GPS}$
  end for

  $P' \leftarrow argmax\ \ f'_{GPS}$
else

  if WLANs are available **then**
    $n_{WLANs} \leftarrow$ Number of observed WLANs
    for $i = 1$ to n_{WLANs} **do**
      $P_{temp} \leftarrow temp + P_{AP}$
    end for
    $P' \leftarrow P_{temp}/n_{WLANs}$

  else
    Find bordering GPS points
    $n_{GSM} \leftarrow$ Number of in range GSM cell towers

    for $i = 1$ to n_{GSM} **do**
      $D(i) \leftarrow$ distance to the bordering GPS points
    end for

    $D_{max} \leftarrow \max_i D$
    $D_{min} \leftarrow \min_i D$
    $P' \leftarrow D_{min}/2$
  end if
end if

user has confirmed as his positions. GPS track of the user is demonstrated by a blue line and the dashed line is the output of GPS Mode. The green Point is the estimated position of the user during the time that we did not have any GPS data. The actual locations of the user at start and end time of the experiment are shown by stars. In this figure we can see the precision of our method during the time that we had GPS track and how accurate we could fill the gaps between consecutive GPS points. In addition, for the time that we did not have any GPS data from the user, we could relay on other collected information and estimate his position based on them.

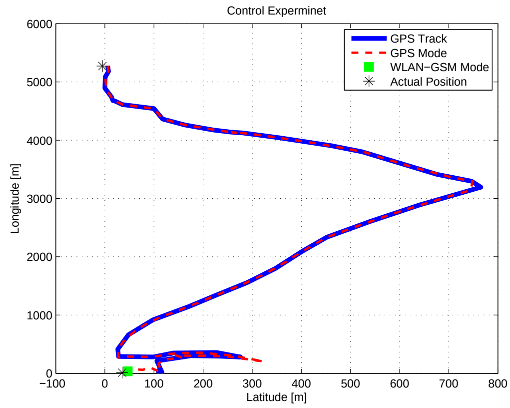


Figure 3.16 – We used our positioning algorithm to find the location of the user during 6 hours. For the first our we used GPS mode and for the remaining hour the algorithm was in the WLAN-GSM mode. The actual location of the user at the start and end time of the experiment are demonstrated by star.

Chapter 4

Trajectory Learning and Future Path Estimation

As we saw in the previous chapter the estimated path needs smoothing so, we will explain some smoothing algorithms and compare their results over GPS trajectories. In addition, We want to predict the most common path that a user takes; for example, from his home to his office. For the purpose we should find the ROIs [1] of the user, which is possible based on WLANs and GSM cell towers.

Having ROIs and routes that the user takes among his ROIs help us to learn the most probable paths for his traveling among ROIs. For learning these routes, we would have many superimposed trajectories that are collected during the time that the user participates in the data collection campaign. Then we have a set of unorganized data points that we use smoothing algorithms to find the most common trajectory that the user takes.

In this chapter we will discuss about Bézier curves and their usage in smoothing data points We also discuss two different smoothing algorithm based on B-spline curves. At last, we will explain future path estimation and the usage of B-spline smoothing in this area.

4.1 Smoothing

We consider GPS points as a set of unorganized data points[2] $\mathbf{X}_k$, $k = 1, 2, \cdots n$, which represents an unknown but not-self intersecting curve that

[1] Region Of Interest

[2] Because in general users take unpredictable moving direction and speed.

we want to compute its approximated curve. Unorganized data points are called *point cloud* and the final approximated curve is called *target curve* [13]. Given data points $\mathbf{X}_k$, we want to find target curve $\boldsymbol{P}(t)$ such that the objective function f is minimized[3].

$$f = \sum_{k=1}^{n} \min_t \|\boldsymbol{P}(t) - \mathbf{X}_k\|^2 + \lambda f_s \tag{4.1}$$

(4.1) means the points of $\boldsymbol{P}(t)$ should have less possible distance to the point cloud. f_s is regularization factor to assure a smooth output, and λ is a weight constant.

To find a desired target curve, we will introduce two curve fitting method which utilize Bźier curves. Then we will benefit of these curve fitting method for smoothing.

4.1.1 Bézier Curve

An important parametric model for smoothing is Bézier curves. Bézier curves were invented by Pierre Bézier for designing automobile body in 1962 [14]. A Bézier curve of degree n is defined as follow:

$$\boldsymbol{P}(t) = \sum_{i=0}^{n} \mathbf{B}_i(t)\mathbf{P}_i \qquad t \in [t_0, t_k]^{4} \tag{4.2}$$

In (4.2) $\mathbf{P}_i$ is control point such that $P(t_0) = \mathbf{P}_0$, $P(t_n) = \mathbf{P}_n$ and $\mathbf{B}_i(t)$ is a Bernstein polynomial represented as:

$$\mathbf{B}_i(t) = \binom{n}{k} (\frac{t_1 - t}{t_1 - t_0})^{n-i} (\frac{t - t_0}{t_1 - t_0})^{i} \qquad i \in \{0, 1, \cdots, n\} \tag{4.3}$$

Bézier curves have some special properties that makes them useful for path planning [14]:

- The curve end points are $P_0,\ P_n$.
- They are within convex hull of the control points.

