Android Application for Indoor Wi-Fi Layout

Zhengyin Qian

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-74
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-74.html

May 16, 2013

Android Application for Indoor Wi-Fi Layout

by

Qian Zhengyin

B.Eng. National University of Singapore 2012

A thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Engineering
in

Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Ikhlaq Sidhu, Chair
Professor Bernhard Boser

Spring 2013

The thesis of Qian Zhengyin is approved.

Chair Date

Date

University of California, Berkeley

April 2013

Android Application for Indoor Wi-Fi Layout

by

Qian Zhengyin

Abstract

Android Application for Indoor Wi-Fi Layout

by

Qian Zhengyin

Master of Engineering in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Ikhlaq Sidhu, Chair

The goal of this project is to develop an Android Phone App that can be used to measure dimensions of a room and overlay it with a Wi-Fi strength map. This is meant to address suboptimal arrangements of Wi-Fi router/access points, which result in a low speed/unstable/inaccessible Wi-Fi signal coverage in an indoor space. The motivation of the project is to reduce the cost of having a specialist to come to customers property to fix Wi-Fi coverage problems. The App developed will enable Android users to identify such problems themselves for free.

There are two major tasks in this App. The first is to measure the room dimension. Camera and Orientation API calls are used to achieve the first task. A user will point his/her Android phone to subsequent corners of a room. Orientation API call provides the Euler angle reading and a real time algorithm, using trigonometric geometry, computes the lengths of each wall measured. Some simple optimizations are then applied to generate the final dimension and coordinates. All measured rooms will be assigned a specific ID and passed to another Android Activity to be displayed [1].

---

[1] One room per measurement. User is expected to rotate and shift all the rooms on display to form a complete floor plan by themselves.

The second task is overlaying the measured rooms with Wi-Fi strength map. An initial map is generated once a user finishes measuring the room. In addition, the user can collect Wi-Fi strength data at 4 corners of the room to improve the map accuracy. The data is provided by the Wi-Fi API calls and associated to corresponding location. After a user has collected enough data points, the App will perform an interpolation based on finite discrete data points. A second order interpolation is used in this estimation algorithm. The Wi-Fi strength is then color-coded and overlaid on the existing floor plan. .

The methods used in this project are simple and effective. The final product developed is the first working App that combines room dimension measurement and Wi-Fi strength measurement, and will be further embedded by QualComm in their future projects.

Professor Ikhlaq Sidhu
Thesis Committee Chair

Task allocation

This is a team project of 4 student: Zhengyin Qian (myself) and Zeying Xin from EECS department, Lumin Zhang and Loris Dzagoyan from IEOR department. Loris is responsible for nontechnical taskes including poster design, presentation slides formatting and market research. Lumin is responsible for API development that makes the sensor data be available as function calls. Zeying is responsible for Graphic User Interface (GUI) design that represents the resulting Wi-Fi map in a separate Android Activity [2]. I am the main developer of the software that is responsible for utilizing the sensor API calls to measure, process and collect all necessary information and feed them to corresponding floor plan generation and Wi-Fi map interpolation algorithms to produce the results that are needed for GUI display. Lumin and I together worked on the integration of different parts.

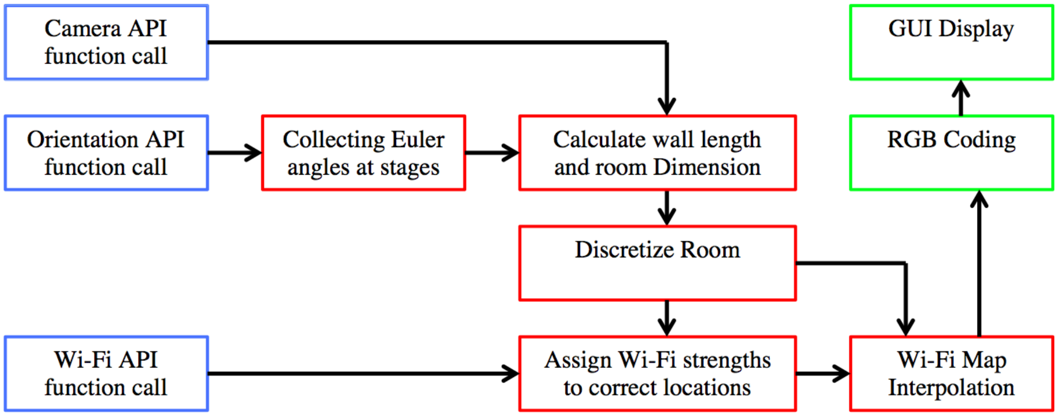


Figure 0.1. Task Allocation

Fig 0.1 shows major tasks of the software. Lumin's responsibilities are enclosed in blue rectangles; my responsibilities are enclosed in red rectangles; Zeyin's responsibilities are enclosed in green rectangles.

---

[2] An Android Actvity is a special class that enables developers to implement single, focused subapplication.

Contents

List of Figures

Chapter 1

Introduction

1.1 Challenge of Wi-Fi Configuration

Evidence has shown the inevitable trend towards an increasingly mobile social dynamic. To lay the infrastructures for the future world, various wireless communication technologies and protocols have been invented and developed. Among the many technologies, Wi-Fi has by far been the most successful and widely deployed wireless communication standard [1]. However, due to the high volume of Wi-Fi subscribers, it is becoming less practical for an Internet Service Provider (ISP) to diagnose the Wi-Fi signal problems for its individual clients and optimize the router configurations in their houses.

