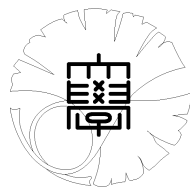


# 修 士 論 文

## Improvement of Pedestrian Positioning Accuracy using Height Aided GNSS and Pedestrian Dead Reckoning (Height Aided GNSS と歩行者デッド レコニングによる歩行者測位精度の 向上)

指導教員

上條 俊介 准教授



東京大学大学院  
情報理工学系研究科  
電子情報学専攻

学籍番号・氏名

48-146459 黄 宇陽

提出日

2016 年 2 月 4 日





## 概要

Currently, There are a lot of technologies could be utilized to provide the pedestrian navigation system in urban environment. For example, the Wi-Fi positioning system (WPS), pedestrian dead reckoning (PDR) system and Global positioning system(GPS). However, in the certain environment which is called urban canyon, none of those system could be satisfied enough to provide accurate (less than 10 meters positioning mean error) pedestrian navigation solution. The WPS is usually only used for providing rough positioning information. And it is hard for WPS to achieve averagely less than 10 meters positioning accuracy in outdoor environment. Instead, in order to get more accurate positioning result, the GPS is usually used in this scenario. However, in urban canyon environment, GPS signals could easily be blocked by tall building and there are a lot of non-line-of-sight and multipath signals come in which will result in serious positioning accuracy degrading for GPS system. The conventional standalone GPS system usually cannot achieve 10 meters positioning accuracy in urban environment. Besides, even though the PDR system could provide relatively accurate trajectory, the standalone PDR system could not provide absolute positioning result. Moreover, the PDR system will suffered from error accumulation and could not perform self-correction. As a result, the standalone PDR system is also not suitable to provide accurate pedestrian navigation service. In this paper, we studied how to improve the accuracy of the pedestrian navigation system in urban canyon environment. Our objective is to provide averagely less than 10 meters positioning accuracy for pedestrian in urban environment. We use altitude map based method to first improve the positioning accuracy of the conventional GPS system. Moreover, we use the same information to evaluate the reliability of GPS result. Finally, we integrate the proposed GPS method with PDR system and achieved around 6.5 meters positioning accuracy in urban environment.





# 目次

第 1 章	Introduction	1
1.1	Background . . . . .	1
1.2	Objective . . . . .	5
1.3	Thesis Structure . . . . .	5
第 2 章	Pedestrian Navigation System in Urban Canyon and Related Works	6
2.1	GPS Positioning Algorithm . . . . .	6
2.2	Problem and Error Source of Global Positioning System . . . . .	11
2.3	Related Work About Improving GNSS Positioning Accuracy In Urban Environment . . . . .	13
2.4	PDR Algorithm and Related Works . . . . .	16
2.5	Problem and Error Source of PDR System . . . . .	21
2.6	Fundamental of Integrated Navigation System and Related Works . . . .	22
第 3 章	The Altitude Map Based Positioning Algorithm	28
3.1	The Proposed Altitude Map Based Pseudorange Correction Method . . .	28
3.2	Experiment Result . . . . .	33
第 4 章	Integration of PDR with Height Aided GNSS method	40
4.1	The Proposed PDR System . . . . .	40
4.2	Evaluation of Purposed PDR System . . . . .	43
4.3	The Proposed PDR with Height Aided GNSS Framework . . . . .	45
4.4	Experiment Result . . . . .	47
第 5 章	Conclusion and Future Works	54
	Thanks	56
	発表文献	57
	Reference	58



# 

1.1	The Conventional GNSS result in urban area. . . . .	2
1.2	The positioning result of standalone PDR system with initial position provided . . . . .	3
1.3	The integration result of PDR + conventional GNSS solution. . . . .	4
2.1	The constellation of GPS satellites.(source:www.faa.gov) . . . . .	7
2.2	The illustration of GPS trilateration. . . . .	7
2.3	The explanation of time relationships in GPS [1]. . . . .	8
2.4	GPS signal must go through ionosphere and troposphere to reach the planet surface. . . . .	12
2.5	The multipath and NLOS effects in an urban canyon. (a) Multipath effect. (b) NLOS propagation [2]. . . . .	14
2.6	The illustration of 3D-GNSS method [2]. . . . .	15
2.7	The pseudo satellite placed in the center of the earth. Its psudorange is the altitude from the 3D city model. . . . .	16
2.8	The overview of step based PDR system [3]. . . . .	17
2.9	The x-y-z coordinate and E-N-U coordinate. . . . .	18
2.10	The placement of MEMS. . . . .	19
2.11	The vertical acceleration data of a waling pedestrian [4]. . . . .	19
2.12	Extraction of motion direction with PCA [5]. . . . .	21
2.13	The error accumulation of PDR system. . . . .	22
2.14	The idea of using kalman fiter to fusion PDR and GPS . . . . .	24
2.15	Relative geometry and dilution of precision: (a) geometry with low DOP, and (b) geometry with high DOP [6]. . . . .	25
2.16	The PCA cluster for GPS result in different accuracy category [7]. . . . .	26
3.1	Relationship between horizontal and vertical errors of GNSS positioning. . . . .	29
3.2	Relationship between horizontal vertical errors of GNSS positioning in urban canyon. . . . .	29
3.3	Relationship between geoid height, orthometric height, and ellipsoid height. . . . .	30

3.4	The fault satellite detection [6]. . . . .	31
3.5	The flowchart of suspicious satellite signal detections. . . . .	32
3.6	The satellite correction procedure. . . . .	33
3.7	The u-blox M8 receiver used in the experiment. . . . .	34
3.8	The experiment environment and walking trajectory. . . . .	34
3.9	The WLS and WLS + height aiding comparison for one single test. . . .	35
3.10	The related epoch comparison between WLS and WLS + HA . . . . .	36
3.11	One epoch case study for WLS + height. . . . .	36
3.12	The pseudorange error correction value and related horizontal and vertical errors. . . . .	37
4.1	Demonstration of the step detection using smoothed accelerations in vertical direction. . . . .	42
4.2	Illustration of the verification of forward direction using forward acceleration. . . . .	43
4.3	The Google Nexus 5 smartphone. . . . .	44
4.4	Positioning result of the smartphone-based PDR with 4 objects. . . . .	45
4.5	Error of heading angle of the two test trails of Object 1. . . . .	46
4.6	The results of altitude map based evaluation method. The <i>GT</i> is the ground truth. . . . .	48
4.7	The horizontal positioning error and accuracy value calculated by Google Nexus 5. . . . .	49
4.8	The horizontal error and conventional accuracy and proposed accuracy calculated for WLS solution. . . . .	50
4.9	The integration result part A. . . . .	52
4.10	The integration result part B. . . . .	53

# 表目次

2.1	The type and properties of GPS pseudorange errors. . . . .	13
2.2	Feature and localization error relationship [7]. . . . .	27
3.1	The mean error and standard deviation comparison of WLS and WLS + height aiding result for total 15 case the unit is meter. . . . .	38
4.1	Sensor embedded in Google Nexus 5. . . . .	41
4.2	The PDR parameters setted in the experiment. . . . .	44
4.3	Performance of the smartphone-based PDR systems. . . . .	44
4.4	The color and related $HR$ . The $T_r$ is set to 10. . . . .	48



## 第 1 章

# Introduction

In this research, we studied how to improve the positioning accuracy of pedestrian navigation system in urban canyon environment. In this chapter, the background, objective and structure of this thesis are discussed.

### 1.1 Background

In urban environment, how to provide accurate pedestrian positioning information is becoming more and more important currently. In big cities, accurate positioning information is required by all kinds of applications such as pedestrian navigation, shop recommendation and autonomous driving etc.. Even though there are many positioning systems could provide positioning result in this environment, such as Wi-Fi Positioning System (WPS) Global Positioning System (GPS) and Pedestrian Dead Reckoning (PDR) system, the positioning accuracy of those type of system is usually not accurate enough (usually larger than 10 meters).

For WPS, it is first developed for simple indoor environment. Even though it can also be used in outdoor environment, and some companies such as Google and Apple already use WPS to provide location service for their users, the positioning accuracy of WPS system is varied from 10 to 40 meters. As a result, such system can not be utilized to provide accurate positioning service. When accurate positioning service are needed in outdoor environment, The GPS system is usually the first choice. For GPS system, it is the most developed satellite-based positioning system in the world, and has been wildly used in all kinds of devices and applications. GPS is capable of providing reliable position information in most cases, which is extremely helpful in the researches and daily applications. There are many kinds of positioning methods based on GPS satellites, for instance, D-GPS (Differential-GPS uses several base stations to facilitate the positioning), A-GPS (Assisted-GPS is designed for mobile terminal use) and PPP (Precise Point Positioning uses signal frequency instead of distance to position). The GPS system usually



## 2 第1章 Introduction

need more than 4 satellites to calculate the positioning result. If the number of received satellite is small and the received signals contained errors, the positioning accuracy of GPS system will degrade seriously. In urban canyon environment, the GPS signals are easily blocked by tall buildings and the signals could also come to receiver though reflection path. In other words, the GPS signal could easily suffered from non-line-of-sight (NLOS) or multipath situations. These two type of errors could not be solved by using only conventional GPS method. As a result, in urban canyon environment, the positioning accurate of standalone GPS system is larger than 10 meters. Even tough in current world, we can use Russian GLONASS or Japanese QZSS, China Beidou(etc.) systems to increase the satellite numbers, in other word, to form Global Navigation Satellite System(GNSS), The satellites in GNSS system also suffered the same problem as standalone GPS system. Fig. 1.1 shows a typical conventional GNSS positoning result using commercial level GNSS receiver embedded in Google Nexus 5 smartphone. Such GNSS receiver could utilized the GPS, GLONASS and Beidou satellites. Even tough the receiver could observe averagely more than 7 satellites during the experiment. Because of the NLOS and Multipath effect, the positioning accuracy of Google Nexus 5 is still around 15 meters. It can seen that the GNSS solutions are still far from the ground truth. Usually, in middle class urban environment, there are more than 2 NLOS and multipath contaminated satellite could be received, those measurements will lead large shift in final GNSS results.

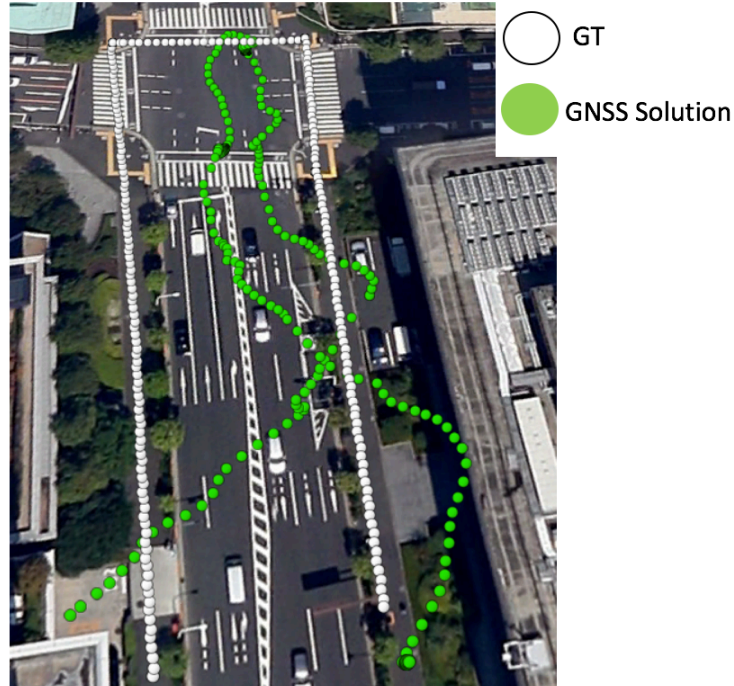


图 1.1. The Conventional GNSS result in urban area.

Besides GPS and WPS, The PDR system could also be utilized provided pedestrian navigation service. The PDR system is using IMU sensors to calculate the movement of pedestrian and is another type of navigation system. For PDR, the dead reckoning technology is using inertial measurement unit (IMU), which consisted of accelerator, magnetometer and gyroscope, to detect every step of the pedestrian and also calculated step length and moving direction of the pedestrian. The PDR system has an advantage that it could produce continues and smooth positioning trajectory. However, the PDR system could not provide absolute positioning result, such system need other system, such as GPS or WPS to provide initial position. Moreover, the PDR system will be suffered from the problem of error accumulation. The Fig. 1.2 shows one trial of standalone PDR system in urban environment with initial position provided manually. We can see that even tough the trajectory is smooth and the shape of PDR result is very close to the ground truth, the direction bias of PDR make the trajectory goes far from the ground truth gradually and the PDR system could not correct such errors by itself.

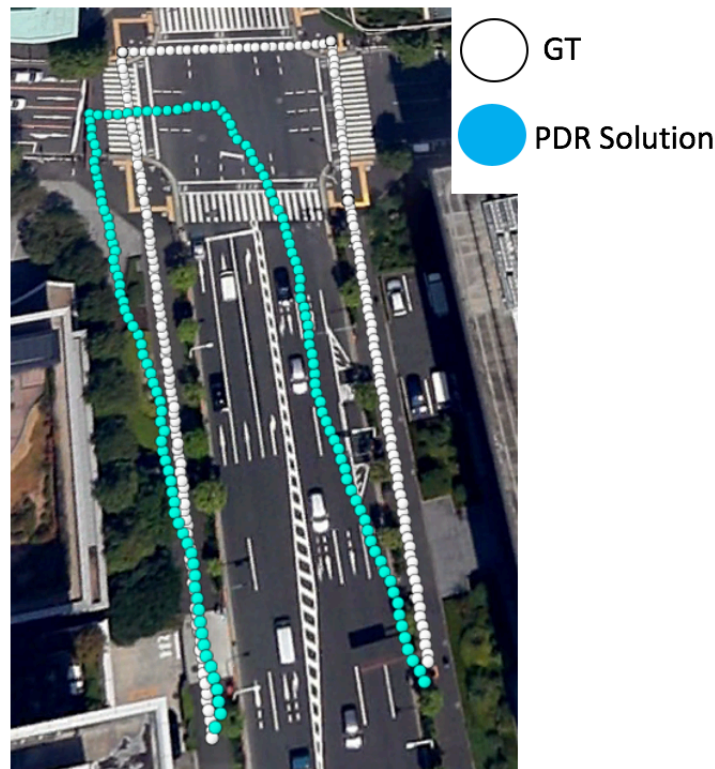


图 1.2. The positioning result of standalone PDR system with initial position provided

As a result, the PDR system is usually considered a supplement system but not a standalone positioning system. The PDR + WPS or PDR + map matching system is already developed for indoor navigation purpose and already could achieve less than 3 meters in

indoor environment. Such system proofed the effectiveness of integrating system. In fact, for outdoor environment navigation, there are already some researchers tried to integrate PDR system with WPS or GPS. However, for integration purpose, we need some ways to evaluate the reliable of observation result (usually the GPS result). Even though theoretically we can evaluate the reliability of GPS positioning result, the conventional formula does not take NLOS or Multipath error into consideration. Therefore, using conventional way to evaluate the GPS positioning result is not reliable in urban canyon environment. If we can not know the reliability of GPS positioning results, simply integrate the GPS and PDR could not achieve enough positioning result improvement. Fig. 1.3 shows the PDR(Fig. 1.2) + GNSS(Fig. 1.1 integration result under an conventional kalman filter framework. As shown in Fig. 1.3, we can not see any improvement of the final positioning result because the conventional GPS accuracy calculation method is not reliable enough so there is no way to know when the GNSS positioning result is reliable.

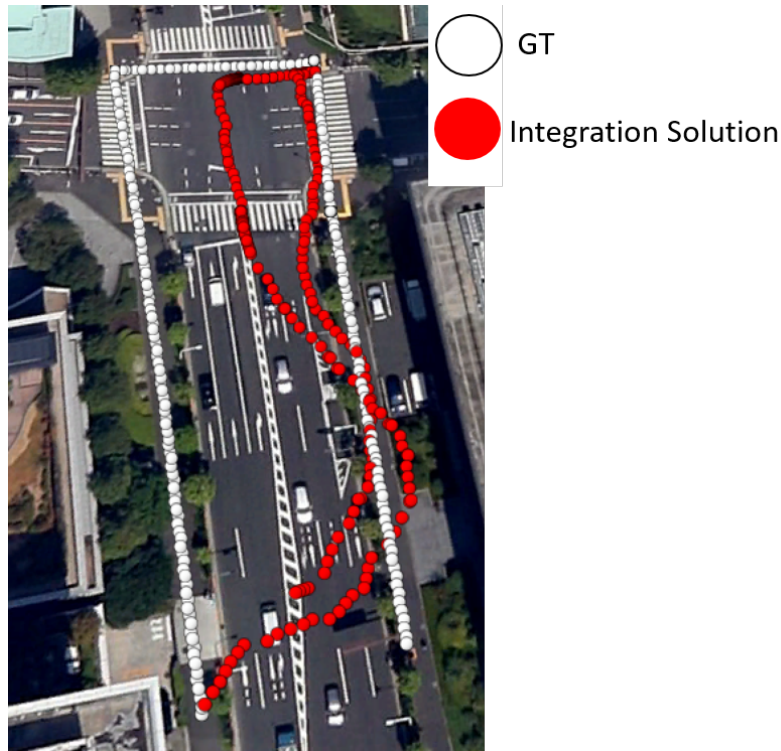


图 1.3. The integration result of PDR + conventional GNSS solution.

In order to improve the positioning accuracy of those positioning methods, some adaptations should be made.

## 1.2 Objective

In this research, we studied the GPS system in a typical urban canyon environment. We tried to improve the positioning accuracy of both standalone GPS system and GPS + PDR integrating system. We want achieve less than 10 meters positioning accuracy in a typical urban canyon environment. There are mainly two works be done in this thesis.

1. Firstly, we try to improve the positioning accuracy of GPS system with the help of an altitude map. With the help of the altitude map, we can estimate the vertical positioning error of positioning result. With the help estimated vertical positioning error, we can correct the erroneous satellite signals and therefore reduce the positioning error horizontally.
2. In order to achieve further improvement, we integrate the GPS and PDR system. We also evaluate the reliability of the GPS positioning result with the help of altitude map. We integrate the GPS and PDR system under a classical kalman filter framework. Our integrating system could achieve around 6.5 meters mean error positioning accuracy in urban environment.

## 1.3 Thesis Structure

The Structure of this paper listed as follows:

In Chapter 2, the basic concepts of GPS, PDR and the integrating system are introduced and the related works are presented.

In Chapter 3 , we introduce the problems of GPS in the urban canyon and the proposed altitude map based GPS signal correction method and show how to improve the positioning accuracy in urban canyon environment.

In Chapter 4, we first introduce how to calculate the GNSS positioning accuracy using altitude map. After that, the integration of the integration of the proposed altitude map based GPS method and PDR system is presented and evaluated.

In Chapter 5, the conclusions and future works are presented.