[3] $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T\mathbf{x}}$

[4] For this report we take $t_0 = 0,\ t_k = 1$.

Now we defined two curve fitting algorithm that use Bézier curves as their fitting function which are different in the number of control points.

4.2 Single Control Point Error Minimization

We start with a simple Bézier curve with degree equal to $n = 3$. We will try to find a curve fitting method based on quadratic Bézier. Bases of quadratic Bézier are $\mathbf{B}_0(t) = (1-t)^2$, $\mathbf{B}_1(t) = 2t(1-t)$, and $\mathbf{B}_2(t) = t^2$. As we mentioned the point cloud is $\mathbf{X}_k$, $k = 1, 2, \cdots n$ in which x_1, and x_k are the user ROIs, which we want to find the trajectory of the user between them. The position of ROIs are know exactly and we want to find the smoothest path that connects them.

We define $\mathbf{P}_0$ as $\mathbf{x}_0$ and $\mathbf{P}_2$ as $\mathbf{x}_k$, which are fixed during all states of problem, now we should choose $\mathbf{P}_1$ in a way that we have the smoothest possible curve. We propose an algorithm based on iterative error minimization and gradient vector $\mathbf{D}$. We guess an initial value for $\mathbf{P}_1$ and iteratively update it by $\mathbf{D}$. Firstly, we propose a way to find a good initial estimation of $\mathbf{P}_1$.

4.2.1 Initial Estimation of $\mathbf{P}_1$

A good initial estimate of $\mathbf{P}_1$ helps us to decrease the number of iterations so, the algorithm diverge faster if we choose an proper value for $\mathbf{P}_1$. For this reason, we will explain an initial estimation method for $\mathbf{P}_1$, which is well suited for this method (more details are available in [15]). We consider space $\mathbf{S}$ spanned by $\{\varphi_i\}_{i\in\mathbf{I}}$ where $\mathbf{I} = 1,\ 2$ as follow:

$$\begin{aligned} \varphi_0 &= \mathbf{P}_0 \\ \varphi_1 &= \frac{\mathbf{P}_2 - \mathbf{P}_0}{\|\mathbf{P}_2 - \mathbf{P}_0\|} \end{aligned} \tag{4.4}$$

Each point $\mathbf{x}_i$ of the point cloud in the space $\mathbf{S}$ is represented as [12]:

$$\mathbf{x}_i = \boldsymbol{P}_{\mathbf{S}}(\mathbf{x}_i) = \sum_{j\in\mathbf{I}} < \varphi_j, \mathbf{x}_i > \varphi_j \tag{4.5}$$

Now we consider the projection of $\mathbf{x}_i$ into subspace S' where as follows:

$$\widehat{\mathbf{x}}_i = \boldsymbol{P}_{\mathbf{S}'}(\mathbf{x}_i) = \varphi_0 + \alpha\varphi_1 \qquad \alpha = \varphi_1^T(\mathbf{x}_i - \varphi_0) \tag{4.6}$$

For the initial estimation of $\mathbf{P}_1$ we take the point in the point cloud with the maximum distance from the subspace $\mathbf{S}'$.

$$i_{max} = \max_i \|\mathbf{x}_i - \widehat{\mathbf{x}}_i\| \tag{4.7}$$

i_{max} is the index of the point with the maximum distance from the subspace $\mathbf{S}'$. We define $\boldsymbol{P}(\hat{t})$ as the corresponding point on the Bézier curve to $\mathbf{x}_{i_{max}}$ on the point cloud where $\hat{t}$ is:

$$\hat{t} = \begin{cases} 0.45 & \text{if } 0 \leq \alpha < s \\ 0.5 & \text{if } s \leq \alpha < 2s \\ 0.55 & \text{otherwise} \end{cases} \tag{4.8}$$

and $s = \|\mathbf{P}_2 - \mathbf{P}_0\|/3$ [15]. $\hat{t}$ is not always the best initial estimation but it can always reduce number of iterations.

We can find the initial value of $\mathbf{P}_1$ as:

$$\mathbf{P}_1 = (\mathbf{x}_{i_{max}} - \mathbf{B}_0(\hat{t})\mathbf{P}_0 - \mathbf{B}_2(\hat{t})\mathbf{P}_1/\mathbf{B}_1(\hat{t})) \tag{4.9}$$

Now that we could find the initial value for $\mathbf{P}_1$, we use gradient vector to update its value.

4.2.2 Gradient Vector

We want to find a Bézier curve that its points have minimum squared distance form the point cloud. We first define the error between the point cloud and the curve. For that we interpolate all the points of the point cloud, for each consecutive pair we generate the interpolated points on the line that connects this pair together. Then we compute the normal line connecting $\overline{\mathbf{P}_0\mathbf{P}_2}$ to a point on the Bézier curve of the current iteration. We define its intersection with the interpolated curve as $\mathbf{I}_{t_i}$. Fig. 4.1 shows the conceptual diagram of this error finding method.

Our objective is to find $\mathbf{P}_1$ that minimize the error between the fitting curve and the point cloud. For a predefined set of knots $\{t_0, t_1, \cdots, t_k\}$ and $\lambda = 0$ the objective function is:

$$f = \sum_{j=1}^{k} \|\boldsymbol{P}(t_j) - \mathbf{I}_{t_j}\|^2 \tag{4.10}$$

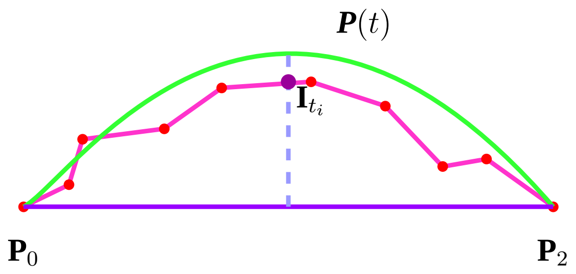


Figure 4.1 – The conceptual diagram of error between fitting Bézier curve and the point cloud.

As we said to minimize f, we iteratively update $\mathbf{P}_1$ by $\mathbf{D}$ in each iteration. For i^{th} iteration we have $\mathbf{P}_1^i = \mathbf{P}_1^{i-1} + \mathbf{D}$, the redefined objective function is:

$$f(\mathbf{P}_1^i) = f(\mathbf{P}_1^{i-1} + \mathbf{D}) = \sum_{j=1}^{k} \|\boldsymbol{P}(t_j) - \mathbf{I}_{t_j}\|^2 \tag{4.11}$$

To find $\mathbf{D}$ we differentiate Eq. (4.11) with respect to $\mathbf{D}$. Then we set it equal to zero.

$$\sum_{j=1}^{k} \mathbf{B}_1(t_j)\left[\boldsymbol{P}(t_j) - \mathbf{I}_{t_j}\right] = 0 \tag{4.12}$$

then

$$\mathbf{D} = \frac{\sum_{j=1}^{k} U(t_j)}{\sum_{j=1}^{k} \mathbf{B}_1^2(t_j)} \tag{4.13}$$

Where $U(t_j) = \mathbf{B}_1(t_j)\mathbf{I}_{t_j} - \mathbf{B}_0(t_j)\mathbf{B}_1(t_j)\mathbf{P}_0 - \mathbf{B}_1(t_j)^2\mathbf{P}_1 - \mathbf{B}_1(t_j)\mathbf{B}_2(t_j)\mathbf{P}_2$. This procedure is iterated until $f(\cdot)$ remains constant. We summarize Single Control Point Error Minimization curve fitting method in algorithm 4.

In Fig. 4.2(a) we made a noisy roundabout and we denoise it by Single Control Point Error Minimization method. But not always we can denoise a point cloud with just 3 control points, as it is represented in Fig. 4.2(a). For this kinds of point cloud we need B-spline with more control points. We discuss curve fitting based on Bézier curves with higher control points in the following section.

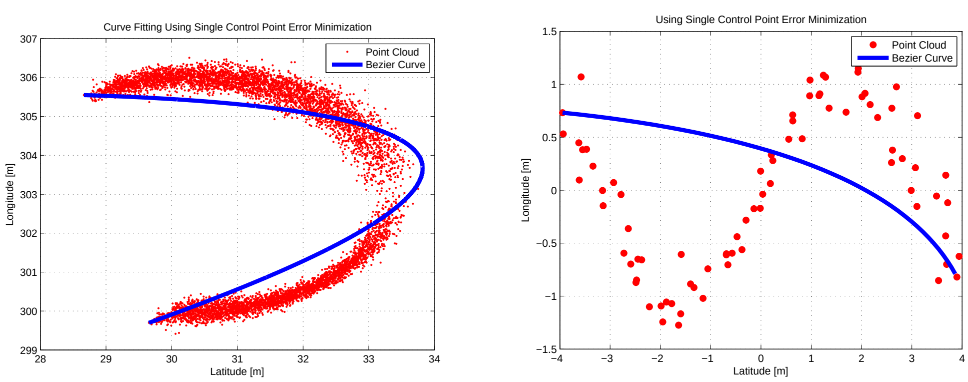
Algorithm 4 Single Control Point Error Minimization