## 第 2 章

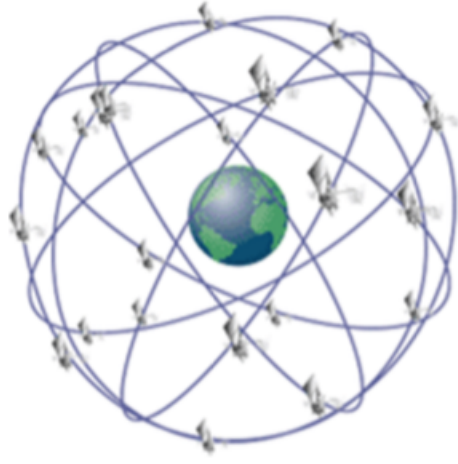
# Pedestrian Navigation System in Urban Canyon and Related Works

In this chapter, we will introduce the fundamental of the pedestrian navigation systems in urban canyon environment. We will mainly introduce the basic concept and knowledge about Global Positioning System (GPS), Pedestrian Dead Reckoning (PDR) system and also PDR + GPS integrated navigation system. We will also introduce the related works has been done relate to evaluate or improve the positioning accuracy of those systems.

### 2.1 GPS Positioning Algorithm

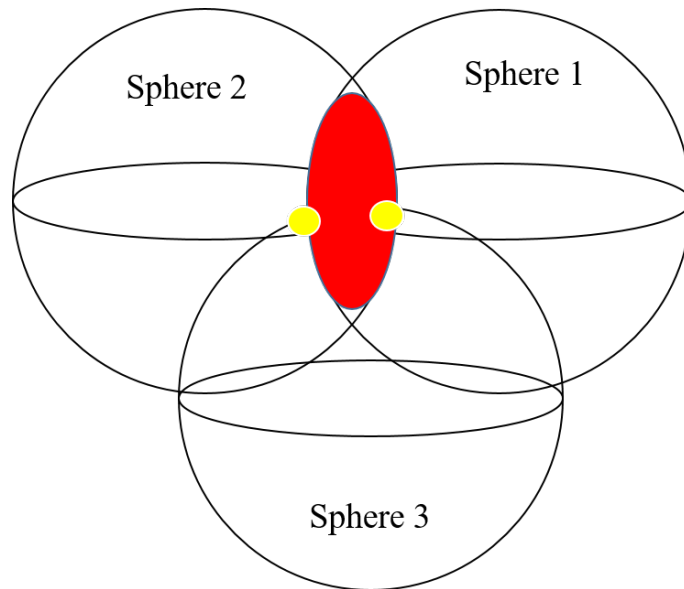
GPS is short for Global Positioning System. GPS is initially developed by America in 1973 and various kinds of researches have been done in this field. Currently, totally 31 satellites has been launched to form the GPS system. The constellation of GPS could be viewed in Fig. 2.1. Beside America, other countries also launched their own satellite navigation system, for example, Russia has GLONASS, Europe has Galileo, China has Bei-Dou and Japan has QZSS etc.. Those satellite has formed the modern Global Navigation Satellite System (GNSS). Most of satellite systems have the same fundamental positioning algorithm. In this section, we will mainly introduce the fundamental algorithm of GPS.

The GPS standalone positioning use only one GPS receiver to calculate user position. It is the most simple GPS positioning algorithm. It is also be called conventional GPS method. The basic mathematics for GPS standalone positioning trilateration. If the coordinates of the satellites are known and the distance between the receiver and satellites are also known, The three sphere could be draw, as illustrated in Fig. 2.2. The intersection of sphere 1 and sphere 2 is a red circle. The red circle intersects the sphere 3 at two yellow points. Usually, only one yellow point from those 2 is reasonable because another is in the space. As a result, in ideal condition, using 3 GPS satellites could provide reliable positioning result. However, in the real world, usually we need at least 4 satellites to



☒ 2.1. The constellation of GPS satellites.(source:www.faa.gov)

calculate the user positioning. The reason is because the receiver uses the signal propagation time to calculate the distance from a certain satellite to the receiver. Nevertheless, the clock in the GPS is not synchronized with the clock in the receiver. The small time shift will result in clock offset and brings errors to the measured distance. As a result, we have 4 unknown parameters: user altitude, longitude, latitude and clock bias (the offset). Mathematically, in order to solve 4 identically unknown parameters, at least four equation is needed, which stands for measurement from 4 satellites.



☒ 2.2. The illustration of GPS trilateration.

Before talking about the formula to calculate the receiver position. We need firstly introduce the receiver measurement, which is called “pseudorange”. The pseudorange is

the receiver calculated distance from receiver to satellite. It is computed by using the propagation time, which is the different between the time signal is sent and the time the signal is received. However, as previously discussed, the receiver clock is not synchronized with the GPS system time. Therefore, the distances computed using the speed of light to multiply propagation time are different with the true geometric distances.

For instance, if a signal from satellite  $n$  arrived at the receiver at GPS time  $t$  and if the signal propagation time is  $\tau$ . The time when the signal was sent is  $t^{SV}(t - \tau)$  and the time when the signal is received is  $t^r(t)$ . The pseudorange from satellite  $n$  can be calculated as follows:

$$\rho_n(t) = c[t^r(t) - t_n^{SV}(t - \tau)] \quad (2.1)$$

In Eqn. 2.1, both  $t$  and  $\tau$  are unknown. The relationship of receiver time  $t^r(t)$  and satellite time  $t_n^{SV}(t - \tau)$  is as follows:

$$\begin{aligned} t^r(t) &= t + \delta t^r(t) \\ t_n^{SV}(t - \tau) &= (t - \tau) + \delta t_n^{SV}(t - \tau) \end{aligned} \quad (2.2)$$

Where,  $\delta t^r(t)$  is the receiver clock with GPS time correction,  $\delta t_n^{SV}(t - \tau)$  is the satellite clock with GPS time correction. By considering this, the pseudorange computation equation could be as follows

$$\begin{aligned} \rho_n(t) &= c[t + \delta t^r(t) - ((t - \tau) + \delta t_n^{SV}(t - \tau))] + \varepsilon(t) \\ &= c\tau + c[\delta t^r(t) - \delta t_n^{SV}(t - \tau)] + \varepsilon(t) \end{aligned} \quad (2.3)$$

In this formula,  $\varepsilon$  means the errors that can not be modeled. Fig.2.3 denotes the relationship.

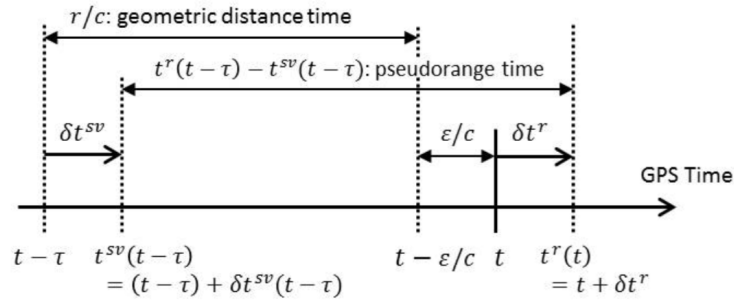


图 2.3. The explanation of time relationships in GPS [1].

Moreover, geometry range between satellite and receiver can be displayed as follows:

$$r_n(t, t - \tau) = c\tau - I_n(t) - T_n(t) \quad (2.4)$$

where  $I_n(t)$  and  $T_n(t)$  denote ionosphere and troposphere delay, which happened during the signal propagated through atmosphere. With the help of Eqn. 2.4, the pseudorange could also be represented as follows:

$$\rho_n = r_n + c[\delta t^r - \delta t_n^{sv}] + I_n + T_n + \varepsilon_n \quad (2.5)$$

We use Eqn. 2.5 to represent pseudorange in this thesis. After the definition of pseudorange, we can discussed about the algorithm used to calculated receiver positioning. This algorithm could be found in [1].

With the definition of pseudorange, we can introduce the weighted least square (WLS) algorithm to provide the GPS positioning solution. At a certain epoch  $t$ , we assume the GPS receiver's position is  $\mathbf{x} = [x, y, z]^T$  and the position of  $n$ th satellite is  $\mathbf{x}_n = [x_n, y_n, z_n]^T$ . The geometry range  $r$  between GPS receiver and satellite can be represent as follows:

$$r_n = \sqrt{(x_n - x)^2 + (y_n - y)^2 + (z_n - z)^2} = \|\mathbf{x}_n - \mathbf{x}\| \quad (2.6)$$

The Eqn. 2.3 becomes:

$$\begin{aligned} \rho_n &= \|\mathbf{x}_n - \mathbf{x}\| + c[\delta t^r - \delta t_n^{sv}] + I_n + T_n + \varepsilon_n \\ \rho'_n &= \|\mathbf{x}_n - \mathbf{x}\| + b + \tilde{\varepsilon}_n \end{aligned} \quad (2.7)$$

The  $\rho'_n$  is the corrected pseudorange in which satellite clock offset, ionospheric delay, tropospheric delay has been corrected in a certain degree. We simply set  $c\delta t^r$  as  $b$ , which represent the user-side clock bias. The  $\tilde{\varepsilon}$  is the error residue after error correcting the pseudorange by error model. In the above equation, we have 4 unknown which is  $x, y, z, b$ . In order to have unique solution, we need at least 4 satellite to be received. We can use Newton-Raphson method to solve the equation. The detail of this method is showed as follows.

Assume, we have a initial assign position  $\mathbf{x}^0 = [x^0, y^0, z^0]^T$ .  $b^0$  is the initial assigned clock bias. The true receiver position should be  $\mathbf{x} = \mathbf{x}^0 + \delta\mathbf{x}$  and the true clock bias should be  $b = b^0 + \delta b$ . The initial corrected pseudorange could be represented as  $\delta\rho_n'^0 = \|\mathbf{x}_n - \mathbf{x}^0\| + b^0$ . We can derive the following equation.

$$\begin{aligned} \delta\rho_n &= \delta\rho'_n - \delta\rho_n'^0 \\ &= \|\mathbf{x}_n - \mathbf{x}\| - \|\mathbf{x}_n - \mathbf{x}^0\| + (b - b^0) + \tilde{\varepsilon}_n \end{aligned} \quad (2.8)$$



By using Taylor expansion, the equation becomes:

$$\begin{aligned}
\|\mathbf{x}_n - \mathbf{x}\| &= \sum_{\alpha} \frac{(\mathbf{x} - \mathbf{x}_0)^{\alpha}}{\alpha!} \partial^{\alpha} \|\mathbf{x}_n - \mathbf{x}\|(\mathbf{x}_0) \\
&\simeq \|\mathbf{x}_n - \mathbf{x}^0\| + \partial \|\mathbf{x}_n - \mathbf{x}\|(\mathbf{x}_0) \\
&= \|\mathbf{x}_n - \mathbf{x}^0\| - \frac{\mathbf{x}_n - \mathbf{x}_0}{\|\mathbf{x}_n - \mathbf{x}_0\|} \cdot \delta \mathbf{x}
\end{aligned} \tag{2.9}$$

When replacing Eqn. 2.9 into Eqn. 2.8, we can derive the following equation:

$$\begin{aligned}
\delta \rho_n &\simeq - \frac{(\mathbf{x}_n - \mathbf{x}^0)}{\|\mathbf{x}_n - \mathbf{x}^0\|} \cdot \delta \mathbf{x} + \delta b + \tilde{\varepsilon}_n \\
&= -\mathbf{l}_1 \cdot \delta \mathbf{x} + \delta b + \tilde{\varepsilon}_n
\end{aligned} \tag{2.10}$$

We have N equations for N satellites:

$$\begin{aligned}
\delta \boldsymbol{\rho} = \begin{bmatrix} \delta \rho_1 \\ \delta \rho_2 \\ \vdots \\ \delta \rho_N \end{bmatrix} &= \begin{bmatrix} (-\mathbf{l}_1)^T & 1 \\ (-\mathbf{l}_2)^T & 1 \\ \vdots & 1 \\ (-\mathbf{l}_N)^T & 1 \end{bmatrix} \begin{bmatrix} \delta \mathbf{x} \\ \delta b \end{bmatrix} + \tilde{\boldsymbol{\varepsilon}} \\
&= G \begin{bmatrix} \delta \mathbf{x} \\ \delta b \end{bmatrix} + \tilde{\boldsymbol{\varepsilon}}
\end{aligned} \tag{2.11}$$

If  $G$  could be inversed, we can solve Eqn. 2.11 as follows:

$$\begin{bmatrix} \delta \mathbf{x} \\ \delta b \end{bmatrix} = G^{-1} \delta \boldsymbol{\rho} \tag{2.12}$$

We have  $-\mathbf{l}_n = \frac{(\mathbf{x}_n - \mathbf{x}^0)}{\|\mathbf{x}_n - \mathbf{x}^0\|}$ . It represents the unit vector from receiver to satellite. Usually, the  $rank(G)$  is greater than 4, as a result, the Eqn. 2.11 is over determinate. In this situation, we need to apply the least square method. We need to minimize the following equation.

$$\min \left\| \delta \boldsymbol{\rho} - G \begin{bmatrix} \delta \hat{\mathbf{x}} \\ \delta \hat{b} \end{bmatrix} \right\|^2 \tag{2.13}$$

The solution of least square method is as follows:

$$\begin{bmatrix} \delta \hat{\mathbf{x}} \\ \delta \hat{b} \end{bmatrix} = (G^T G)^{-1} G^T \delta \boldsymbol{\rho} \tag{2.14}$$

The  $G^T G$  is derived by considering satellite geometry. We update the  $\mathbf{x}$  and  $b$  as  $\mathbf{x}^1 = \mathbf{x}^0 + \delta\mathbf{x}$ ,  $b^1 = b^0 + \delta b$ , and apply the same procedure several times until  $\delta\mathbf{x} \approx 0$  to get the final solution. Usually, the equation will get converge after 3 or 4 iteration.

However, the importance of satellite to final solution should be different. For example, the satellite at high elevation angle is get less chance to contain error than that at low elevation angle. By considering the importance of satellite, We can assign weight to each satellite and make the least square solution become weighted least square solution:

$$\begin{bmatrix} \delta\hat{\mathbf{x}} \\ \delta\hat{b} \end{bmatrix} = (G^T W G)^{-1} G^T W \delta\mathbf{p} \quad (2.15)$$

The  $W$  is diagonal matrix that  $\{w_1, w_2, \dots, w_N\}$  is at each diagonal. In this thesis, we use WLS method to calculate the final solution.

## 2.2 Problem and Error Source of Global Positioning System

As mentioned above, GPS uses pseudorange in positioning. The pseudoranges are computed using signal propagation time. During the calculation, there are several places where errors could occurs. Those errors could be random or have some patterns. The errors that have pattern are relatively easy to deal with, because we can formulate model to estimate the errors. Others errors that does not have patterns will be hard to analysis.

Firstly, the satellite side could have some errors, which are satellite clock bias and satellite coordinate bias. Those errors has pattern and could be pre-estimate, the ephemeris sent by the satellites contains the clock correction information. The ephemeris information sent by the satellites contains the clock correction  $a_{f_0}, a_{f_1}, a_{f_2}$  that can be used to compute the accurate time based on GPS time. The equation is as follow:

$$\delta t^{sv} = a_{f_0} + a_{f_1}(t - t_{oc}) + a_{f_2}(t - t_{oc})^2 + \Delta t_r \quad (2.16)$$

This bias can sometimes cause around 5 meters error [6]. Recently, the new technologies make satellites equipped with new atomic clock. It can be expected that satellite clock bias could be minimized in the future.

Secondly, while penetrating the earth atmosphere, the signal has to go through ionosphere and troposphere. While penetrating, the speed of the signal will slow down because of the sun activity ,geomagnetic activity and other atmosphere activities. Such slow down effect will introduce ionospheric error and tropospheric error. Because the ionospheric and tropospheric errors can be affected by temperature, humidity, seasons and day night alternation, it is very difficult to predict the exact time delay for the ionospheric and tropospheric errors. Fig. 2.4 illustrates the signal propagation through the atmosphere.

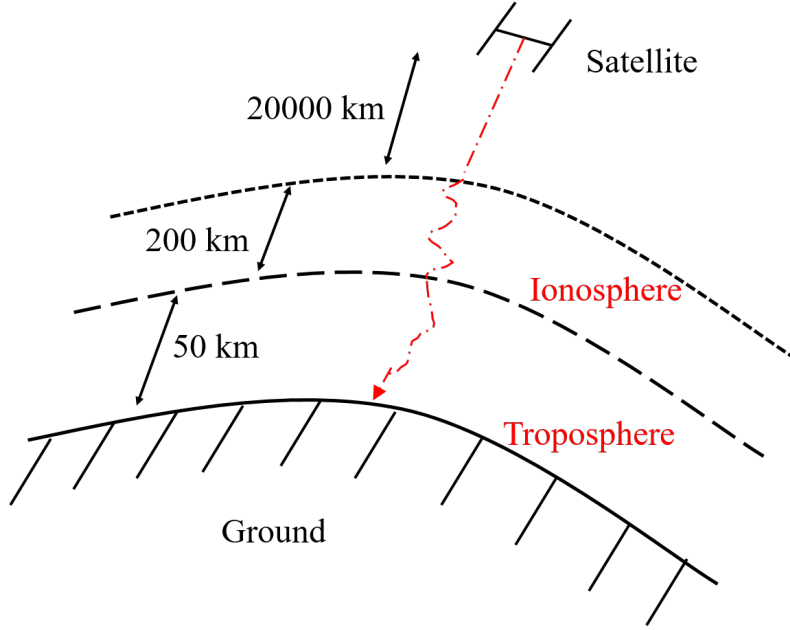


图 2.4. GPS signal must go through ionosphere and troposphere to reach the planet surface.

Fortunately, currently we have Klobuchar model for ionospheric delay and Saastamoinen model for tropospheric delay [6]. Klobuchar model could correct 50% of error caused by ionosphere and Saastamoinen model could correct more than 90% of errors caused by troposphere.

Klobuchar model is developed to solve the ionospheric delay. Such delay is mainly caused by ionization. In Klobuchar model, the ionospheric delay  $I$  can be computed with the following equation:

$$\frac{I}{c} = \begin{cases} A_1 + A_2 \cos\left(\frac{2\pi(t - A_3)}{A_4}\right) & \text{if } |t - A_3| < A_4/4 \\ A_1 & \text{otherwise} \end{cases} \quad (2.17)$$

where  $A_1$  is the delay in the night ( $5 \times 10^{-9}s$ ),  $A_2$  is the amplitude of cosin function used in day time  $A_3$  is the phase of the cosin function and  $A_4$  is the period of the cosin function.

Saastamoinen is developed to solve the tropospheric delay. Tropospheric delay is caused mainly by the dry air water vapor that refracts signal. The equation for the Saastamoinen model is as follows:

$$T = \frac{2.277 \cdot 10^{-3}}{\cos z} \left[ P + \left( \frac{1255}{T} + 0.05 \right) e - B \tan^2 z \right] + \delta R \quad (2.18)$$

Where the  $z$  is the elevation angle of the satellite,  $T$  is temperature,  $P$  is barometric

pressure,  $e$  is water vapor partial pressure,  $B$  is a correction factor to the height, and  $\delta R$  is the correction term.

Thirdly, the receiver's noise can also cause errors. The receiver's noises are considered as antenna error, cable error, shaking error and other type of errors. Finally, in urban canyon environment, the main error source for GPS is error caused by non-line-of-sight (NLOS) and multipath. Those error does not have patterns, and can not be easily corrected. The type of and properties of errors is shown in Tab. 2.1.

表 2.1. The type and properties of GPS pseudorange errors.