$\mathbf{P}_0 \leftarrow \mathbf{x}_0$
$\mathbf{P}_1 \leftarrow \mathbf{x}_k$
$\mathbf{P}_1 \leftarrow (\mathbf{x}_{i_{max}} - \mathbf{B}_0(\hat{t})\mathbf{P}_0 - \mathbf{B}_2(\hat{t})\mathbf{P}_1/\mathbf{B}_1(\hat{t}))$

while $f_i > f_{i-1}$ **do**

$\quad \mathbf{D} \leftarrow \sum_{j=1}^{k} U(t_j)/\sum_{j=1}^{k} \mathbf{B}_1^2(t_j)$
$\quad \mathbf{P}_1^i \leftarrow \mathbf{P}_1^{i-1} + \mathbf{D}$
$\quad f_i \leftarrow \sum_{j=1}^{k} \|\boldsymbol{P}(t_j) - \mathbf{I}_{t_j}\|^2$

end while



(a) (b)

Figure 4.2 – Using iterative error minimization to fit a Bézier curve with 3 control points to a point cloud 4.2(a) is and example in which Single Control Point Error Minimization method works properly. 4.2(b) shows a case that this fitting algorithm does not work properly.

4.3 Multiple Control Points Error Minimization

We previously said because Single Control Point Error Minimization uses B-spline with three control points which does not work properly for many cases. We need a method that we could freely choose number of control points based on maxima and minima of the GPS trajectory. To approximate the scattered GPS data points, we need to starts with a proper initial B-spline and use iterative error minimization toward a best fitting curve to the point cloud. The advantage of this method is ability to freely choose number of control points. Thus we should have a better approximation with this method. Moreover, we utilize a new error term which can be devised to yield fast convergence and a better error approximation. Therefore, this error term is provided by a curvature-based quadratic approximation of squared distance of point cloud from the fitting curve [13]. In contrast with Single Control Point Error Minimization, there is no constraint for the ending control points of B-spline. This also contributes to a better approximation for the ending of the point could.

4.3.1 Second Order Approximation of Squared Distance

In Euclidean $\mathbb{R}^2$, we consider a curve $\boldsymbol{C}(t)$ with parameter $(c_1(t), c_2(t))$. The normal and tangent vector at each point of the curve $\boldsymbol{C}(t)$ are denoted by $\boldsymbol{N}$ and $\boldsymbol{T}$ respectively. For each point $\mathbf{x}$ the shortest distance to curve $\boldsymbol{C}(t)$ is given by $d = \|\boldsymbol{C}(t) - \mathbf{x}\|$ [9].

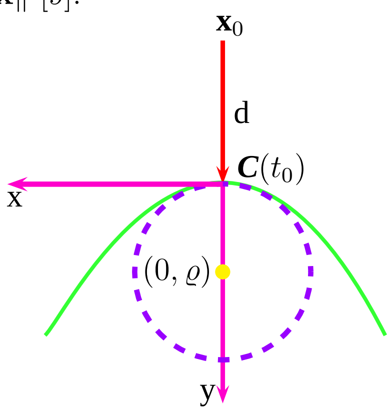


Figure 4.3 – Second order approximation of squared distance function to the curve $\mathbf{C}(t)$.

Consider a fixed point $\mathbf{x}_0$ and let $\boldsymbol{C}(t_0)$ be the normal footprint of $\mathbf{x}_0$ on $\boldsymbol{C}(t_0)$. In addition, consider the local Frenet frame of $\boldsymbol{C}(t_0)$ with its origin at $\mathbf{c}(t_0)$ and its coordinates axes parallel to the tangent and normal vector of $\boldsymbol{C}(t_0)$ at $\boldsymbol{C}(t_0)$. Coordinates of $\mathbf{x}$ in this axes are $(0, d)$. The coordinates of curvature center $\mathbf{k}(t_0)$ at $\boldsymbol{C}(t_0)$ are $(0, \varrho)$ (Fig. 4.3). ϱ is radius of $\boldsymbol{C}(t_0)$ which is inverse curvature and has the same sign as the curvature, depending on the orientation of the curve. Consider a point $\mathbf{x} = (x, y)$ in the neighborhood of $\mathbf{x}_0$, then the squared distance from $\mathbf{x}$ to $\boldsymbol{C}(t)$ is:

$$f(x, y) = \left(\sqrt{x^2 + (y - \varrho)^2} - |\varrho|\right)^2 \tag{4.14}$$

The second order Taylor approximation $\boldsymbol{F}_d$ of f at $(0, d)$ is: [9]:

$$\boldsymbol{F}_d(x, y) = \frac{d}{d - \varrho} x^2 + y^2 \tag{4.15}$$

4.3.2 Curve Fitting

We use iterative updates of control points to reach to the target curve and second order approximation of the squared distance to increase the convergence speed.