Source of Error	Cause	Properties of Error
Satellites	Orbit factors	2 m (bias)
	Clock offset	2 m (bias)
Signal propagation	Ionospheric delay	2 ~around 10 m (bias)
	Tropospheric delay	2.3 ~ 2.6 m (bias)
	Multipath and NLOS	~ 100 m (random)
Receiver based	pseudorange calculation noise	~ 1 m(random)

As shown in Tab. 2.1 the error caused by multipath and NLOS is much more serious than other error factors. Fig. 2.5 shows the NLOS and multipath effect. We can see that if NLOS and multipath happened, the estimated pseudorange will contains large errors due to the reflection signals.

## 2.3 Related Work About Improving GNSS Positioning Accuracy In Urban Environment

There are many researches about correcting and evaluating GNSS positioning results. Walter et al.[8],Blanch et al. [9] have proposed the weighted RAIM method which could find the unhealthy satellite signals and exclude them to improve the positioning result. Later researches [10] [11] have proven the RAIM could also be used to other GNSS systems(Galileo, GLONASS etc.). These researches have shown that RAIM can improve the GNSS positioning result in opensky environments. The detail of RAIM algorithm will be introduced in later section.

The urban canyon scenarios are much more complicated. The assumption of traditional RAIM is not always satisfied [12], because of the serve effects of NLOS and Multipath situations. Meguro et al. [13] use IR image sensor to detect suspicious NLOS or multiple satellite signals. The Japanese GNSS augmentation system QZSS [14] provided an additional satellite with high elevation angle. Using such satellite could correct the po-

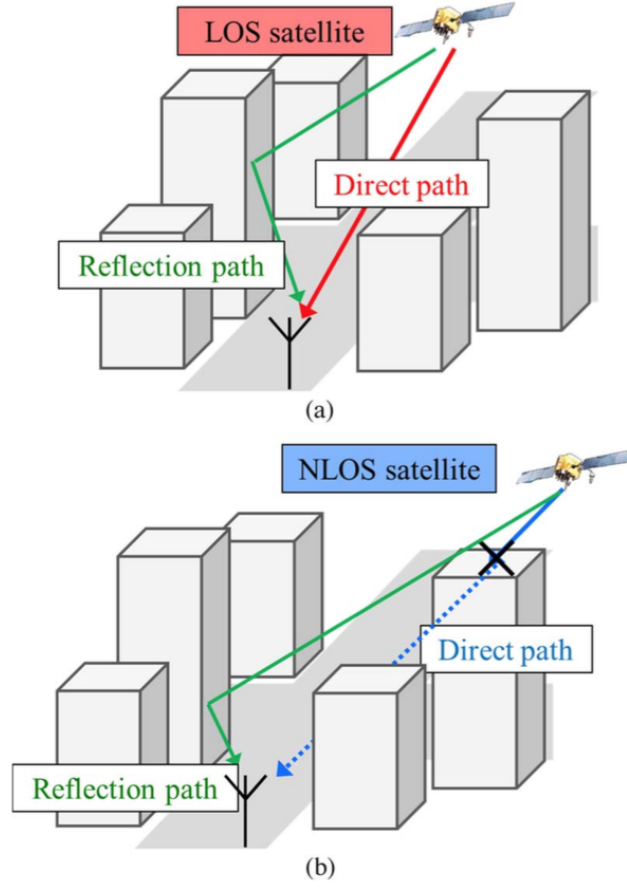


图 2.5. The multipath and NLOS effects in an urban canyon. (a) Multipath effect. (b) NLOS propagation [2].

sitioning result in a certain degree because high elevation angle satellite is hard to suffer from NLOS or Multipath effects. Su et al. [15] conducted a study on RAIM availability in urban canyon area using GPS/BeiDou/QZSS. Tokura et al. [16] introduced  $C/N_0$  and RAIM based satellite selection method combined with D-GNSS. Jiang et al. [17] implemented top down satellite selection method. Groves and Jiang [18] [19] used RANSAC algorithm to performing trustful satellite selection from bottom up. There are also some researches [7] [20] about evaluating finding the trustful GPS(GNSS) positioning result in urban canyon environment using speed and other receiver provided information.

Besides using only GPS data, ideas that combine other data with GPS data to improve the positioning accuracy have also been proposed. For instance, Miura [2] proposed ray tracing algorithm using 3D map of urban area, as illustrated by Fig. 2.6, to detect whether a signal is blocked by building. As can be observed, the 3D building geometry of the urban area is created based on reliable 2D and building height data. After creating the 3D map, ray tracing can be performed so that LOS, NLOS and multipath signals can be detected.

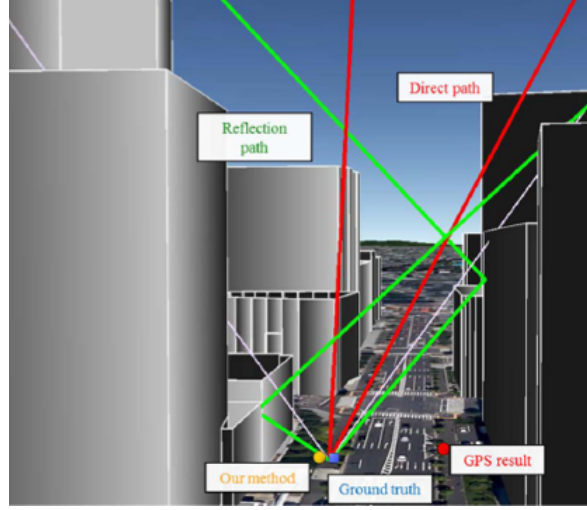


Fig. 2.6. The illustration of 3D-GNSS method [2].

Recent researches also indicated that the altitude (height) information can also be helpful in correcting GNSS positioning result. Because the error in vertical direction is relatively easy to be estimated than it is for the horizontal direction. Kubo et al. [21] used altitude aiding to resolve the ambiguities when doing instantaneous RTK Positioning calculations. Groves et al. [19] used height obtained from terrain height database (usually 3D city model) to calculate an additional ranging measurement from the centre of earth (As illustrated in Fig. 2.7). Because the pseudorange of the pseudo-satellite has less chance to contain error. When calculating the positioning result, we have one more accurate measurement. Therefore, the positioning accuracy could be improved. Iwase et al. [12] used altitude information to identify erroneous satellite signals and mitigate them. They use altitude information to assign weight to each satellite signals. First, they select 3 healthy satellites, then they add one satellite in to calculate the WLS solution, the calculated height minus the height in the altitude map becomes the weight for latterly added satellite signals. The weight assigning equation is as follows:

$$w_i = \begin{cases} \frac{1}{|e_i|} & |e_i| \geq 1 \\ 1 - \log(|e_i|) & |e_i| < 1 \end{cases} \quad (2.19)$$

where the  $|e_i|$  is the calculated vertical error when add satellite  $i$  in.

Current researches, which used altitude information, focused on using such information to exclude or mitigate erroneous satellites [12] or use altitude as additional measurement [19]. In this paper, the altitude information is used to directly correct the erroneous satellite signals and evaluate the GNSS positioning result.

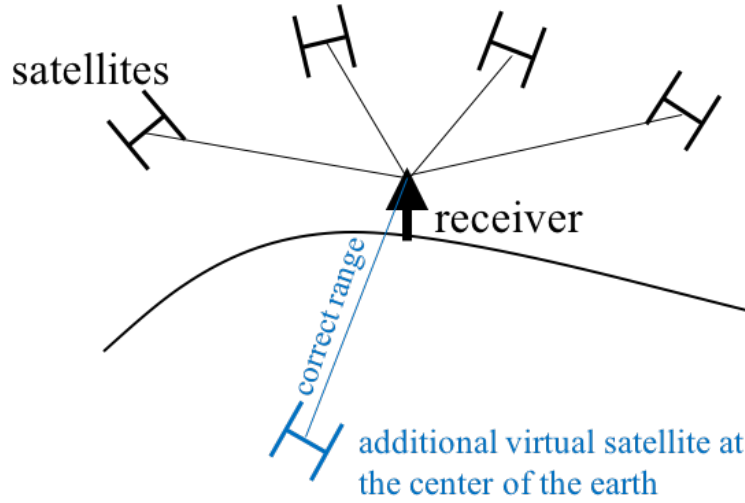


图 2.7. The pseudo satellite placed in the center of the earth. Its pseudorange is the altitude from the 3D city model.

## 2.4 PDR Algorithm and Related Works

In this section, we will introduce the fundamental of pedestrian dead reckoning and also related works with PDR technologies.

With the development of low cost MEMS (micro-electro-mechanical-system), the Inertial Measurement Unit (IMU), the Inertial Navigation System (INS) has been well developed in recent years. The INS is considered as good supporting navigation system using dead reckoning concept. In navigation, dead reckoning is the process of calculating one's current position by using a previously determined position. The most common way to use IMU is to perform double integration of the acceleration to calculate the walking distance, and trust the gyroscope and magnetometer to calculate the moving direction. It is commonly used in airplane and ship navigation. Such concept could also be used for pedestrian navigation. However, the INS could be suffered the problem of error accumulation. If the high end level MEMS could be used, it can provide highly accurate measurement values. As a result, it will not result in large biased positioning errors. However, it is difficult for pedestrian to carry high end MEMS with them. For low cost commercial level MEMS which is embedded in almost all type of smartphone, simple integration method will result in huge error. Because at each step, the measurement values contained some small errors and those errors will accumulate.

In order to avoid such kind of problem, for pedestrian dead reckoning system, researchers preferred to formulate a walking model for pedestrian, instead of using simple integration method. Because the pedestrian has walking cycle, if every step is detected and the

moving direction and step length is calculated, we can get the trajectory of pedestrian step by steps. The error accumulation in this model will not as serious as using simple integration method.

Fig. 2.8 shows the overview of PDR system. For this figure of PDR, only accelerator and magnetometer is used to calculate PDR solution. However, there are also other type of system use gyroscope collected data to calculated user moving direction.

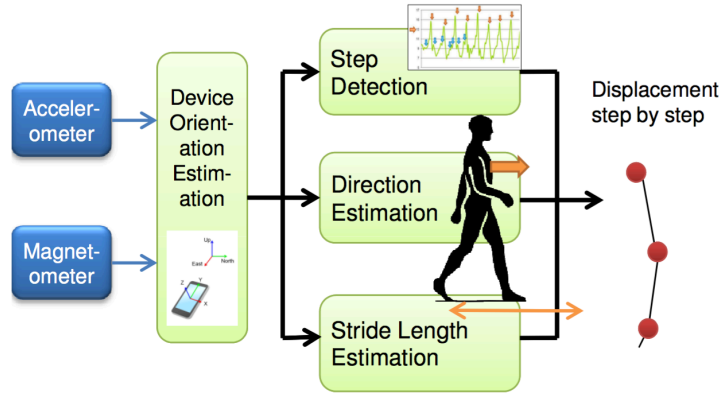


Fig. 2.8. The overview of step based PDR system [3].

The equation to calculate the user's trajectory are as follows:

$$\mathbf{x}_n = \mathbf{x}_{n-1} + \begin{pmatrix} l \cos \theta \\ l \sin \theta \end{pmatrix} \quad (2.20)$$

As shown in the equation, the PDR system could calculate step length  $l$  and moving direction  $\theta$  step by step. We can derive the Nth step 's positioning by accumulating the step from 1 to n-1.

In other words, The core parts of PDR algorithm could be summarized as follows:

- Step Detection
- Stride Length
- Heading orientation estimation

For PDR system, the MEMS could be place at every space in pedestrian's body. The Fig. 2.10 shows the placement of MEMS. For example, it could be attached to pedestrian's foot [22, 4], be attached to pedestrian's waist [23] be placed in pedestrian's pocket [5], be hold by the pedestrian [24] or be mounted on pedestrian's head [25]. This feature brings difficult for PDR system, because the coordinate system for MEMS's local coordinate is different from the Global coordinate System. Therefore, before talk about detailed PDR algorithm. We need first transform measured data in MEMS's local x-y-z coordinate to Global E-N-U coordinate. The relation of X-Y-Z coordinate and E-N-U coordinate can



be viewed in Fig. 2.9 This is because most PDR algorithm are needed to be calculated in global coordinate system. As a result, before calculating the PDR solutions, we need first perform coordinate transformation.

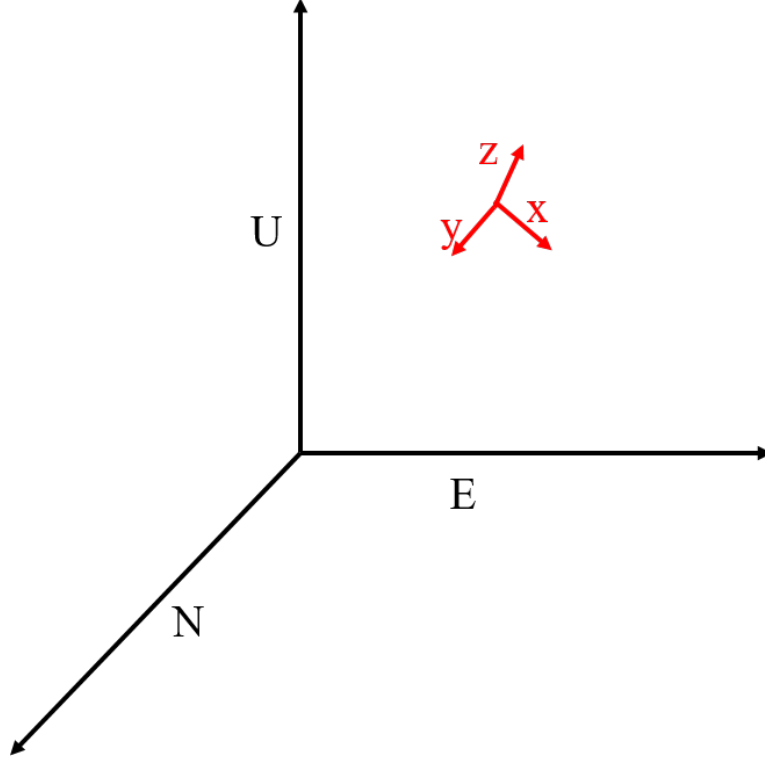


图 2.9. The x-y-z coordinate and E-N-U coordinate.

For coordinate transformation, if gyroscope can be used, it can be derived through accumulating the roll, yaw, pitch angle. However, such method need the initial positioning and placement information of MEMS. Another method to calculate transformation matrix to convert the data from local coordinate system to global coordinate system is using acceleration and magnitude data. The equation to calculate the rotation matrix  $\mathbf{R}$  is as follows:

$$\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \mathbf{R} \times \text{gravity}$$

$$\begin{bmatrix} 0 \\ m_N \\ m_U \end{bmatrix} = \mathbf{R} \times \text{geomagnetic} \quad (2.21)$$

As shown in the equation  $\mathbf{R}$  is the rotation matrix. The *gravity* and *geomagnetic* are the gravity measurement and geomagnetic measurement component in local coordinate system. By solving Eqn. 2.21, we can get the rotation matrix.



Fig. 2.10. The placement of MEMS.

For step detection, most of researchers used acceleration data in global vertical axis. Fig. 2.11 shows the vertical acceleration data of a walking pedestrian.

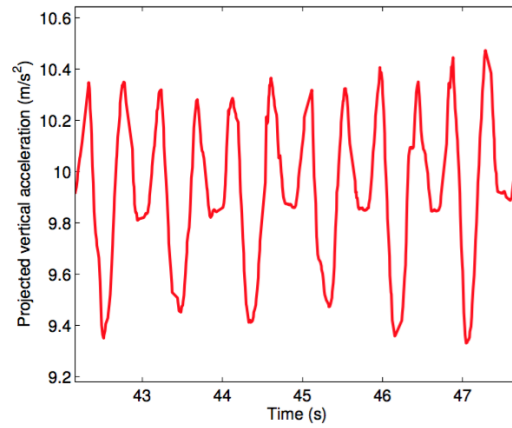


Fig. 2.11. The vertical acceleration data of a walking pedestrian [4].

As shown in Fig. 2.11, the changing of vertical acceleration data has its circulation period. Many researchers try to find the feature to detect the start or stop of each circulation. Jimenez et al. [22] use magnitude and variance of vertical acceleration to detect the swing phase and stance phase for each step. A step is detected when swing phase ends and stance phase starts. Steinhoff et al. [5] also use variance of vertical acceleration. However, they only detect the peak of variance as each step. As a result,

when utilizing their method, the acceleration data firstly needed to be pass through a low-pass-filter. Jin et al. [26] proposed to detect the peak or bottom directly on acceleration. All of those methods have its pros and cons and suitable for different placement of sensors.

Regrading to the step length estimation, if the sensor is mounted on pedestrian's foot, Jimenez et al. [22] use zero velocity updated (ZUPT) algorithm to calculated the stride length. The ZUPT algorithm is considered as the most reliable and versatile method regardless of the user and displacement patterns(walk, run, climb etc.) At foot stance the velocity is known to be zero, so the basic idea of the algorithm is to correct the linear velocities obtained after integrating the accelerometer at each step. The correction equation is as follows:

$$\check{v}_i = v_i - \frac{[u_k(i - i_{(k-1)}) + u_{k-1}(i_{(k)} - i)]}{m_k} \quad (2.22)$$

Note that the  $u_k$  represents the velocity error accumulated at step k (it should be zero) and  $i_{(k)}$  represents the sample index of k stance occurrence. The  $m_k$  is the number of samples at step k. By using ZUPT algorithm, the error of estimated stride could be corrected at each step.

Weinberg [27] instead, proposed a nonlinear method that uses the maximum and minimum peaks obtained during the step detection phase, and a constant that can be fixed or adjusted for different users or type of activity, such as walking or running. This method is given by the following equations:

$$Lstep \approx \sqrt[4]{A_{max} - A_{min}} \times K \quad (2.23)$$

where  $Lstep$  stands for the stride length,  $A_{max}$  and  $A_{min}$  are the maximum and minimum vertical acceleration values within the step, and  $K$  the adjustable constant mentioned before. The Weinberg equation is only an approximation of the stride length. However, it is relatively resistant to noise.

Shin et al. [28] proposed an adaptive method known as the linear method as follows:

$$Lstep = a \times f + b \times v + c \quad (2.24)$$

In this method, the distance is estimated by calculating the walking frequency (f) and acceleration variance (v) and scaling the results using the constants a, b, c that can be calculated during a training stage using linear regression.

The most difficult part for PDR system is the heading orientation estimation. If the device is attached to a fix position of human body, we can directly integrate the acceleration to derive the moving direction [22] [4] [25]. There are also many researches using

magnetiude and gyroscope data to get the moving direction [29] [30]. Steinhoff et al. use different approaches [5]. They first project the acceleration data into E-N plane, and perform principle component analysis (PCA) on acceleration data in every two seconds. The first principle component is the moving direction of the user. This method has the advantage that it does not need the sensor to be relatively fixed on the pedestrian's body. It allows the sensor to be loosely placed in pedestrian's pocket. The PCA method could be applied to 3 dimension acceleration or 2 dimension acceleration (map acceleration to E-N plane), Fig. 2.12 shows the extraction of moving direction using both PCA2D and PCA3D method. Based on Steinhoff et al.'s research [5], the PCA2D method have best performance.

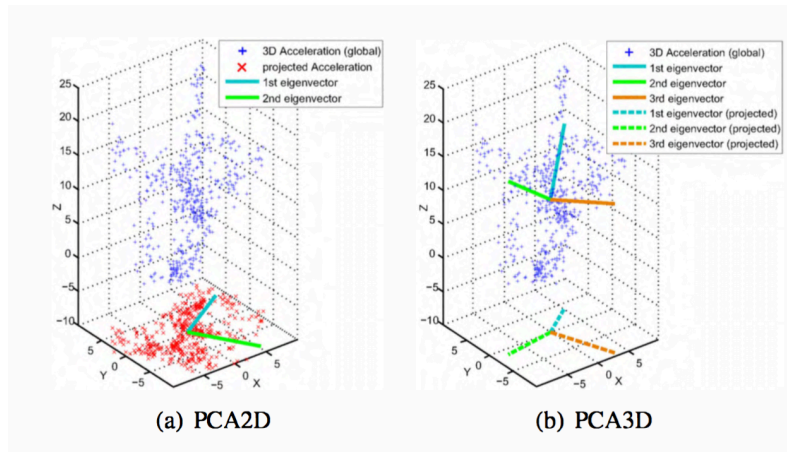


Fig. 2.12. Extraction of motion direction with PCA [5].

## 2.5 Problem and Error Source of PDR System

Although using the IMU sensor to implement the PDR algorithm could get a relatively correct pedestrian's trajectory. The PDR systems has its problems. One problem is that it can not provide absolute positioning solutions. It needed manually provided initial position or need the help from other type of positioning system such as GPS to provide the initial positioning. If the PDR is used in indoor environment, manually providing the initial positioning would not be a big problem. However, if we need to develop the navigation system for outdoor environment, manually initialized the PDR system would be infeasible and we usually need the help from other positioning system to do initialization for PDR.

Another problem for PDR system is more seriously and make the simple PDR system could not work standalone. The problem is error accumulation. When PDR accumulate each step, it also accumulate the error of each step. The PDR system could not correct or

even estimate the error accumulation. As shown in Fig. 2.13 if the stride length estimation or heading orientation estimation algorithm has mis-calculated in a certain epoch. The error will also influence the following epoch. Finally, the PDR trajectory will contain large bias.

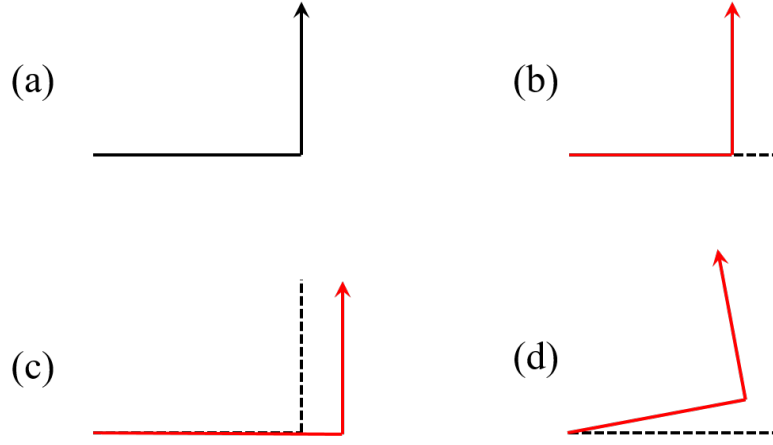


图 2.13. The error accumulation of PDR system.

As a result, it is hard to use PDR system alone to provide the positioning service. Usually, the PDR is integrated with other type of system. In indoor environment, integrating PDR with map matching method [31] or WPS [32]. In outdoor environment, the GPS is the first choice for integration [33].

## 2.6 Fundamental of Integrated Navigation System and Related Works

As discussed before, the WPS, GPS and PDR system both have its problems. Even though there are a lot of researchers try to mitigate the error for those standalone system, integrating those system could be an more effective way to solve those problem. The most popular integration combination is PDR + WPS or PDR + GPS. And kalman filter framework is the most popular integrating framework.

The kalman filter [34] is a mathematical optimization algorithm that makes an estimation of an observed variable using a predation and update(or measurement) equations. This method performs much better than using the update value using signal measurement alone. Traditional kalman filter can only be applied to liner system.

The measurement and state model of kalman filter can be described as follows:

$$x_{t+1} = A_t x_t + G_t w_t \quad (2.25)$$

$$y_t = H_t x_t + v_t \quad (2.26)$$

Where  $x_t$  is the state vector,  $A_t$  is the state transition matrix,  $G_t$  is the error effect matrix,  $y_t$  is the observation vector and  $H_t$  is the measurement Matrix.  $w_t \sim N(0, Q)$ ,  $v_t \sim N(0, R)$ , represents the Gaussian control noise and the measurement noise.

The predation equations are:

$$\hat{x}_{t|t-1} = A_{t-1}\hat{x}_{t-1} \quad (2.27)$$

$$P_{t|t-1} = A_{t-1}P_{t-1}A_{t-1}^T + G_{t-1}Q_tG_{t-1}^T \quad (2.28)$$

Where the  $\hat{x}$  is the approximate state and P is the measured covariance of the variable of interest.

The update equations are:

$$K_t = P_{t|t-1}H_t^T (H_tP_{t|t-1}H_t^T + R_t)^{-1} \quad (2.29)$$

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t (y_t - H_t\hat{x}_{t|t-1}) \quad (2.30)$$

$$P_t = (I - K_tH_t)P_{t|t-1} \quad (2.31)$$

Where the  $K$  is the kalman gain. The prediction equation find the *priori* distribution by *predicting* the mean and variance of the next value of the observed variable. The update equations calculate the *posterior* distribution by using the new weighted measurements.

The basic idea to use kalman filter is to update the state iteratively. The pedestrian's walking pattern could be approximate by an liner model, therefore, PDR system is suitable for being the state model in kalman filter. The WPS and GPS usually are used to provide observation. The PDR + WPS is usually be used in indoor environment [32]. Moreover, in indoor environment, the PDR could be integrated with map matching algorithm [4] [31]. With the help of map matching, the PDR trajectory could be corrected at some corner point, or with the help of knowing the wall position.

Nevertheless, in outdoor environment, the integration work is much harder. On one hand, in outdoor environment, the positioning accuracy for WPS is not accurate enough for integration purpose, usually varied from 10 to 40 meters [35]. Moreover, it is also hard to evaluate the positioning result of the WPS system. As a result, the GPS + PDR integration framework is much popular in outdoor scenario. There are many research about integrating GPS and PDR system could be found in [33] [3]. Usually, the PDR + GPS using kalman filter or particle filter as integration framework. In this thesis, we mainly discuss kalman filter based integration framework. Fig. 2.14 shows the basic idea of integrating PDR and GPS under kalman filter framework.

As shown in Fig. 2.14, the PDR is used as state model and GPS is used as observation. As a result, the critical part is to evaluate the positioning accuracy of GPS. The GPS accuracy value reveals the error in GPS solution of a certian epoch. It contains geometry

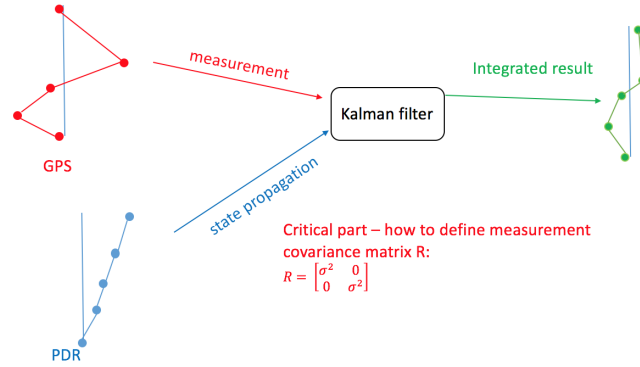


图 2.14. The idea of using kalman filter to fusion PDR and GPS

factor and pseudorange error factor[6]. The GPS accuracy is defined as follows:

$$(\text{error in GPS solution}) = (\text{geometry factor}) \times (\text{pseudorange error factor}) \quad (2.32)$$

Geometry factor represents the error brought by the satellite geometry. In the presence of measurement errors, the range rings used to compute the user's location will be in error and result in error in the computed position. The concept of dilution of precision is used to estimate the position error that results from measurement errors depends on the user/satellite relative geometry. These ideas are illustrated in Fig. 2.15. Two geometries are indicated.

In Fig. 2.15(a) the satellites are located approximately at right angles with respect to the user location. In Fig. 2.15(b) the angle between the satellites as viewed from the user is much smaller. As can be seen, if the DOP value is large, it means the satellite geometry is not good and the uncertainty range is larger than when DOP value is small.

The pseudorange error factor is the pseudorange error residues described in previous section. If the value is large, it means the positioning result gets more chance to have errors.

Usually, the GPS accuracy can be calculated as follows [6]:

$$\sigma = HDOP \times \sigma_{URE} \quad (2.33)$$

$$\sigma_{URE} = \sqrt{\sigma_{eph}^2 + \sigma_{clk}^2 + \sigma_{ion}^2 + \sigma_{trop}^2 + \sigma_{Multipath}^2} \quad (2.34)$$

$\sigma_{eph}$ : Ephemeris Error,  $\sigma_{clk}$ : Satellite Clock Error,  $\sigma_{ion}$ : Ionospheric Delay,  $\sigma_{trop}$ : Tropospheric Delay,  $\sigma_{Multipath}$ : Multipath Error.  $HDOP$ : horizontal dilution of position.

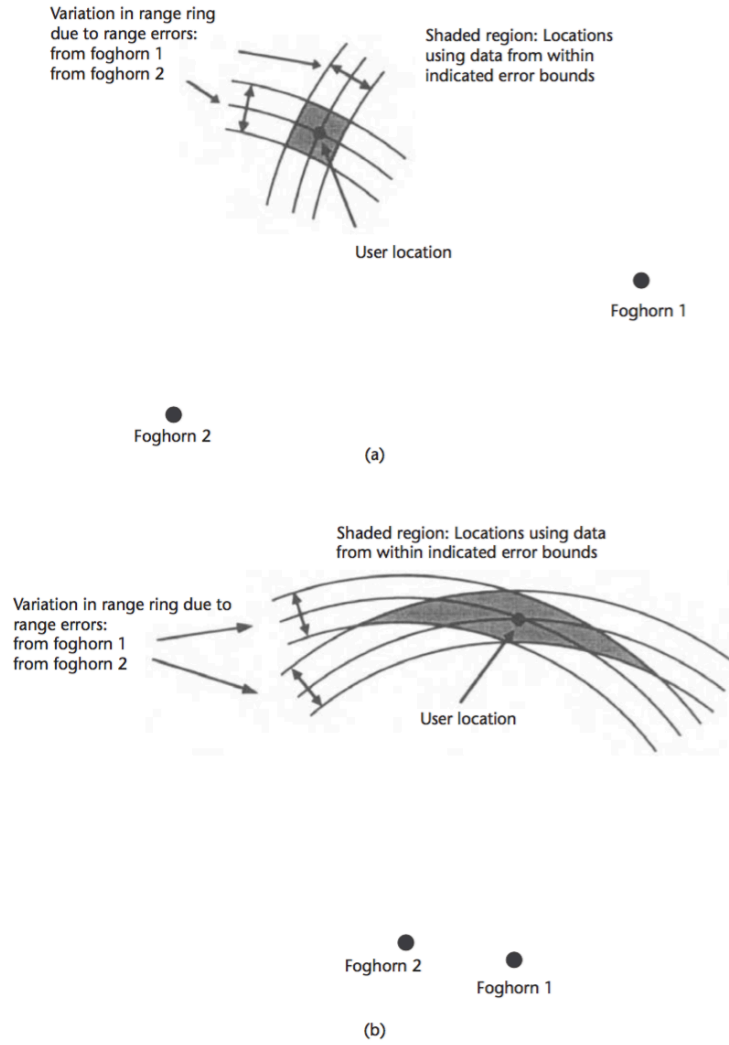


Fig 2.15. Relative geometry and dilution of precision: (a) geometry with low DOP, and (b) geometry with high DOP [6].

The DOP value is calculated using  $(G^T G)^{-1}$  components in weighted least square calculation. The  $(G^T G)^{-1}$  can be expanded as follows:

$$(G^T G)^{-1} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} \\ D_{21} & D_{22} & D_{23} & D_{24} \\ D_{31} & D_{32} & D_{33} & D_{34} \\ D_{41} & D_{42} & D_{43} & D_{44} \end{bmatrix} \quad (2.35)$$

The most general parameter to denote the satellite geometry is termed the geometric dilution of precision (GDOP) and is defined by the formula:

$$GDOP = \sqrt{D_{11} + D_{22} + D_{33} + D_{44}} \quad (2.36)$$



Several other DOP parameters in common use are useful to characterize the accuracy of various components of the position/time solution. These are termed position dilution of precision (PDOP), horizontal dilution of precision (HDOP), vertical dilution of precision (VDOP), and time dilution of precision (TDOP). Those values can be calculated as follows:

$$PDOP = \sqrt{D_{11} + D_{22} + D_{33}} \quad (2.37)$$

$$HDOP = \sqrt{D_{11} + D_{22}} \quad (2.38)$$

$$VDOP = \sqrt{D_{33}} \quad (2.39)$$

$$TDOP = \frac{\sqrt{D_{44}}}{c} \quad (2.40)$$

Even though the idea of GPS accuracy is clear, this calculation is not sufficient enough to represent the reliability of GPS positioning result in urban environment. The main reason is most of errors residue(except multipath error) are defined as a constant based on the statistical result, and there is no clear way to measure the pseudorange errors brought by NLOS and Multipath effect. As a result, how to accurate estimate the accuracy of GPS positioning results in urban environment is still a big challenge in current world.

Drawil el. al [7] recently published a novel method for GPS accuracy estimation for cars in urban environment. Based on their research, they classified the GPS result into accurate (less than 10 meters), marginally-accurate(10-20 meters) and inaccurate(larger than 20 meters) category. They use PCA method to perform classification. As shown in Fig.2.16 the GPS result could relatively be classified into those three categories.

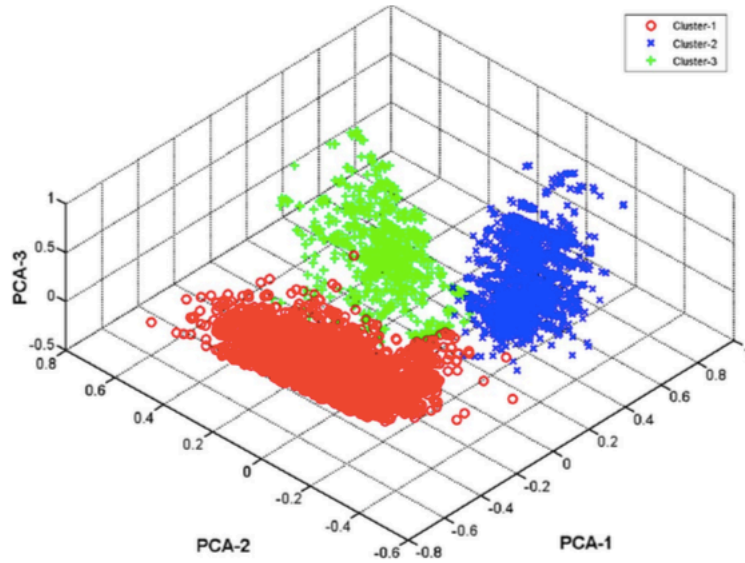


图 2.16. The PCA cluster for GPS result in different accuracy category [7].

The feature they used in the classification is the number of satellite received, DOP

表 2.2. Feature and localization error relationship [7].

Accuracy Band	Feature	Range
Accurate	# of Sat	$> 10$
	DOP	$\leq 1$
	$\mu_{SNR}$	$> 35$ db
	$\sigma_{SNR}$	$< 12$ db
	Speed	$> 20$ Km/h
Marginally-Accurate	# of Sat	$\geq 4$
	DOP	$\leq 6$
	$\mu_{SNR}$	$< 35$ db
	$\sigma_{SNR}$	$> 10$ db
	Speed	$< 40$ Km/h
Inaccurate	# of Sat	$= 4$
	DOP	$> 6$
	$\mu_{SNR}$	$< 35$ db
	$\sigma_{SNR}$	$> 10$ db
	Speed	$< 30$ Km/h

value, mean and variance of the received satellites' SNR, and the speed of the car. The result classification threshold is shown in Tab. 2.2.

Even though their method could classify the accuracy of GPS positioning result in a good accuracy, their method was designed for car application and needed the pre-trained classifier. For pedestrian integration purpose, some new way to calculate the poisoning accuracy is still needed to be developed.

## 第 3 章

# The Altitude Map Based Positioning Algorithm

In this chapter, we will introduce the altitude map based GNSS method. We also call our method height aided GNSS or WLS + height aiding. In Section 3.1 we will introduce the proposed method and in Section 3.2 the experiment setup and experiment result are discussed.

### 3.1 The Proposed Altitude Map Based Pseudorange Correction Method

As mentioned in previous chapter, previously there are many researchers use altitude map to improve the positioning accuracy in urban environment. For example, Grove et al. [19] add one psudo-satellite at the center of the earth with the help of the 3D city map. Iwase et al. [12] use altitude map based weight to mitigate the erroneous satellite signals. In this paper, we will try to use altitude map to direct correct erroneous pseudorange of the suspicious satellites, therefore, improving the positioning accuracy of GNSS positioning.

The idea of the proposed method is to use the vertical error as a reference to correct the NLOS or multipath contaminated satellite signals. If the GNSS signal suffered from those effects, the result of altitude solution would change drastically because of the large pseudorange error. This large pseudorange error hence induces a large positioning error in both horizontal and vertical directions. Ideally, in opensky environments, the horizontal and vertical errors (HE and VE) are related to each other through pseudorange because of the satellite geometry. As show in Fig. 3.1 the VE and HE should both increased when one of the satellite signals contained erroneous pseudorange.

When multiple signals contain errors, or even if the relationship is hard to be revealed through geometry, such relationship should still be preserved. In urban canyon environments, when contaminated signals are taken into consideration, it is difficult to describe

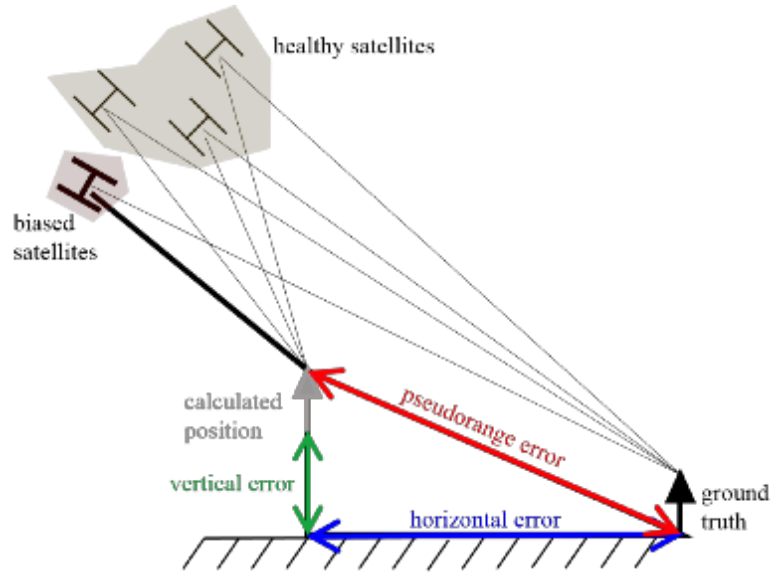


Fig. 3.1. Relationship between horizontal and vertical errors of GNSS positioning.

the relationship between horizontal errors, pseudorange errors and vertical errors in geometry structure. However, the multipath effect finally also results in large pseudorange errors.