Given the B-spline curve of the current iteration $\boldsymbol{P}(t) = \sum_{ij=0}^{n} \mathbf{B}_j(t)\mathbf{P}_j$ with control points $\mathcal{P} = (\mathbf{P}_0, \mathbf{P}_1, \cdots, \mathbf{P}_n)$, the updated control points are $\mathcal{P}_+ = \mathcal{P} + \mathcal{D}$, where $\mathcal{D} = (\mathbf{D}_0, \mathbf{D}_1, \cdots, \mathbf{D}_n)$. Then the updated curve is denoted by $\boldsymbol{P}_+(t) = \sum_{j=0}^{n} \mathbf{B}_j(t)(\mathbf{P}_j + \mathbf{D}_j)$.

Suppose that footprint of $\mathbf{x}_i$ on the curve $\boldsymbol{P}(t)$ is $\boldsymbol{P}(t_i)$ and $\boldsymbol{T}_i$, $\boldsymbol{N}_i$ are tangent and normal vector of the curve at the footprint. ϱ is the curvature radius of $\boldsymbol{P}(t)$ at $\boldsymbol{P}(t_i)$ and $|d| = \|\boldsymbol{P}(t_i) - \mathbf{x}_i\|$. $d < 0$ when $\mathbf{x}_i$ and curvature center of $\boldsymbol{P}(t)$ at $\boldsymbol{P}(t_i)$ are on opposite side of $\boldsymbol{P}(t)$ and $d > 0$ if they are on the same side.

To obtain quadratic approximation of squared distance function from $\mathbf{x}_i$ to $\boldsymbol{P}(t)$, we consider differential translation from $\boldsymbol{P}(t)$ to $\boldsymbol{P}_+(t)$. Thus we can use Eq. (4.15) to approximate squared distance from $\mathbf{x}_i$ to $\boldsymbol{P}_+(t)$ as follows:

$$\boldsymbol{F}_i(\mathcal{D}) = \frac{d}{d - \varrho} \left[(\boldsymbol{P}_+(t_i) - \mathbf{x}_i)^T \boldsymbol{T}_i \right]^2 + \left[(\boldsymbol{P}_+(t_i) - \mathbf{x}_i)^T \boldsymbol{N}_i \right]^2 \tag{4.16}$$

In Eq. (4.16) if $0 < d < \varrho$ then $\boldsymbol{F}_i(\mathcal{D})$ is negative. So positive and semidefinite error term based on $\boldsymbol{F}_i(\mathcal{D})$ is defined as:

$$e_i(\mathcal{D}) = \begin{cases} \dfrac{d}{d - \varrho} \left[(\boldsymbol{P}_+(t_i) - \mathbf{x}_i)^T \boldsymbol{T}_i \right]^2 + \left[(\boldsymbol{P}_+(t_i) - \mathbf{x}_i)^T \boldsymbol{N}_i \right]^2 & \text{if } d < 0 \\ \left[(\boldsymbol{P}_+(t_i) - \mathbf{x}_i)^T \boldsymbol{N}_i \right]^2 & \text{if } 0 < d < \varrho \end{cases} \tag{4.17}$$

4.3.3 Regularization Factor

To have a smooth output of the estimated curve $\lambda f_s = \alpha F_1 + \beta F_2$ where[5] $\alpha,\ \beta \geq 0$ and $F_1,\ F_2$ are energy terms defined as:

$$F_1 = \int \|\boldsymbol{P}'(t)\|^2 \mathrm{d}t, \qquad F_2 = \int \|\boldsymbol{P}''(t)\|^2 \mathrm{d}t \tag{4.18}$$

4.3.4 Open B-spline

The problem of open curves are with their ending points. If all the end points could be projected to the curve, we could easily use our error term. But there are cases that all points of the point cloud cannot be mapped to a point on the curve. For these situations we specify a new error term.

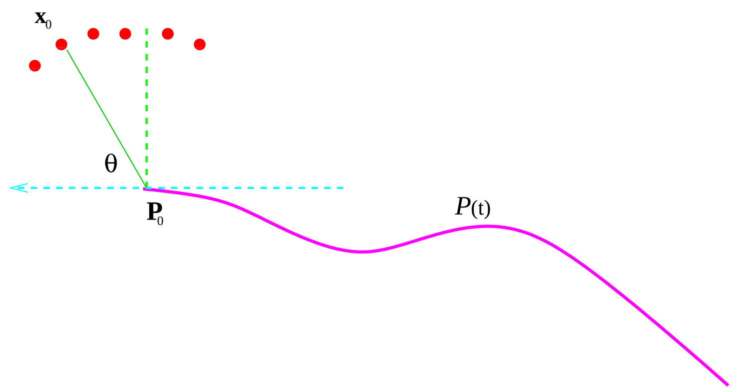


Figure 4.4 – Error term of ending points.