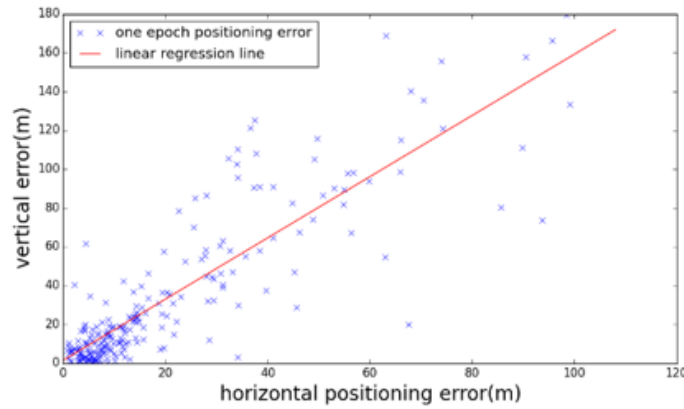


Fig. 3.2. Relationship between horizontal vertical errors of GNSS positioning in urban canyon.

Fig. 3.2 shows the HE and VE relation in urban canyon environment. The blue 'x' represent a HE and VE of a GPS positioning result. The red line is the liner regression line of the HE and VE in this experiment. It can be seen that the HE and VE are also close related to each other. As a result, HE can be reduced if we can correct VE.

The VE can be calculated based on an altitude map as shown in Eqn. 3.1

$$VE = height_{WLS} - height_{map}(lat, lon) \quad (3.1)$$

where  $height_{WLS}$  and  $height_{map}$  are height position calculated by weighted least square (WLS) and altitude map, respectively. The WLS calculation is discussed in previous chapter. The height calculated by WLS is ellipsoid height. To obtain an ellipsoid height, the orthometric and geoid heights are required as shown in Fig. 3.3. This paper obtains the geoid height from geoid map and get the orthometric height from the Google API. We make an assumption that the pedestrian or vehicle are moving on the ground, so the altitude of the object should not be varied drastically in short distance. If the assumption is correct, the correct altitude can be calculated from only a rough horizontal positioning result using altitude map. The altitude map takes the rough latitude and longitude (calculated by WLS in this paper) as inputs and gives a reference vertical position as an output.

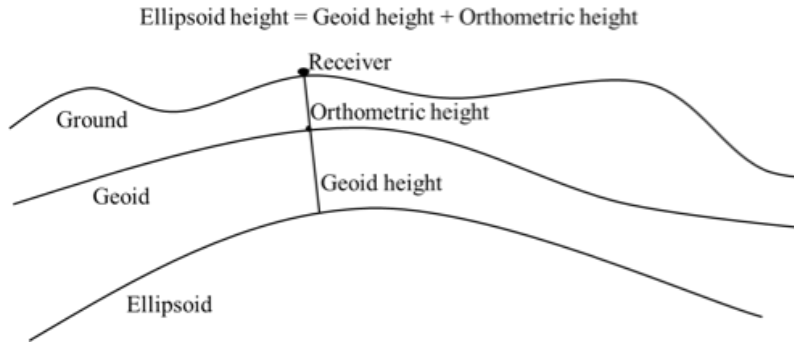


图 3.3. Relationship between geoid height, orthometric height, and ellipsoid height.

Note that the VE is related to pseudorange error as below:

$$VE = VDOP \times \rho^{err} \quad (3.2)$$

where, VDOP denotes the vertical dilution of precision (VDOP), it has been discussed in previous chapter.  $\rho^{err}$  denotes the pseudorange error factor, which is pseudorange error residual and can be calculated as follows:

$$\rho^{err} = \rho - r - b_{rcv} \quad (3.3)$$

where  $\rho$  is the measured pseudorange,  $r$  is the geometry distance between satellite and receiver, and  $b_{rcv}$  is the user side clock bias.

Ideally, VE can be reduced to zero if we can eliminate pseudorange error. Therefore, an algorithm has been proposed to use the VE as an objective function to add corrections

to pseudorange of erroneous satellites. As a result, the proposed method is finding the suspicious error satellite and correcting its pseudorange with the help of altitude map.

Before correcting the pseudorange, it is important to decide which satellite is contaminated. We use The receiver autonomous integrity monitoring (RAIM) like consistency check algorithm to classified the satellite signals into healthy subset and suspicious subset. The RAIM was first be introduced in [8], and was designed to detect consistent epoch and inconsistent epoch. Moreover, according to [6], The RAIM could also be utilized to detect the fault satellite. As shown in Fig. 3.4, if satellite 1 contains error, when excluding the erroneous satellite, the positioning solution will near the ground truth.

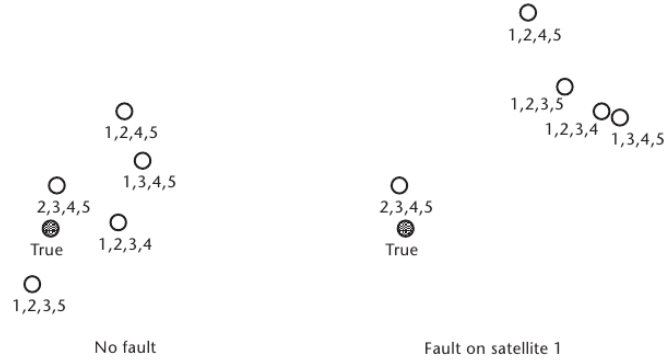


Fig 3.4. The fault satellite detection [6].

The formula to perform consistency check is as follows:

$$y = G.x + \varepsilon \quad (3.4)$$

$$\hat{x} = (G^T.W.G)^{-1}.G^T.W.y = K.y \quad (3.5)$$

$$\hat{\varepsilon} = y - G.\hat{x} = (I - G.K).y = (I - P).y \quad (3.6)$$

$$WSSE = \hat{\varepsilon}^T.W.\hat{\varepsilon} = y^T.W.(I - P).y \quad (3.7)$$

Where  $x$  is the four dimensional position vector (north, east, up and clock) about which the linearization has been made,  $y$  is an  $N$  dimensional vector containing the raw pseudorange measurements minus the expected ranging values based on the location of the satellites and the location of the user. The  $G$  is the observation matrix,  $W$  is the Weighting matrix  $\sqrt{WSSE}$  could be use as test statistic in order to judge the goodness of the least squares fit, in other world, the quality of GPS position solution. The  $\hat{\varepsilon}$  is the pseudorange error residue and it is follows chi-square distribution. We can use chi-square test to exam the quality solution.

The flow-chart for satellite signals classification is show in Fig. 3.5.

As shown in Fig. 3.5, we check the consistency of satellite signals iteratively. At each

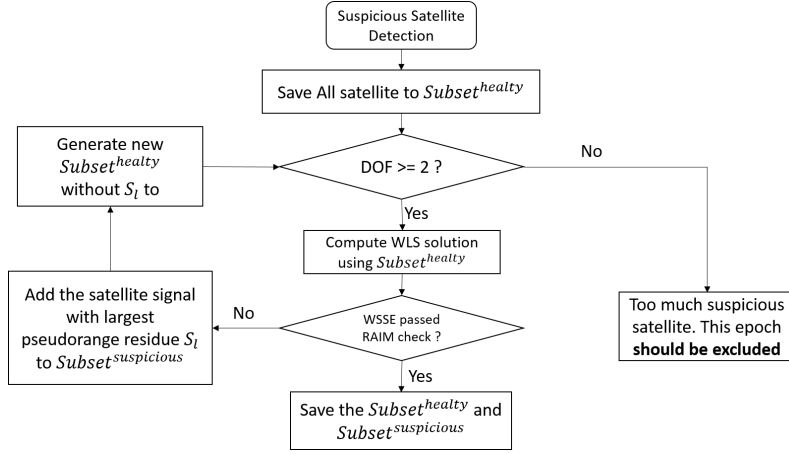


图 3.5. The flowchart of suspicious satellite signal detections.

iteration, we add the satellite with largest pseudorange error residue into suspicious satellite set. After that, we check the consistency of reset satellite signals until we can divide the satellite signals into healthy satellite subset and suspicious satellite subset. If too much suspicious satellite signals was received in this epoch, and we cannot find enough healthy satellite, we marked this epoch as erroneous epoch and do not provide positioning solution in this epoch. The DOF stands for degree of freedom. The formula to calculate the degree of freedom is as follows:

$$DOF = N_{SV} - N_{stat} \quad (3.8)$$

The  $N_{SV}$  stands for number of received satellite signals in this epoch, the  $N_{stat}$  stands for number of unknowns in weighted least square formula. If we use both GPS, GLONASS and QZSS, we have five unknowns in weighted least square equations.

After we successfully divide the satellite signals into healthy subset and suspicious subset, we correct the satellite signals in the suspicious subset one by one. The procedure of correction is shown in Fig. 3.6.

So the important part is how to correct the satellite signals, we use the idea of global optimization, which means we manually add correction to the pseudorange of the satellite signals. The algorithm to correct the satellite signals is shown as follows:

As shown in Algorithm 1, we correct one target suspicious satellite signal  $s_e$  from suspicious satellite subset  $subset^{suspicious}$  at each iteration. Before correction procedure, we initialized corrected signal set  $set^{corrected}$  as healthy signal subset  $subset^{healthy}$ . At each iteration, we first add target suspicious satellite signal  $s_e$  to  $set^{corrected}$ . Then, we add pseudorange corrections  $\delta\rho$  from  $-TP$  to  $TP$  at one meter step to pseudorange of  $s_e$ . The  $TP$  is the possible pseudorange errors happened in urban canyon. Base on the experimental result, we set  $TP$  equals to 200. At each time we add correction  $\delta\rho$  to

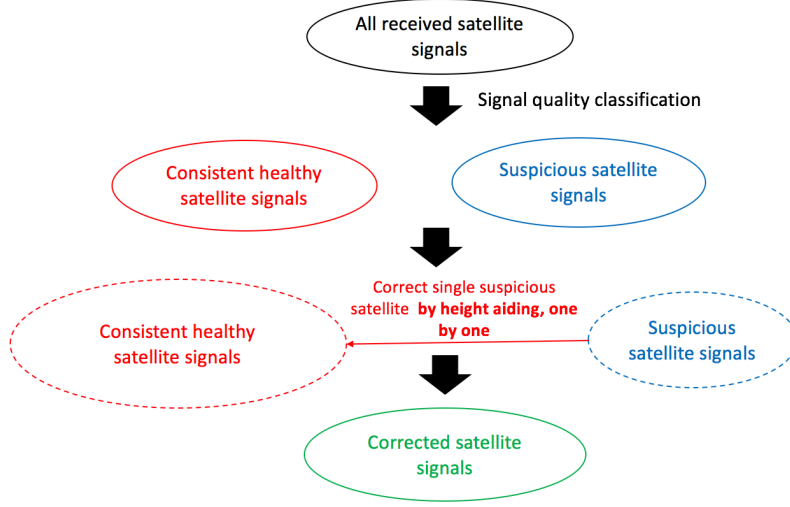


图 3.6. The satellite correction procedure.

pseudorange of satellite  $s_e$ . We save the correction value that make  $VE$  to be smallest as  $\delta\rho_{best}$ . The  $\delta\rho_{best}$  is the final pseudorange correction value for satellite  $s_e$ . After correcting one satellite, we goes to next iteration.

## 3.2 Experiment Result

A low-cost u-blox M8 receiver (As shown in Fig. 3.7, which could receive both GPS and GLONASS signals, is used during experiment. This paper selects an urban canyon in Hitotsubashi areas of Tokyo city as the experiment area. In the experiment area the orthometric height is around 2 to 5 meters and the geoid height is around 37 to 38 meters. The receiver is put on the pedestrian's shoulder and the pedestrian walks on a 200 meters experiment path. The experiment environment and walking trajectory is shown in Fig. 3.8.

In our experiment, we have 15 tests from 3 different people. The experiment result of one test is shown in Fig. 3.9.

As shown in Fig. 3.9. After using altitude map based height aiding, the some GNSS solution is moved to right place. For example, the GNSS solutions in the left bottom part of WLS method figure is moving to trajectory's right start point in WLS + height aiding figure, which the right place. Moreover, some epoch that contains too much suspicious satellite or not enough satellite to perform consistency check is discarded. By viewing the two figure, we can find most outlier is been deleted or moved to the right place.

In order to study the improvement by satellite pseudorange correction, we want to exclude the epoch that is excluded by consistency check (too much erroneous satellites) and consistent epoch. Fig. 3.10 shows those epochs that the suspicious satellite is find and



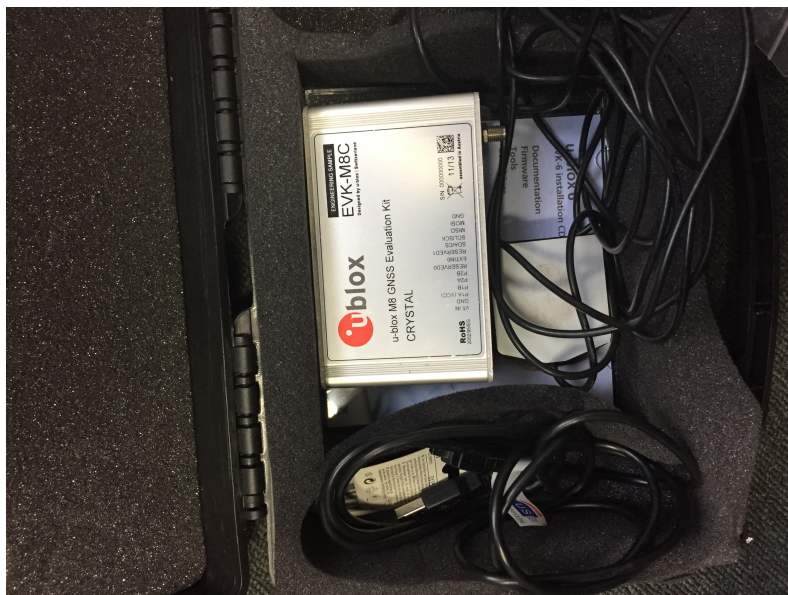


图 3.7. The u-blox M8 receiver used in the experiment.



图 3.8. The experiment environment and walking trajectory.

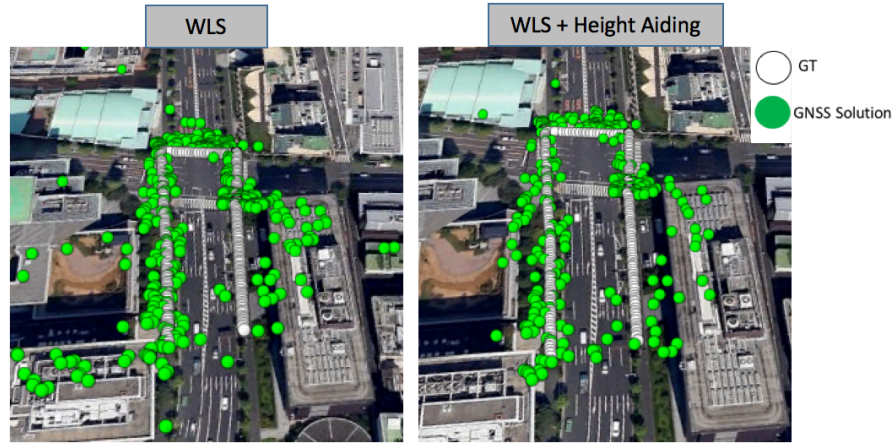


Fig. 3.9. The WLS and WLS + height aiding comparison for one single test.

pseudorange correction algorithm has functioned. We called those epoch related epochs. There are around 23% of epochs is related epochs in this test. By viewing Fig. 3.10 we can see that over 70% of related epoch the positioning error could be improved after correcting the pseudorange of the suspicious satellite signals. The mean error of related epoch before applying height based correction is 12.97 meters. After height based correction, the error has been minimized to 8.14 meters. In other word, around 5 meters improvement has been achieved after applying height aiding. One thing need to be mentioned that is there are around 30% of epochs, their positioning error is degraded after apply height aiding. This is mainly because the complicated environment of urban canyon make the satellite signal classification algorithm not so accurate. If some erroneous satellite signals have been added to consistent set or some healthy satellite signals have been added to suspicious set, our assumption of height aiding algorithm becomes wrong and therefore, the correction algorithm would not work.

We pick up one epoch to do case study. The Fig. 3.11 shows the one epoch position result before height aiding and after height aiding. The yellow dot is the positioning result in epoch 276364 before applying height aiding, the red dot is the positioning result after applying height aiding. We can see the position mean error is minimized from 35.2 meters to 3.89 meters. By viewing the left table, we can see that our consistency check algorithm marked the satellite with PRN number 23 as suspicious satellite, because after adding discarding the satellite with PRN number 23, the rest satellites is consistent and passed RAIM check, we can see that the pseudorange error residual (PSER) of the rest satellite is small. For satellite with PRN number 23, we can see that its elevation angle is 30.91 degree, so the multipath or NLOS effect could be more easily happened. After correcting

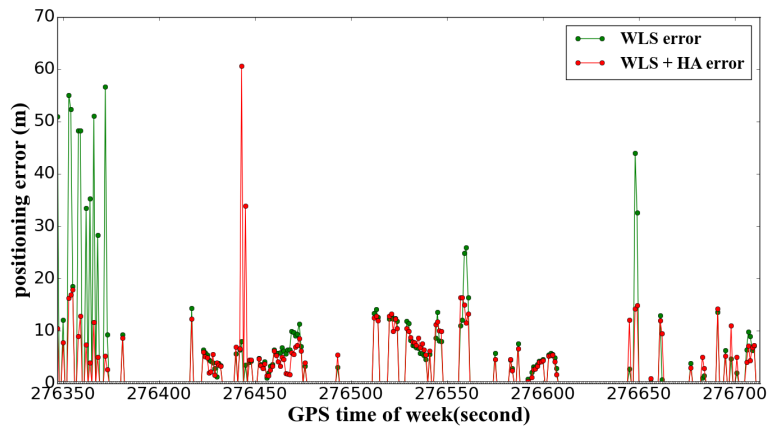


图 3.10. The related epoch comparison between WLS and WLS + HA

the pseudorange of suspicious satellite, the position error residue has been minimized to 0.62 meters. The Fig. 3.12 shows the pseudorange error correction value added to the satellite with PRN number 23. It can be seen that with the pseudorange error correction value range form -200 to 200 meters, it is clear that the tendency of changing vertical error is consistent with horizontal error. Finally, when the pseudorange error correction value is -80 meters, the vertical error is minimized and we can see that the horizontal error is also almost be minimized.

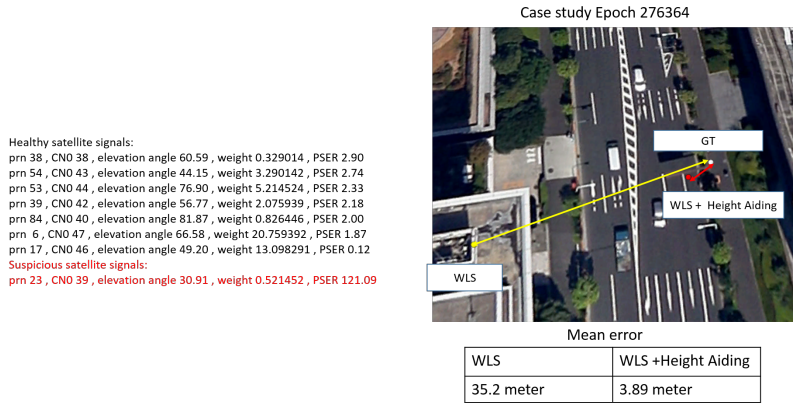


图 3.11. One epoch case study for WLS + height.

The Tab. 3.1 shows the position result of totally 15 test. We first compare position solutions of all epochs between WLS and WLS + Height Aiding (HA) method. However, only this comparison is not sufficient, because some epoch is discarded in WLS + Height Aiding method because this epoch does not pass the consistency check. Moreover, there are also many epochs that is already consistent that we can not find any suspicious satellite in those epoch. In order to do the comparison more clearly, we add one more additional

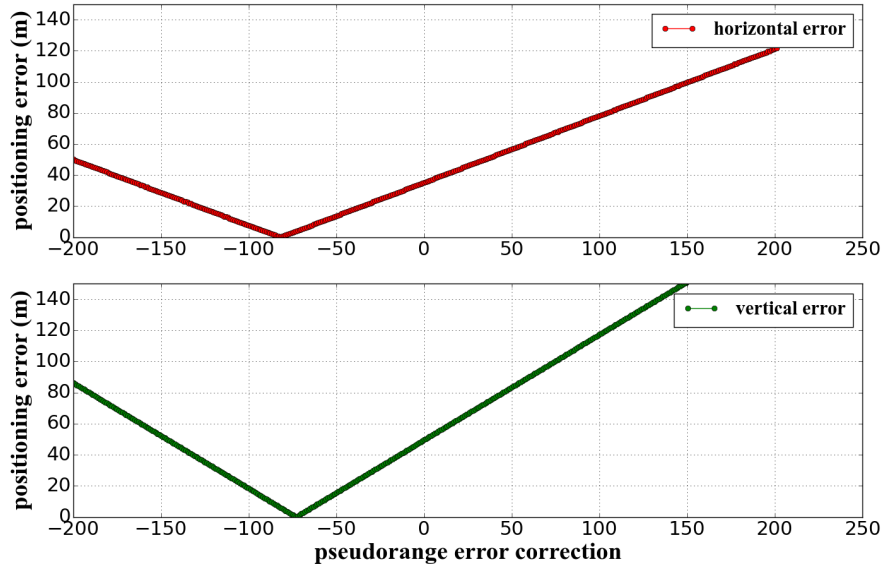


Fig 3.12. The pseudorange error correction value and related horizontal and vertical errors.

comparison, which is comparison between the related epochs. The related epochs are the epoch that the algorithm is successfully find the consistent healthy satellite subset and suspicious satellite subset and perform the pseudorange correction algorithm with the help of altitude map. By observing this table, we can see that in 13 over 15 test, applying height aiding could improve the position result. We can also observed that the related epochs have relatively low availability(avail.), this means when doing positioning in urban environment, the epoch that can be corrected is relatively low. When considering totally 15 tests, it can be find that we can achieve around 5 meters improvement after applying height aiding. By only considering related epochs, we can see that the altitude map based correction could brings around 4 meters positioning mean error improvement.

表 3.1. The mean error and standard deviation comparison of WLS and WLS + height aiding result for total 15 case the unit is meter.

data	WLS			WLS + height aiding			related epochs			
	mean	std	avail.	mean	std	avail.	WLS		WLS + HA	
							mean	std	mean	WLS + HA std
Test1	11.44	15.68	99.73%	6.62	5.47	81.30%	12.97	14.07	8.14	3.84
Test2	19.52	19.77	95.19%	10.48	12.56	52.15%	16.19	13.62	13.52	11.25
Test3	14.09	13.13	99.72%	12.58	10.69	89.33%	16.92	12.89	16.12	13.53
Test4	11.48	12.2	100%	9.49	9.93	97.11%	16.04	14.32	11.68	12.52
Test5	11.85	16.77	99.74%	7.45	8.82	84.54%	15.68	19.89	10.56	12.53
Test6	16.31	17.72	97.91%	8.3	6.62	65.27%	9.54	8.49	7.66	7.04
Test7	17.77	18.34	96.11%	9.51	7.87	70.00%	14.23	11.25	9.8	9.62
Test8	13.8	14.35	100%	15.86	17.78	64.71%	17.49	17.92	17.59	21.17
Test9	20.96	16.29	96.88%	15.71	8.37	67.45%	27.8	17.97	17.45	12
Test10	24.2	24.48	99.20%	16.41	18.54	65.06%	16.36	13.88	14.11	12.88
Test11	22	15.4	95.93%	19.06	14.72	73.28%	22.64	14.6	16.16	13.62
Test12	24.94	18.65	100%	18.78	16.53	82.61%	26.93	14.53	18.52	17.69
Test13	16.9	15.13	99.73%	13.51	12.81	81.59%	17.57	13.13	13.99	11.38
Test14	15.23	15.86	99.72%	12.28	9.93	67.71%	17.21	12.45	15.01	9.93
Test15	21.73	18.7	97.21%	10.33	10.87	42.39%	8.89	7.61	8.83	9.68
ALL	17.30	17.43		12.29	12.46		16.97	14.76	13.12	13.23

---

**Algorithm 1:** Altitude map based satellite signal correction algorithm.

---

**Data:** Satellite Signal  $subset^{healthy}$  and  $subset^{suspicious}$

**Result:** Satellite Signal  $set^{corrected}$

init  $set^{corrected}$  as empty set ;

add all satellite signals in  $subset^{healthy}$  into  $set^{corrected}$ ;

**foreach** suspicious satellite  $s_e$  in  $subset^{suspicious}$  **do**

    add  $s_e$  into  $set^{corrected}$  ;

    calculated position result using  $set^{corrected}$ ;

    calculate the height difference ( $hd$ ) between WLS result and reference height from altitude map;

    remember current  $hd$  as current smallest height difference( $shd$ ); remember

    current pseudorange of  $s_e$  as  $\rho_{e_{old}}$  **for** Select  $\delta\rho$  correction from  $-TP$  to  $TP$  **do**

        set pseudorange of  $s_e$  as  $\rho_e = \rho_{e_{old}} + \delta\rho$  calculate position result using using  $set^{corrected}$ ;

        calculate the height difference ( $hd$ ) between WLS result and reference height from altitude map;

**if**  $hd < shd$  **then**

$shd := hd$  ;

            remember current  $\delta\rho$  as  $\delta\rho_{best}$

**end**

**end**

    set pseudorange of  $s_e$  as  $\rho_e = \rho_{e_{old}} + \delta\rho_{best}$

**end**

---

## 第 4 章

# Integration of PDR with Height Aided GNSS method

In previous chapter, we introduced a new altitude map based GNSS method called GNSS + Height Aiding. By using this method, we can correct the pseudorange of the suspicious satellite signals. Therefore, we could improve the positioning accuracy around 5 meters. However, because urban environment is really complicate, there are still a lot of epochs that cannot be corrected by current method. However, as shown in previous section the VE could be positive related to HE. As a result, we can use VE to estimate HE. In other words, with the help of altitude map, we can also evaluate the quality of GNSS positioning result. If the quality can be estimate, the GNSS positioning result could be utilized to integrat with other type of sensors to provide more accurate positioning result. In this chapter, we will focus on GNSS + PDR integration. In Section 4.1 and 4.2, the PDR framework is introduced. In Section 4.3 the proposed PDR system with GNSS + Height Aiding framework is introduced. In Section 4.4, the experiment setup and experiment result is discussed.

### 4.1 The Proposed PDR System

We use Google Nexus 5 as our platform to implement PDR algorithm. The PDR algorithm implemented by us is basically the same as Kakiuchi et al. [3]. The smartphone based PDR implemented by us only use two sensors and they are accelerometer and magnetometer. We do not use the gyroscope in the system because the proposed PDR is put the smartphone in the pocket of trousers. The measurement of the gyroscope becomes very noisy in this setup. The proposed PDR first determines the orientation of device using the measured acceleration and magnetic field. The type and update rate of accelerometer and magnetometer embedded in Google Nexus 5 can be viewed in Tab. 4.1. As discussed in [3], at every measurement of the acceleration and magnetic field, the



Sensor Type	Model	Update Rate
Accelerometer	MPU6515	32 Hz
Magnetometer	AK8963	32 Hz

表 4.1. Sensor embedded in Google Nexus 5.

orientation of the device is estimated and the accelerometer readings are transformed into global coordinate. Afterwards, the step detection, stride estimation and heading orientation are applied subsequently. Steps are detected from the vertical acceleration sequence. At each step, the stride length is estimated using the previous introduced weinberg equation. Then, the moving direction of user is estimated from the acceleration in global E-N dimension. The position is incremented from the previous step according to this direction and distance. It should be noted that the step detection component detects the steps of the left or right leg, depending on in which trouser pocket the device is located. This means that the stride length refers to the distance between two step points made by the same foot. The details of each algorithm are introduced as follows:

For device orientation estimation, we use Andorid API called *getRotationMatrix*, which use sensor measured 3-dimension acceleration data and magnitude field as input and output the transformation matrix which could map the sensor data in local x-y-z coordinate to global E-N-U coordinate. One thing needed to be mentioned is that before pass the sensor measured data to the Android API, we first apply an low-pass-filter to filter out the noise in both acceleration data and magnetic field data. We apply the simplest low-pass-filter in our system. The equation for the low-pass-filter are shown as follows:

$$\begin{aligned}\hat{a}_i &= (1 - \alpha^{acc})a_i + \alpha^{acc}\hat{a}_{i-1} \\ \hat{m}_i &= (1 - \alpha^{mag})m_i + \alpha^{mag}\hat{m}_{i-1}\end{aligned}\quad (4.1)$$

where,  $a$  and  $m$  denote the measured acceleration and magnetic field, respectively. The  $\hat{\cdot}$  indicates the filtered measurements.  $\alpha$  denotes the smoothing factor, which is different with different smartphone. This research used a Google Nexus 5 to provide the measurements. We empirically set  $\alpha^{acc}$  and  $\alpha^{mag}$  as 0.6 and 0.84, respectively. With the help of the Android API, a rotation matrix is calculated. Hence, the rotated measurement of the acceleration can be calculated.

For step detection, we use the  $a_u$ , which is the acceleration value in global U axis. As shown in Fig. 4.1, we detect the dips of vertical acceleration as steps.

In this paper, the definition of stride is a distance of a cycle of walking motion, namely the distance travelled during a same leg to lift again. To avoid the false detection resulting from shallow dips (another step in Fig. 4.1) we set a dip threshold with  $7.5 \text{ m/s}^2$  for our



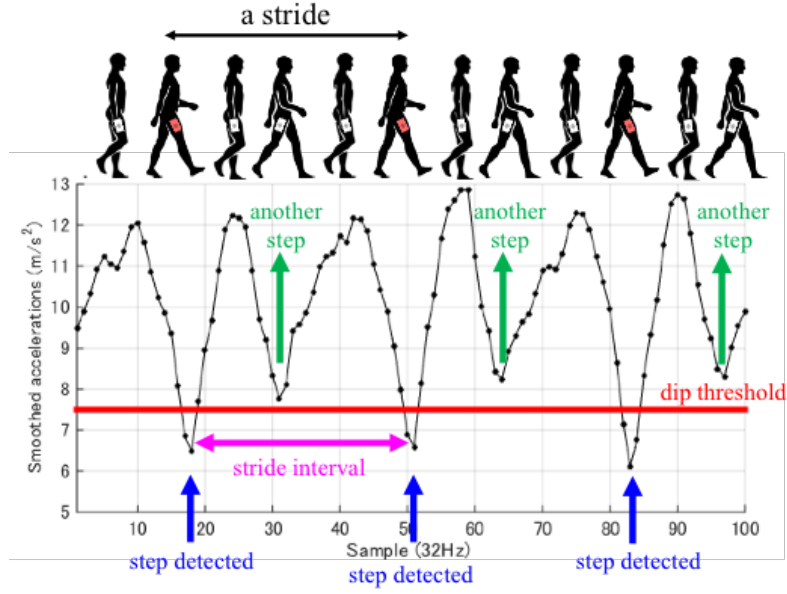


图 4.1. Demonstration of the step detection using smoothed accelerations in vertical direction.

smartphone. Additionally, we also add a threshold of stride interval (pink in Fig. 4.1) to prevent the false detection of the shallow dips. The threshold is that the number of the accumulated samples between two step detections should be larger than one-third of samples in a second. Generally speaking, it is difficult to walk three steps in a second. The update rate of the smartphone used in this paper is 32 Hz. Therefore, the threshold of stride interval is 11 samples in this paper.

For stride length estimation. We Simply applied the Weinberg equation introduced in Chapter 2. The constant  $K$  in our system is 0.73 for male and 0.65 for female.

In this paper, the heading direction is estimated in two steps. We first estimate the motion axis and then find its forward direction. The first step is based on principal component analysis (PCA) in 2-dimentional plane, following the method proposed in [5]. The PCA is applied in the sequence of east and north components of acceleration. We first, smoothed the north and east acceleration data with moving average. Then, we apply the eigen value decomposition of the covariance matrix to them. The resulting first eigenvalue gives the estimation of the direction that should be parallel to the motion axis. The length of time window for PCA in this paper is setted to 2 seconds. Although PCA method can estimate the motion axis, it is difficult to decide which direction of axis is the forward direction. To provide correct forward direction, we use the measurement forward acceleration. Note that the original forward direction acceleration is given by PCA method. To verify the original forward direction is correct or not, we check whether the forward acceleration has a peak or a dip after step detection as shown in Fig. 4.2.

The cyclic patterns of vertical and forward acceleration have a temporal relation [36]. The correct relation is a step detection is closely followed by a forward peak and vice versa as shown in Fig. 4.2.

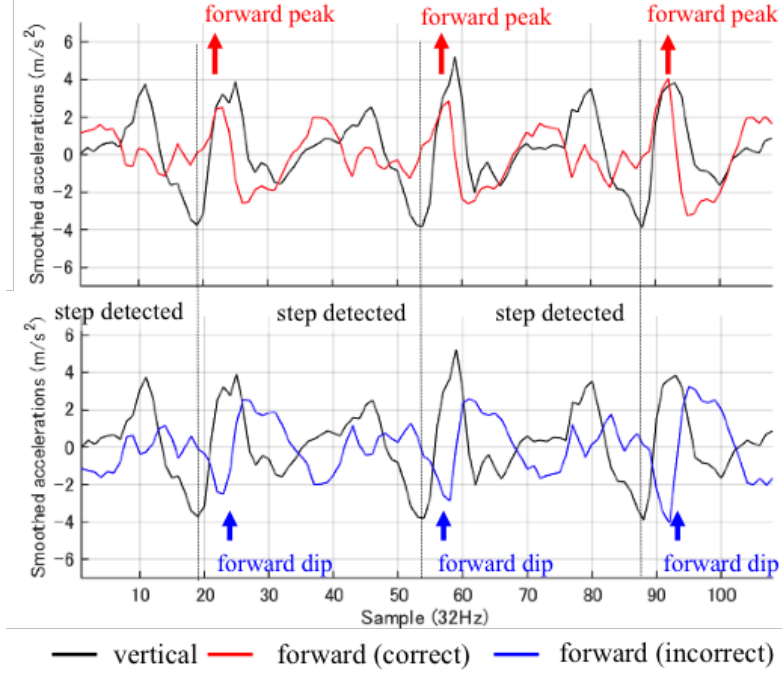


Fig. 4.2. Illustration of the verification of forward direction using forward acceleration.

## 4.2 Evaluation of Purposed PDR System

In this section, we will evaluate the proposed PDR algorithm and we will show it is an effective PDR algorithm and is qualified enough to be used for integration purpose. The Tab. 4.2 summarized the parameter setup used in our experiment. We use Google Nexus 5 (as shown in Fig. 4.3) as our test equipment.

We test our PDR system in a general city environment. Each object represents the trajectory twice. Both male and female are the candidate in our experiment. We put the smartphone in pedestrian's pocket.

The travelled distances of the walking tests are about 420 meters. Fig. 4.4 shows the walking trajectories estimated by the smartphone-based PDR. As shown in Fig. 4.4, the test trajectory is typical for general pedestrians. The shapes of the trajectories are very similar to the ground truth. The performance of Fig. 4.4 is listed in Tab. 4.3.

The mean of heading error is less than 15 degrees in all the objects. With regards to the positioning error, the range is in between about 8 to 32 meters. The positioning error of objects 3 and 4 are larger than that of objects 1 and 2. After the Segment 1,

表 4.2. The PDR parameters setted in the experiment.

accelerometer	sample rate	32 Hz
	lpf $\alpha$	0.6
magnetometer	sample rate	32 Hz
	lpf $\alpha$	0.84
step detection	moving average window size	4 samples
	dip detection window size	32 samples
	dip detection threshold	$7.5 \text{ m/s}^2$
stride length estimation	K	0.71(male), 0.68(female)
moving direction estimation	moving average window size	3 samples
	PCA window size	64 samples

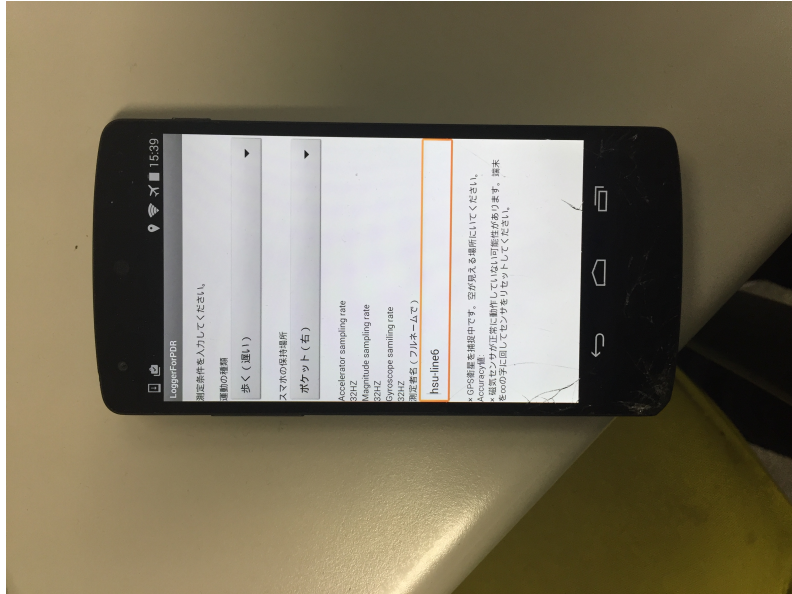


図 4.3. The Google Nexus 5 smartphone.

表 4.3. Performance of the smartphone-based PDR systems.