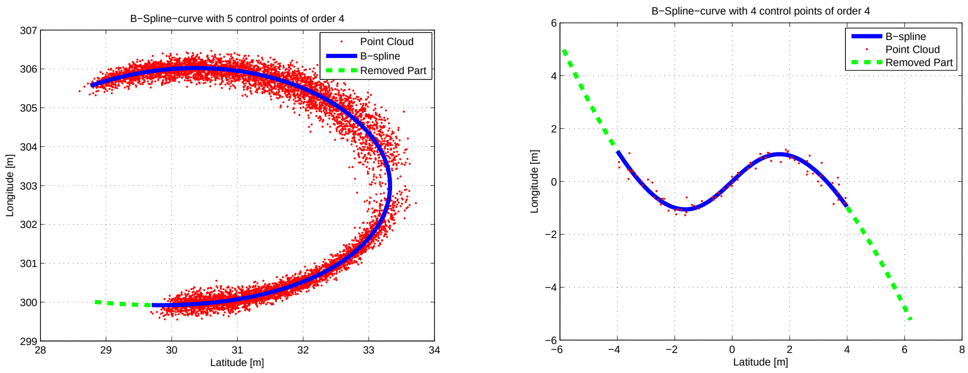
Fig. 4.4 is an example for cases that ending points cannot be mapped to a point on the fitting curve. T_0 is tangent vector of $\boldsymbol{P}(\mathrm{t})$ at end point $\boldsymbol{P}_0$. $\mathbf{x}_0$ is outer point and $\boldsymbol{P}_0$ is the nearest point in the curve to it. θ denotes the angel between T_0 and the vector $\mathbf{x}_0 - \boldsymbol{P}_0$ and $|\theta| < \pi/2$. Then the error term for ending point $\mathbf{x}_0$ is:

$$\begin{aligned} e_{endings,0}(\mathcal{D}) &= \cos\theta e_{d,0}(\mathcal{D}) + (1-\cos\theta)e_0(\mathcal{D}) \\ e_{d,0}(\mathcal{D}) &= \|\boldsymbol{P}_+(t_0) - \mathbf{x}_0\|^2 \end{aligned} \tag{4.19}$$

Ending points error term helps, through iterations, the fitting curve to be pulled towards the target end points. Note that at each iterations points that do not have footprints can be different. For points in point cloud that

[5] $\alpha = 0,\ \beta = 0.001$ for all the examples [13]

we have footprint we use the previous error term.



(a) (b)

Figure 4.5 – Using Multiple Control Points Error Minimization to fit a B-spline to a point cloud 4.5(a) We used a B-spline with 5 control points 4.5(b) B-spline with 4 control points is used. Dashed lines are parts that are not considered for the actual fitting curve.

In Fig. 4.5 we used the same point cloud as for Single Control Point Error Minimization, we see that this method works far better from the previous method and we can reconstruct better shapes with help of different number of B-spline control points. Because we removed the constraint that the ending control points of B-spline should be the end points of the point cloud, this method may extend the ending points, which are shown by dashed line in the figure, that should be removed from the fitting curve. Multiple Control Points Error Minimization is abridged in algorithm 5.

4.4 GPS Smoothing

In the previous section we compared two curve fitting algorithms for noisy data. We illustrated that *Multiple Control Points Error Minimization* works better for point clouds that have many maxima and minima. A typical GPS trajectory is composed of tops and bottoms so, this algorithm should work properly for smoothing of GPS points.

In Fig. 4.6 we used GPS mode to estimate the path of the user, then we used Multiple Control Points Error Minimization to have a smooth trajectory of the user path. We can also see in this figure that the smoothing method works properly for GPS data smoothing.

Algorithm 5 Multiple Control Points Error Minimization

Identify a proper initial B-spline

while $e_k(\mathcal{D}) \rangle$ *Threshold* **do**

$\quad F_1 \leftarrow \int \|\boldsymbol{P}'(t)\|^2 \mathrm{d}t$

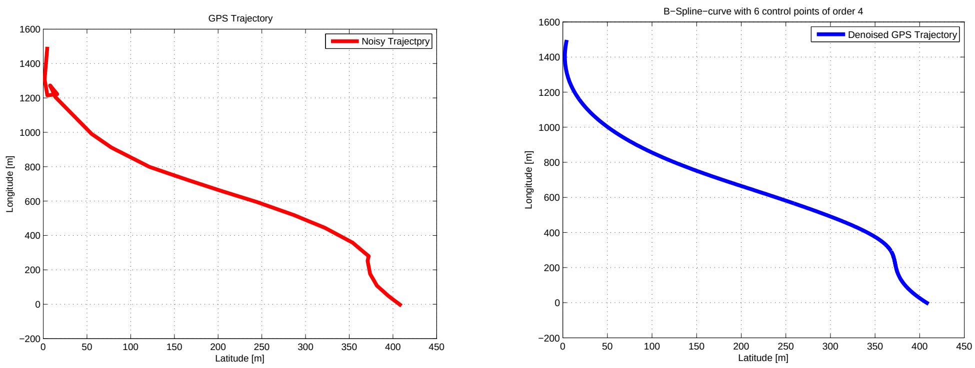
$\quad F_2 \leftarrow \int \|\boldsymbol{P}''(t)\|^2 \mathrm{d}t$