Object	Heading error		lateral positioning error	
	mean	std.	mean	std.
1	14.45°	15.33°	9.65 m	8.99 m
2	11.44°	17.78°	7.88 m	5.17 m
3	14.80°	16.51°	23.06 m	23.06 m
4	13.94°	17.17°	31.78 m	17.97 m

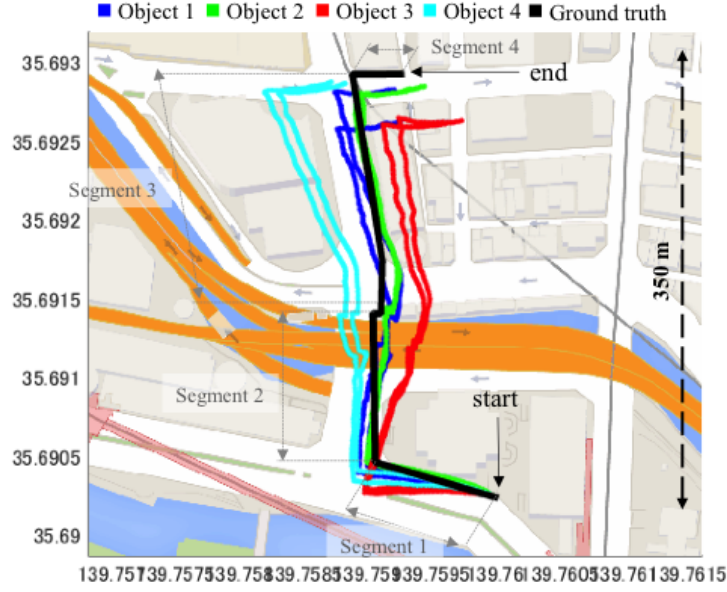


Fig. 4.4. Positioning result of the smartphone-based PDR with 4 objects.

as shown in the bottom of Fig. 4.4, the erroneous heading angle leads the accumulation of the positioning error in Segments 2, 3 and 4. This phenomenon is common in PDR systems. Note that because we use PCA method, the  $180^\circ$  ambiguity problem can not be avoid. For example, the second trail of the Object 1 is a bit shorter than the other trajectory. This is because the  $180^\circ$  ambiguity happened more often in the second trail. Fig. 4.5 demonstrates the heading error of tests of Object 1. The red dot indicates the epoch with  $180^\circ$  ambiguity estimation. As shown in Fig. 4.5, the number of the epoch of the first trial is much less than that in the second trial. The correct estimation of the  $180^\circ$  ambiguity in the second test is about 95%, which is the worst in the eight trials. In overall, the developed smartphone PDR demonstrates not only its capability to provide continues solutions but also potential to indicate the pedestrian motions.

### 4.3 The Proposed PDR with Height Aided GNSS Framework

For PDR + GNSS integration, we use the kalman filter framework that is proposed in [33]. The Kourogı et al. [33] proposed a kalman filter framework for PDR and GPS integration. The state  $x_t$  is  $\begin{pmatrix} x_t \\ y_t \end{pmatrix}$ . The state transition model is:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = \begin{pmatrix} x_t + l_t \cos \theta_t - \sin \theta_t \\ y_t + l_t \sin \theta_t + \cos \theta_t \end{pmatrix} = A_t x_t \quad (4.2)$$

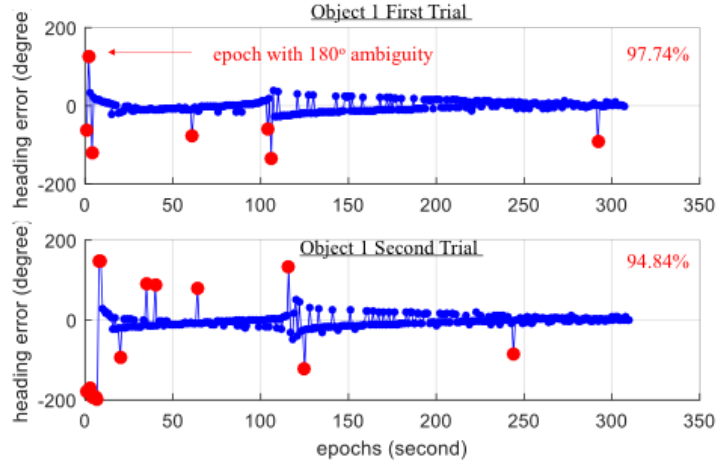


图 4.5. Error of heading angle of the two test trails of Object 1.

where  $A_t = \begin{pmatrix} 1 & 0 & \cos\theta_t & -\sin\theta_t \\ 0 & 1 & \sin\theta_t & \cos\theta_t \end{pmatrix}$

They define error covariance of a priori estimation as:

$$P_{t|t-1} = A_t \begin{pmatrix} P_{t-1} & 0 & 0 \\ 0 & \sigma_{l,t}^2 & 0 \\ 0 & 0 & (l_t \tan\sigma_\theta)^2 \end{pmatrix} A_t^T \quad (4.3)$$

Where  $P_0 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ ,  $\sigma_{l,t}^2$  is the variance in stride length estimation. It is proportion to the  $t$ th stride length. The  $l_t \tan\sigma_\theta$  is the variance to measure horizontal deviation and is set to a constant value.

For measurement error covariance matrix. They define:

$$R_t = \alpha \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} \quad (4.4)$$

Where  $\alpha$  is a constant and  $\sigma^2$  is the square of GPS accuracy. The detail of GPS accuracy has been discussed in chapter 2. The GPS accuracy reveals the reliability of a GPS position result. The GPS accuracy could be in multiple space, but in this application, we only care about the GPS accuracy in horizontal space. The detail of GPS accuracy is discussed in chapter 2. The GPS accuracy contains geometry error factor and pseudorange error factor. The geometry error factor is given by DOP values. The pseudorange error factor is usually related to pseudorange error residuals. In chapter 2, the pseudorange error factor is calculated as  $\sigma_{URE}$ . However, this value is only a statical evaluation and is hardly be used to given an evaluation for GNSS solutions epoch by epoch. In many receiver, the GPS accuracy is evaluated using the following equations:

$$\begin{aligned}
(\text{pseudorange error factor}) &= \frac{\sum_{i=1}^n \rho_i^{err}}{n} \\
\sigma &= \frac{\sum_{i=1}^n \rho_i^{err}}{n} \times \text{HDOP} \times 2
\end{aligned} \tag{4.5}$$

However, based on previous research the conventional GPS accuracy calculation is not accurate enough and has its problems [7]. With the help of altitude map, we can have a more reliable accuracy value for integration purpose. Because we known the error estimation vertically which is VE. We can map this error to pseudorange domain by divide by geometry factor which is VDOP. Then, we can estimate the horizontal accuracy which is also horizontal reliability (HR) as follows:

$$\sigma = HR = VE \times \frac{HDOP}{VDOP} \tag{4.6}$$

## 4.4 Experiment Result

This this Section, we will first evaluated the proposed new accuracy. After that, we will compare the integration result using conventional GNSS with conventional accuracy and Height Aided GNSS with new accuracy.

Fig. 4.4 shows the evaluation of positioning result in our test environment. The white points denotes the ground truth. The color of other points is represent the *HR* level (Tab. 4.4). We set  $T_r$  is set to 10, because in the urban canyon environment, achieving less than 10 meters positioning accuracy is desirable. In this test, the quality of positioning results varied a lot and the proposed new accuracy could be used to evaluated the positioning results in a certain degree. For example, reliable points (red points) are generally distributed near the ground truth but the unreliable points (blue points) are generally distributed far from ground truth (as shown in Fig. 4.4). It also can be seen that the reliable points are more often occurring on the intersection than on sidewalks. Because on the intersection, there are more LOS satellites can be seen and the satellite geometry is better.

Tab. 4.4 also shows the relationship between *HR*, the positioning mean error and the percentage of data within the certain *HR* level. It can be seen that the *HR* given a trustful bounds of the positioning results. The mean error of positioning results at each reliable level is almost less than the upper bound of *HR*. If the positioning results are divided into 6 level based on *HR*, it can be seen that each *HR* contains 12% to 27% of positioning results. In the urban canyon environment, it is hard to calculate positioning results correctly. However, with the help of *HR*, we can recognized 48% of relatively reliable (*HR* less than 20) positioning results.

表 4.4. The color and related  $HR$ . The  $T_r$  is set to 10.

color	red	orange	yellow	green	light blue	dark blue
$HR$	less than $T_r$	$T_r \sim 2T_r$	$2T_r \sim 3T_r$	$3T_r \sim 4T_r$	$4T_r \sim 5T_r$	greater than $5T_r$
mean error	6.20	8.22	16.84	25.86	47.89	84.83
percentage of results	27%	22%	17%	15%	12%	7%

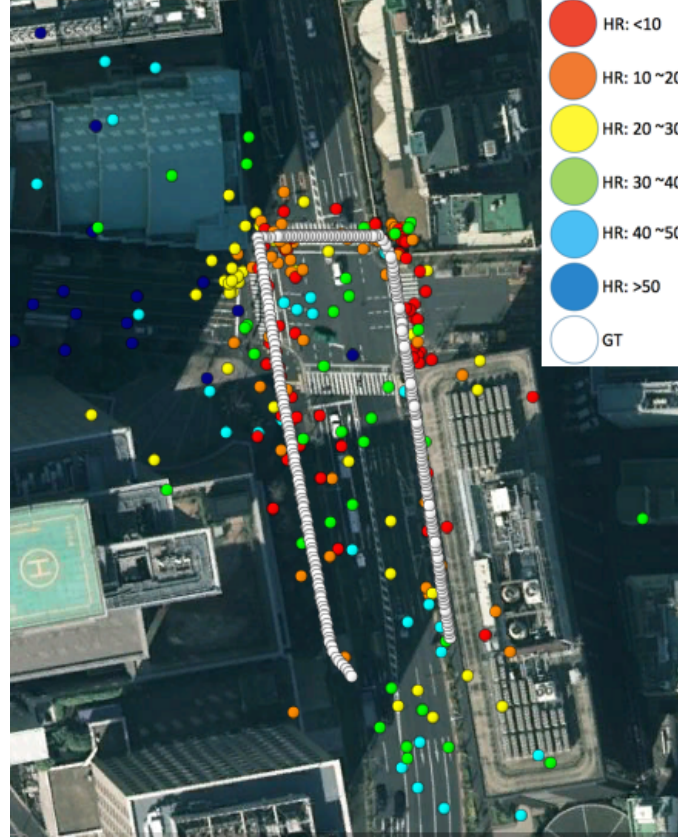
图 4.6. The results of altitude map based evaluation method. The  $GT$  is the ground truth.

Fig. 4.7 shows the accuracy and positioning mean error calculated by Google Nexus 5. It shows that the Google Nexus 5 output accuracy is not correlate with the positioning error. It could have two reasons, one is the conventional accuracy calculation itself is not so correct, the other reason could be that the Google Nexus 5 GNSS result is already passed through some strong filter which makes the positioning result is in consistent with GPS accuracy.

Fig. 4.8 shows that the conventional accuracy calculation and height based accuracy calculation for an raw WLS result. It can be seen that the height based accuracy calculation is more correlate with the position error. When positioning result has large jump, in most time the height based accuracy also has jumps in same period and successfully covered the positioning result line. Moreover, when viewing left part of Fig. 4.8, it can be seen that the conventional accuracy can not identified the epoch in which the positioning



error is large (around gps tow 276400). However, the height accuracy could reveal that those point is unreliable.

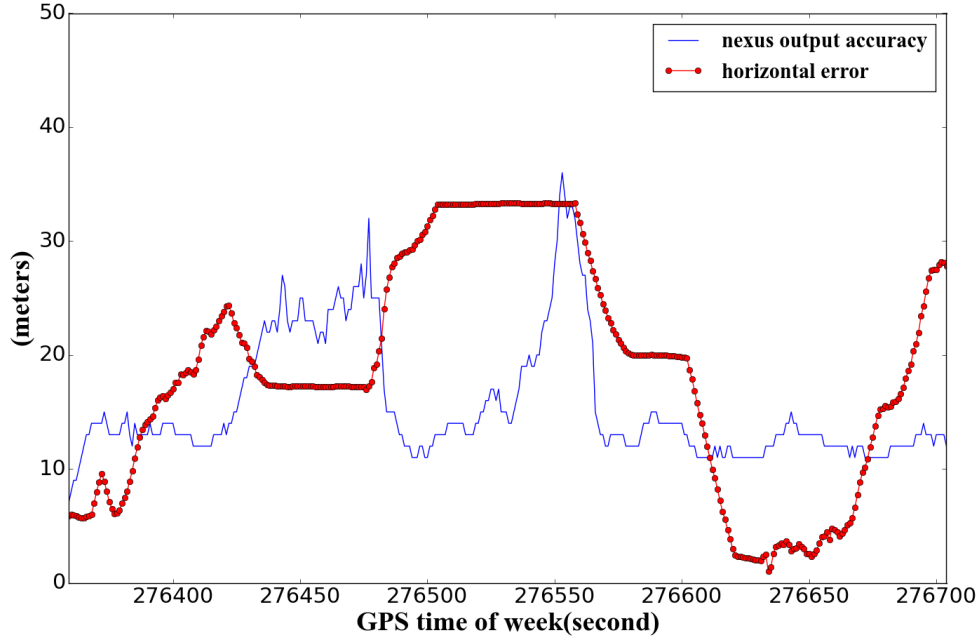


Fig. 4.7. The horizontal positioning error and accuracy value calculated by Google Nexus 5.

Fig. 4.9 and Fig. 4.10 shows the PDR + conventional GNSS, PDR + NEXUS GNSS and PDR + height aided GNSS integration result. In those figures, the first information we can derive is that the HR reliability could help us to filter out the unreliable result. The mean error and standard deviation of Height Aided GNSS result both been minimized if only take the point whose HR is less than 50 into consideration. If the result already is Good (Case 1), the improvement is small. However, if the position is still have more than 10 meters mean error after applying height aiding, such as case 3 and case 4, selecting reliable positioning result by HR could provide 40% mean error improvement.

When observing the integration result, we can see that when the GNSS result is correct in most of epoch, even the conventional WLS + conventional GNSS accuracy could achieve around 4 meters positioning accuracy (case 1). However, by observing case 1, we can see that there are still some jumping point in conventional GNSS + PDR graph (left side link). It is because the conventional accuracy is not reliable enough to distinguish the unreliable positioning result. Nevertheless, the altitude map based GNSS accuracy could distinguish unreliable positioning result. That is why in PDR + Height Aiding based GNSS graph (case 1), the right side link is much smooth than PDR + Conventional GNSS graph. Even though when observing the mean error, there are only 0.7 meters improvement, PDR +



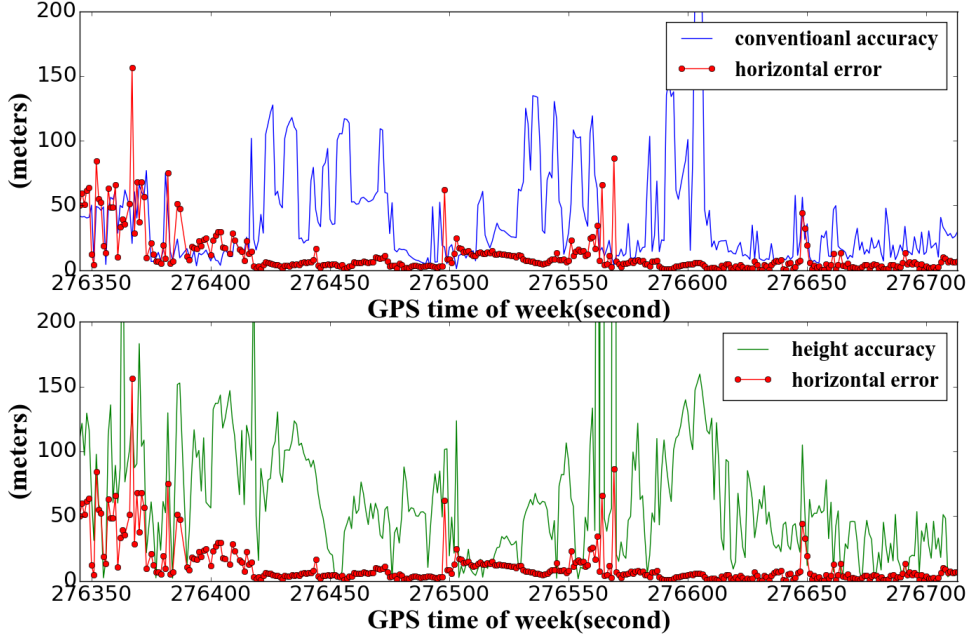


图 4.8. The horizontal error and conventional accuracy and proposed accuracy calculated for WLS solution.

Height Aided GNSS graph is more smooth than PDR + conventional GNSS graph.

When only half epoch of GNSS solution is correct (case 2), the conventional accuracy still could not identified the incorrect position result. When doing integrating, the PDR + conventional integration is blindly trust almost all the GNSS result. When the GNSS result is correct, the integration result is near the ground truth. However, when viewing right side link, we can only a few GNSS result has small positioning error. As a result, the integration result also has large error. We can find that in the right side link the PDR + conventional GNSS graph even does not have smoothed trajectory. Nevertheless, the height aided GNSS method could identify that the GNSS result on the right side link is unreliable and could not be corrected, so on the right side link, the integration algorithm only trust the PDR result. In this case, the PDR result on the right side link does not contain direction bias. Therefore, the PDR + height aiding based GNSS integration is smoothed and relatively correct along all the trajectory. We can see that the mean error of integration result is improved from 7.23 meters to 3.65 meters. One thing need to be mentioned that is PDR usually contains direction bias on all the trajectory. In this case, if the PDR also contains direction bias, the integration result will not be as good as this case.

When observing case 3, we can see that the lake of GNSS result and wrong PDR result will bring problems to the integration result. When viewing the PDR + height aided

GNSS graph. We can see that in the left side link. There are no reliable or correctable GNSS result and the PDR also contained bias in this part. Therefore, the integration result also contained direction bias. When the pedestrian walked close to the intersection, the reliable GNSS result comes in and the direction bias has been corrected. In this case, the PDR + conventional GNSS integration result is almost lost the trajectory shape and the mean error becomes 12.61 meters. However, the PDR + height aided GNSS result could keep the shape in more than 70% of the trajectory and its mean error is 5.12 meters.

The case 4 is a case that is much harder for the WLS + height aiding method. Even though when observing the mean error of the weight least square result, this case is easy than case 3. However, the performance of the proposed method is not as good as case 3. This is because in this case, the satellite geometry is a bit worse than case 2 as a result, the satellite signal correction algorithm is not as effective as that in case 2. Moreover, the resolution of our altitude map based evaluation is also has its limitation. Those two reasons explain why the integration result is not as good as it is in case 3. However, the PDR + height aided GNSS integration still outperform the PDR + conventional GNSS integration. For PDR + height aided GNSS integration, even though the mean error is 6.6 meters, the rough pedestrian walking trajectory is still can be identified. However, in the PDR + conventional GNSS graph, the shape of walking trajectory of the pedestrian is lost and the mean error grows to 10.18 meters.

Finally, by viewing Nexus GNSS result and PDR + Nexus integrating result, we can see that they are both incorrect in those 4 case. The reason could be that the Nexus GNSS result is already be applied strong filter in the chip. Therefore, integrating with PDR is not so useful, because it is hard to evaluate the reliability of GNSS results after they were applied with strong filter.