$\quad f(\mathcal{D}) \leftarrow \sum_{j=1}^{n} e_j(\mathcal{D}) + \alpha F_1 + \beta F_2$

$\quad \mathcal{D} \leftarrow \min_{\mathcal{D}} f(\mathcal{D})$

$\quad \mathcal{P}_+ \leftarrow \mathcal{D} + \mathcal{P}$

end while



(a) (b)

Figure 4.6 – Using Multiple Control Points Error Minimization curve fitting method to fit a B-spline to a GPS trajectory 4.6(a) Estimated GPS trajectory with help of GPS mode 4.6(b) The smoothed GPS trajectory.

4.5 Trajectory Estimation

There are many frequent trajectories in our every day life; for example, your way from home to office and vice versa. You most probably take the same path every day so, you will visit invariable GSM cell towers. The question arises in this matter is if we are in such paths, then why we should use GPS devices to find our position.

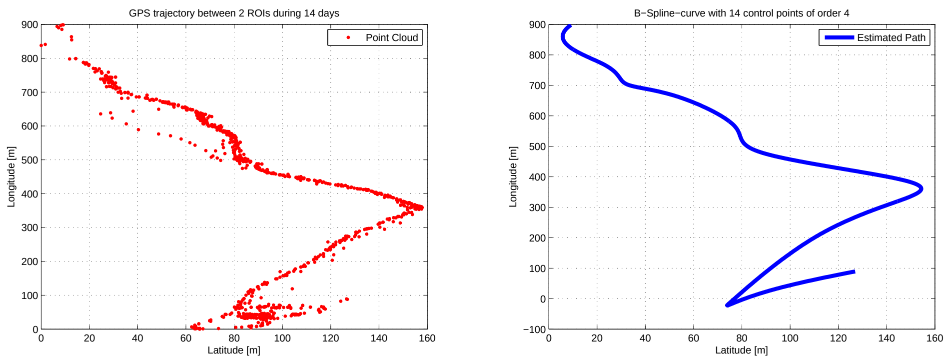
Your office and home are places that users visit them quite often such places are so-called Region Of Interests, which are unique per user. An effective way to obtain ROIs is based on WLANs and GSM cell towers. If the user spends specific time of his day or week seeing certain WLANs or cell towers, the location of the WLANs[6] can be one of the user ROIs. This experiment can be continued until finding all the ROIs. It is also possible to install a software on the mobile phone, which is able to store all the ROIs of its owner.

Now that we could find ROIs, it is possible to find the most probable path that the user travels among his ROIs. Then for the next time that you travel from one of your ROIs to the other, mobile phone can turn off its GPS sensor and navigate you based on the estimated path that it already has. Navigation is based on history of visited GSM cell towers and compare them with the current one so, it can predict your position and show your future path.

To estimate a user common path between two of his ROIs, we use the data campaign. We collect all of his taken trajectories between these two ROIs during his participation in campaign, then we use Multiple Control Points Error Minimization method to assign a smooth B-spline as his way between these two ROIs. We can store the control points in the mobile phone and the client can reconstruct the way by the control points.

Fig. 4.7 is the path that one of the participants gets from his home to office during 14 days. Then we used Multiple Control Points Error Minimization algorithm to find a B-spline that matches his path, as the most probable route that he takes from his home to his office. From now on there is no need of GPS navigator and the mobile phone can navigate the user for these ROIs.

[6]We discussed WLAN positions are more reliable than cell towers location so, we estimated the user position by WLANs location.



(a) (b)

Figure 4.7 – Using Multiple Control Points Error Minimization method to fit a B-spline to paths that a user has taken during 14 days between his ROIs, as his most probable future way. 4.7(a) Routes that the user traveled during these days from his home to his office 4.7(b) The most probable way that the user will take for his next travel between his ROIs.

Chapter 5

Conclusion

GPS is widely used in practice for positioning. As we have shown, although GPS have been designed in a way to be as accurate as possible, in practice there are some error sources that cause imprecise positioning and we explained the most common error sources. Moreover, because we deal with GPS receiver of a mobile phone, in which the quality of GPS receiver is not good, positioning has more error comparing to other GPS receivers. The other problem with GPS is its availability, that is, GPS is not accessible in places that people spend most of their time.

Against these backdrops, we have proposed a novel and efficient positioning method that does not just depend on GPS. Based on the availability of the GPS signal, our proposed method divides into two different operating mode. For situations that GPS is accessible, we suggested a probability density for GPS points. Locating of the user is based on maximization of joint probability of GPS points around the time that we want to find the location of the user. On the other hand, we spend most of our social life in places that GPS signal is not available for these kind of situations our proposed algorithm enters to its second mode of operation. With the help of ongoing data collection campaign by NRC-Lausanne, we could access to other information of users except from GPS data. This mode depends on WLANs and GSM cell towers that cover the user. For this mode we firstly should find position of WLANs and cell towers, WLANs position estimation is based on the guess that each campaign participant has from his visited WLAN. If the user does not have GPS points for a specific time duration, we use the second model and positioning depends on other information than GPS. We have shown empirically that our proposed positioning superior performance in both convergence and accuracy.

We have also proposed a smoothing algorithm for fitting B-spline curves to GPS points. The curve fitting algorithm, called Multiple Control Points Error Minimization, makes use of curvature information to give a close approximation of the squared distance function and this contributes to its simplicity and efficiency.

In addition, We used Multiple Control Points Error Minimization to smooth the estimated user trajectory. Interestingly, we have proposed a navigation method among the user's ROIs that unlike most other navigation systems does not depend on GPS. For this method we fit a B-spline to the most common path that the user takes among his ROIs. This navigation method arises as the consequence of participating in the data collection campaign that helps us to have a history of the user's trajectories.

We expect to see more studies on finding user's ROIs and the most probable path problems both theoretically and from the view-point of mobile phone applications. Although we have tried to consider all the available information to estimated the user position, an immediate extension is to consider other sources like discovered BTs by mobile phones in each time instance to improve the precession of the localization. Other problems include to optimize the knots and weights of the curve fitting method. In general, to improve the smoothness of the trajectories by map matching so that they have more compatibility with actual routes.

Bibliography

[1] Shane Ahern, Marc Davis, Simon King, Mor Naaman, and Rahul Nair. Reliable, User-Contributed GSM Cell-Tower Positioning Using Context-Aware Photos. *Th Eight International Conference on Ubiquitous Computing*, 2006.

[2] Reha Metin ALKAN. GPS-Single Point Positioning Without Selective Availability. In *Proceedings of U.S. Hydrographic Conference*, pages 22–24. The Hydrographic Society of America, 2001.

[3] Anthony LaMarca1 et al. Place Lab: Device Positioning Using Radio Beacons in the Wild. In *Proceedings of the Third International Conference on Pervasive Computing*, pages 116–133, Munich, Germany, 2005. Springer.

[4] Garmin International. About GPS. `http://www.garmin.com/aboutGPS/`, 2001.

[5] Niko Kiukkonen. Technical Report: Data Collection Campaign. *Technical report*, 2009.

[6] Dr. Anja Köhne and Dr. Michael Wößner. Sources of Errors in GPS. `http://www.kowoma.de/en/gps/errors.htm`, 2009.

[7] Binghao Li, James Salter, Andrew G.Dempster, and Chris Rizos. Indoor Positioning Techniques Based on Wireless LAN. In *LAN, First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications*, pages 13–16, Sydney, Australia, 2006.

[8] Min Mun, Sasank Reddy, Katie Shilton, Nathan Yau, Jeff Burke, Deborah Estrin, Mark Hansen, Eric Howard, Ruth West, and Pter Boda. PEIR, The Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research. In *International Conference On Mobile Systems, Applications And Services*, pages 55–68, Poland, 2009. Association for Computing Machinery.

[9] Helmut Pottmann and Michael Hofer. Geometry of the Squared Distance Function to Curves and Surfaces. In *Visualization and Mathematics III*, pages 223–244. Springer, 2003.

[10] Matthew Rabinowitz and Jr. James J. Spilker. A New Positioning System Using Television Synchronization Signals. *Rosum Corporation white paper*, 2004.

[11] Timothy Sohn, Alex Varshavsky, Anthony Lamarca, Mike Y. Chen, Tanzeem Choudhury, Ian Smith, Sunny Consolvo, Jeffrey Hightower, William G. Griswold, and Eyal De Lara. Mobility Detection Using Everyday GSM Traces. In *Proceedings of the Eighth International Conference on Ubiquitous Computing*, pages 212–224. Springer, 2006.

[12] Martin Vetterli and Jelena Kovačević. *Wavelets and Subband Coding* , pages 25–27. Prentice Hall, USA, 2007.

[13] Wenping Wang, Helmut Pottmann, and Yang Liu. Fitting B-Spline Curves to Point Clouds by Squared Distance Minimization. *ACM Transactions on Graphics*, 25, 2004.

[14] Ji wung Choi and Gabriel Hugh Elkaim. Bézier Curve for Trajectory Guidance. *World Congress on Engineering and Computer Science*, 2008.

[15] Junyeong Yang and Hyeran Byun. Curve Fitting Algorithm Using Iterative Error Minimization for Sketch Beautification. In *International Conference on Pattern Recognition*, Florida, USA, 2009.

[16] Daniel Zwillinger. *Standard Mathematical Tables and Formulae* , pages 32–35. CRC Press, 30th edition, 1996.