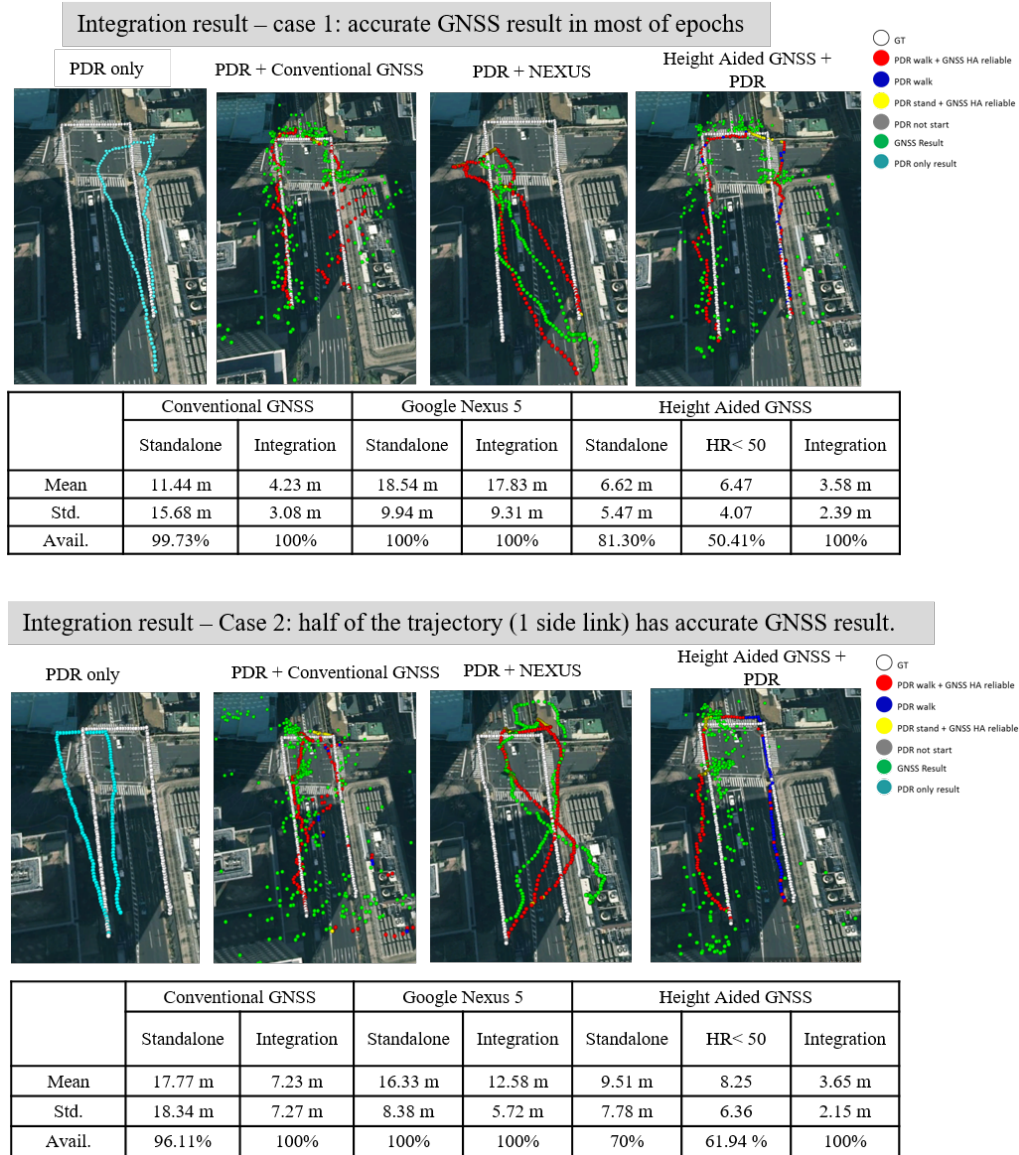
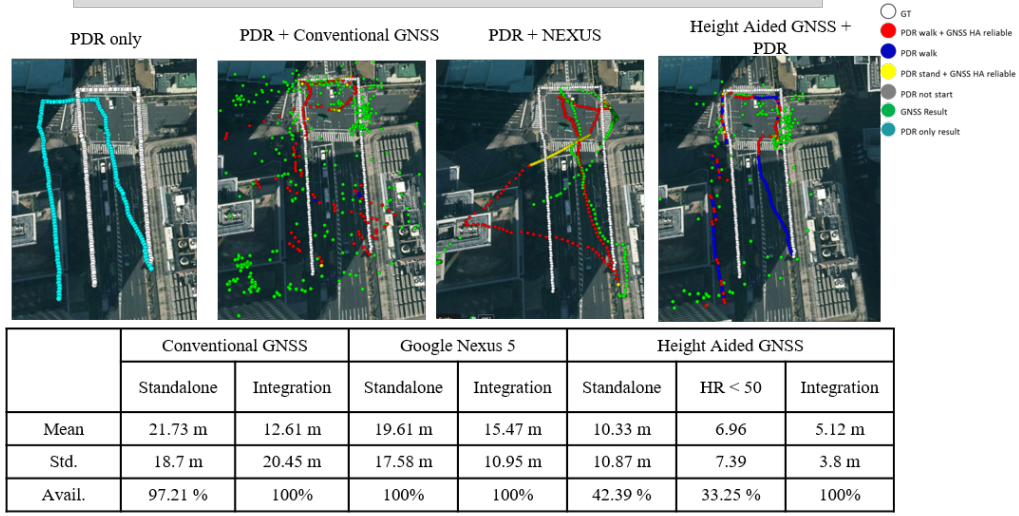
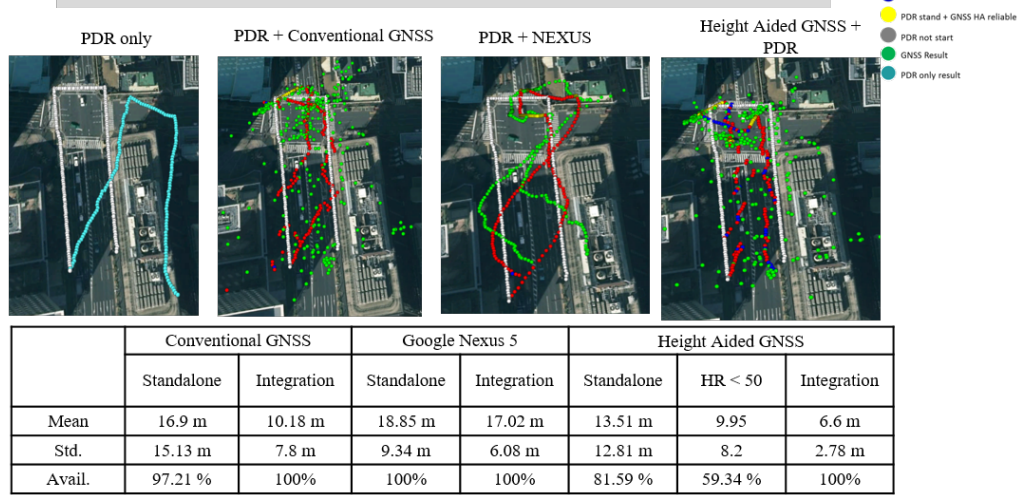


图 4.9. The integration result part A.

Integration result – Case 3: a quarter of epochs has accurate GNSS solution



Integration result – Case 4: GNSS solution is inaccurate in most of the epochs



4.10. The integration result part B.

## 第 5 章

# Conclusion and Future Works

In this thesis, we present a study on pedestrian navigation system by using various kind of navigation technologies in urban canyon environment.

First, we try to improve the positioning accuracy of conventional GNSS method using altitude information. If the pedestrian is walking on the ground, pedestrian's altitude position will not change so much. Therefore, we can pre-build the altitude map to estimate the positioning error in vertical direction. Because the vertical error and horizontal error is close relate to each other, we can add correction value to erroneous satellite signals to minimize the vertical error. If the vertical error is minimized, the horizontal error could also be minimized in a certain degree. Through applying the proposed correction algorithm, we can achieve around 5 meters positioning mean error improvement in a middle class urban canyon environment.

In order to further improve the positioning accuracy in urban environment, we integrate the proposed altitude map based GNSS system with the pedestrian dead reckoning system. For pedestrian dead reckoning system, we implement a conventional PDR system using accelerator and magnetometer on Google Nexus 5 platform. We use PDR system to provide state model and use GNSS system to provide the observation. Previously there are already many researches about PDR + GPS integration, the most challenge task for integration is how to figure out the reliability of the GNSS solutions. The conventional way to calculate the positioning accuracy of the GNSS solutions is not reliable enough. As a result, the integration result is not accurate. However, with the help of altitude map, we can use estimated vertical error to figure out the reliable and unreliable GNSS solutions. Using this way to calculate GNSS positioning accuracy is more reliable than conventional accuracy. We integrate the PDR and GNSS system under a kalman filter framework. Because we can accurately evaluate the GNSS solution, utilizing reliable GNSS epoch to correct the IMU trajectory is effective enough to correct the bias error for PDR system.

In the future, there are still some works need to be done. Currently, we improve the positioning accuracy of the conventional GNSS and evaluate the positioning result. Even

though, we can improve the positioning accuracy of the conventional GNSS method in a certain degree, the positioning error is still large. The Differential GNSS (DGNSS) system could correct the user side clock bias in the urban canyon, therefore improve the positioning accuracy of GNSS system. In the future, we also want to incorporate the DGNSS system in our GNSS system. For PDR system, currently we only use previous developed PDR algorithm. However, the PDR algorithm is still can be improved. Current PDR system only use the magnetometer and accelerator. However, the modern smartphone often embedded with gyroscope. It can be expected that utilizing gyroscope in the PDR system could also improve the accuracy of PDR system. For integration part, currently, we only integrate the PDR and GNSS system under a simple kalman filter framework. However, there are no assurance that the pedestrian's movement is a liner model. As a result, simple kalman filter may be not accurate enough for integration. The extend kalman filter or particle filter should also be tested. Finally, our objective is to build the seamless pedestrian navigation system in the urban environment. Currently, we only studied PDR + GNSS system, In the future, we need to find some ways to integrate the PDR + GNSS system with other systems such as WPS system or image based navigation system to provide more accurate navigation solution. In other word, we need to develop criteria to judge where and when we should trust which system.

# Thanks

This thesis is under the instruction of Professor Kamijo. He gives me the direction and advices not only on the researches but also on my life. Ms. Miwa, Professor Kamijo ' s secretary, also provided huge help to my research life in the lab. Also researcher Gu and Qmo are the people who gives me a lot of important advice on my researchs. Nothing in this thesis can be achieved without their help. To the other members in the lab, Liu, Bao, Mahdi, Ehsan, Wada, Hashimoto, Xie, Wang, Prathana, I want to thank them for companying me through the 2 years and make all these time more meaningful time. Lastly, I want to thank my family and friends for their supporting during this two years.

February 4th, 2016

## 発表文献

### 国際学会

- [1] Yuyang Huang, Li-Ta Hsu, Yanlei Gu, Haitao Wang and Shunsuke Kamijo “Assessment of Outdoor Wi-Fi Fingerprint Calibration using Different GNSS Approaches” International Symposium on GNSS 2015 Kyoto, Japan. September 16-19, 2015.(reviewed)

### 国内学会

- [2] Yuyang Huang, Li-Ta Hsu, Yanlei Gu and Shunsuke Kamijo “GNSS Positioning Correction Method Using Altitude Map” 第13回 ITS シンポジウム 2015, Tokyo, December 3, 2015. (reviewed)

### 論文誌

- [3] Li-Ta Hsu, Yanlei Gu, Yuyang Huang, and Shunsuke Kamijo, “Urban Pedestrian Navigation using Smartphone-based Dead Reckoning and 3D Maps Aided GNSS” IEEE Sensors, November 2015.
- [4] Yuyang Huang, Li-Ta Hsu, Yanlei Gu, Haitao Wang, and Shunsuke Kamijo, “ Database Calibration for outdoor Wi-Fi Positioning System” IEICE Transactions A (Fundamentals of Electronics, Communications and Computer Sciences)( November 2015 submitted )



## Reference

- [1] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*. Lincoln, Mass.: Ganga-Jamuna Press, Dec. 2010.
- [2] S. Miura, L.-T. Hsu, F. Chen, and S. Kamijo, “GPS Error Correction With Pseudorange Evaluation Using Three-Dimensional Maps,” *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–12, 2015.
- [3] N. Kakiuchi and S. Kamijo, “Pedestrian dead reckoning for mobile phones through walking and running mode recognition,” in *2013 16th International IEEE Conference on Intelligent Transportation Systems - (ITSC)*, pp. 261–267, Oct. 2013.
- [4] S. Asano, Y. Wakuda, N. Koshizuka, and K. Sakamura, “A robust pedestrian dead-reckoning positioning based on pedestrian behavior and sensor validity,” in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pp. 328–333, IEEE, 2012.
- [5] U. Steinhoff and B. Schiele, “Dead reckoning from the pocket - An experimental study,” in *2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 162–170, Mar. 2010.
- [6] E. D. Kaplan, *Understanding GPS: Principles and Applications, Second Edition*. Artec House, 2 ed., Nov. 2005.
- [7] N. Drawil, H. Amar, and O. Basir, “GPS Localization Accuracy Classification: A Context-Based Approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, pp. 262–273, Mar. 2013.
- [8] T. Walter, a. T. Walter, and P. Enge, *Weighted RAIM for Precision Approach*. 1995.
- [9] J. Blanch, T. Walter, P. Enge, S. Wallner, F. Amarillo Fernandez, R. Dellago, R. Ioannides, I. Fernandez Hernandez, B. Belabbas, A. Spletter, *et al.*, “Critical elements for a multi-constellation advanced raim,” *Navigation*, vol. 60, no. 1, pp. 53–69, 2013.
- [10] S. Hewitson and J. Wang, “GNSS receiver autonomous integrity monitoring (RAIM) performance analysis,” *GPS Solutions*, vol. 10, pp. 155–170, July 2006.
- [11] S. Hewitson, H. Kyu Lee, and J. Wang, “Localizability Analysis for GPS/Galileo Receiver Autonomous Integrity Monitoring,” *The Journal of Navigation*, vol. 57, pp. 245–259, May 2004.

- [12] T. Iwase, N. Suzuki, and Y. Watanabe, "Estimation and exclusion of multipath range error for robust positioning," *GPS Solutions*, vol. 17, pp. 53–62, Jan. 2013.
- [13] J. Meguro, T. Murata, J. Takiguchi, Y. Amano, and T. Hashizume, "GPS Multipath Mitigation for Urban Area Using Omnidirectional Infrared Camera," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 22–30, Mar. 2009.
- [14] F. Wu, N. Kubo, and A. Yasuda, "Performance evaluation of GPS augmentation using Quasi-Zenith Satellite System," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, pp. 1249–1260, Oct. 2004.
- [15] X.-L. Su, X. Zhan, M. Niu, and Y. Zhang, "Receiver autonomous integrity monitoring (RAIM) performances of combined GPS/BeiDou/QZSS in urban canyon," *IEEE Transactions on Electrical and Electronic Engineering*, vol. 9, no. 3, pp. 275–281, 2014.
- [16] H. Tokura, H. Yamada, N. Kubo, S. Pullen, and others, "Using Multiple GNSS Constellations with Strict Quality Constraints for More Accurate Positioning in Urban Environments," *Positioning*, vol. 5, no. 04, p. 85, 2014.
- [17] Z. Jiang, P. D. Groves, W. Y. Ochieng, S. Feng, C. D. Milner, and P. G. Matos, "Multi-Constellation GNSS Multipath Mitigation Using Consistency Checking," 2011.
- [18] Z. Jiang and P. D. Groves, "GNSS NLOS and Multipath Error Mitigation using Advanced Multi-Constellation Consistency Checking with Height Aiding," Sept. 2012.
- [19] P. D. Groves and Z. Jiang, "Height Aiding, C/N<sub>0</sub> Weighting and Consistency Checking for GNSS NLOS and Multipath Mitigation in Urban Areas," *Journal of Navigation*, vol. 66, pp. 653–669, Sept. 2013.
- [20] D. Maier and A. Kleiner, "Improved GPS sensor model for mobile robots in urban terrain," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4385–4390, May 2010.
- [21] N. Kubo and A. Yasuda, "Instantaneous RTK Positioning with Altitude-aiding for ITS Applications," *Proc. Int. Tech. Meet. Satellite Div. ION GNSS*, pp. 1447–1455, 2007.
- [22] A. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU," in *IEEE International Symposium on Intelligent Signal Processing, 2009. WISP 2009*, pp. 37–42, Aug. 2009.
- [23] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A Reliable and Accurate Indoor Localization Method Using Phone Inertial Sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp '12*, (New York, NY, USA), pp. 421–430, ACM, 2012.
- [24] L. Fang, P. Antsaklis, L. Montestruque, M. McMickell, M. Lemmon, Y. Sun, H. Fang, I. Koutroulis, M. Haenggi, M. Xie, and X. Xie, "Design of a Wireless Assisted Pedes-

- trian Dead Reckoning System The NavMote Experience,” *IEEE Transactions on Instrumentation and Measurement*, vol. 54, pp. 2342–2358, Dec. 2005.
- [25] S. Beauregard, “A Helmet-Mounted Pedestrian Dead Reckoning System,” in *2006 3rd International Forum on Applied Wearable Computing (IFAWC)*, pp. 1–11, Mar. 2006.
- [26] Y. Jin, H.-S. Toh, W.-S. Soh, and W.-C. Wong, “A robust dead-reckoning pedestrian tracking system with low cost sensors,” in *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 222–230, Mar. 2011.
- [27] H. Weinberg, “Using the ADXL202 in pedometer and personal navigation applications,” *Analog Devices AN-602 application note*, 2002.
- [28] S. Shin, C. Park, J. Kim, H. Hong, and J. Lee, “Adaptive Step Length Estimation Algorithm Using Low-Cost MEMS Inertial Sensors,” in *IEEE Sensors Applications Symposium, 2007. SAS '07*, pp. 1–5, Feb. 2007.
- [29] W. Kang and Y. Han, “SmartPDR: Smartphone-Based Pedestrian Dead Reckoning for Indoor Localization,” *IEEE Sensors Journal*, vol. 15, pp. 2906–2916, May 2015.
- [30] H. Bao and W.-C. Wong, “Improved PCA Based Step Direction Estimation for Dead-Reckoning Localization,” in *2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 325–331, Oct. 2013.
- [31] L.-H. Chen, E.-K. Wu, M.-H. Jin, and G.-H. Chen, “Intelligent Fusion of Wi-Fi and Inertial Sensor-Based Positioning Systems for Indoor Pedestrian Navigation,” *IEEE Sensors Journal*, vol. 14, pp. 4034–4042, Nov. 2014.
- [32] D. Navarro and G. Benet, “Magnetic map building for mobile robot localization purpose,” in *IEEE Conference on Emerging Technologies Factory Automation, 2009. ETFA 2009*, pp. 1–4, Sept. 2009.
- [33] M. Kourogi, N. Sakata, T. Okuma, and T. Kurata, “Indoor/Outdoor Pedestrian Navigation with an Embedded GPS/RFID/Self-contained Sensor System,” in *Advances in Artificial Reality and Tele-Existence* (Z. Pan, A. Cheok, M. Haller, R. W. H. Lau, H. Saito, and R. Liang, eds.), no. 4282 in Lecture Notes in Computer Science, pp. 1310–1321, Springer Berlin Heidelberg, Jan. 2006.
- [34] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [35] P. A. Zandbergen, “Accuracy of iPhone Locations: A Comparison of Assisted GPS, WiFi and Cellular Positioning,” *Transactions in GIS*, vol. 13, pp. 5–25, June 2009.
- [36] D. Kamisaka, S. Muramatsu, T. Iwamoto, and H. Yokoyama, “Design and Implementation of Pedestrian Dead Reckoning System on a Mobile Phone,” *IEICE TRANSACTIONS on Information and Systems*, vol. E94-D, pp. 1137–1146, June 2011.