On the other hand, it is a challenging task for an ordinary end user to place a limited number of routers optimally. Actually, it is sometimes even difficult for users to realize there is a problem since they have no information about the Wi-Fi signal in their houses. As a result, people usually suffer from suboptimal Wi-Fi configurations and have few choices but to bear with a weak or inconsistent Wi-Fi signal. When the problem becomes too serious, a network specialist must be hired to thoroughly address the problem. However, human resource costs can be expensive and impose unnecessary costs for users. In addition, having designated employees solving this kind of problem is also a waste of manpower for ISPs and raises the operation cost.

Therefore, the cost inccured by manually configuring a WiFi network present an opportunity to develop a more cost and labor-efficient approach to help ordinary users to identify Wi-Fi configuration problems and to position Wi-Fi routers in their places optimally. The approach should require as little professional knowledge as possible and be easily accessible to ordinary people. Bearing these requirements in mind, we have developed an Android smart-phone application that intuitively shows the Wi-Fi signal strength as a color-coded map overlaying the house floor plan.

1.2 Rport Structure

In this report, I will first present a review of existing technologies that may be leveraged to achieve our goal. In the second chapter, I will identify the key challenges in developing the application and discuss the methodology adopted. It will include the two algorithms tried in identifying the room dimension with their individual pros and cons, as well as the algorithm used in generating the Wi-Fi map. In the fourth chapter, I will list the performance measurement matrix and the experiment results of our application. In the last chapter, I will summarize the project and propose some future works.

This paper presents a portion of the overall project, focusing on the technical implementation of the application, especially on the overall program design, room dimension measurement, algorithm implementation and mathematic foundations. Major experiment results and performance indexes will be another emphasize in my report.

Chapter 2

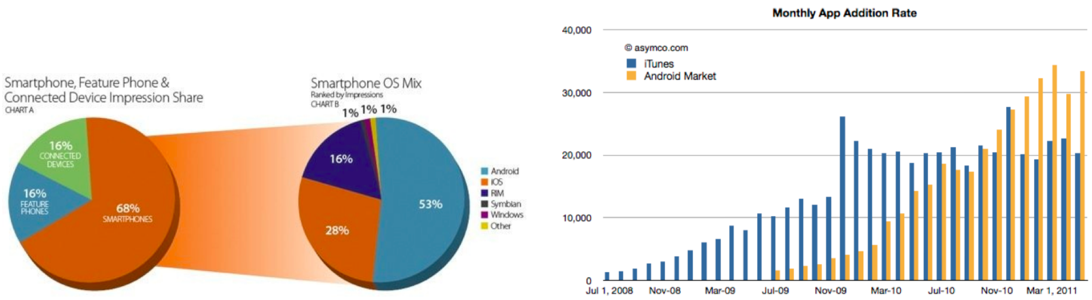
Literature Review

2.1 Current landscape of Android market

For development of a mobile app, it is always important to determine which operation system (OS) to use. Therefore, we have done some research on major operation systems in the market. Statistics has shown the dominancy of Android OS in the global smartphone market [2]. As we can see from Fig 2.1(a), the market share of Android Phone is over 53% by 2011 [3]. In addition, the Android development community is being increasingly active compared to another major player iOS [4]. It is clearly reflected by the number of new Apps on each platform per month as shown in Fig 2.1(b).

2.2 Room Dimension Measurement

As mentioned above, the first part of our project is to measure the room dimension. We are to achieve this goal with only the sensors on a smart phone. Usually the



(a) Market Share (b) Number of New App

Figure 2.1. Comparison of Different Mobile OSs

common sensors a typical smart phone carry are Inertial Measurement Unit (IMU), Gyroscope, Magnetometers and Camera.

Given these constraints, the following two approaches are of particular interest:

2.2.1 Inertial Navigation Approach

Inertial navigation systems (INS) were originally developed for rockets. American rocket pioneer Robert Goddard experimented with rudimentary gyroscopic systems. Dr. Goddard's systems were of great interest to projects like the Lunar Landing Program [5]. The systems entered more widespread use with the advent of spacecraft, guided missiles, and commercial airliners.

Technical wise, an INS is a navigation aid that uses a computer, motion sensors (accelerometers) and rotation sensors (gyroscopes) to continuously calculate via dead reckoning the position, orientation, and velocity of a moving object without the need for external references. It is widely used on vehicles such as ships, aircraft, submarines, guided missiles, and spacecraft [6].

As shown in Fig 2.2, the task of navigation is to estimate the orientation and position of the vehicle in the global axis in terms of the projection matrix P and

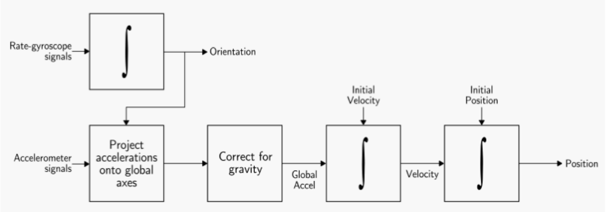


Figure 2.2. INS Algorithm

translational vector t, respectively [7]. It will be very beneficial in our project if we can analogue this idea of INS and ask the user to move the phone along the perimeter of his house, we can easily get the room dimension.

2.2.2 Vision Approach

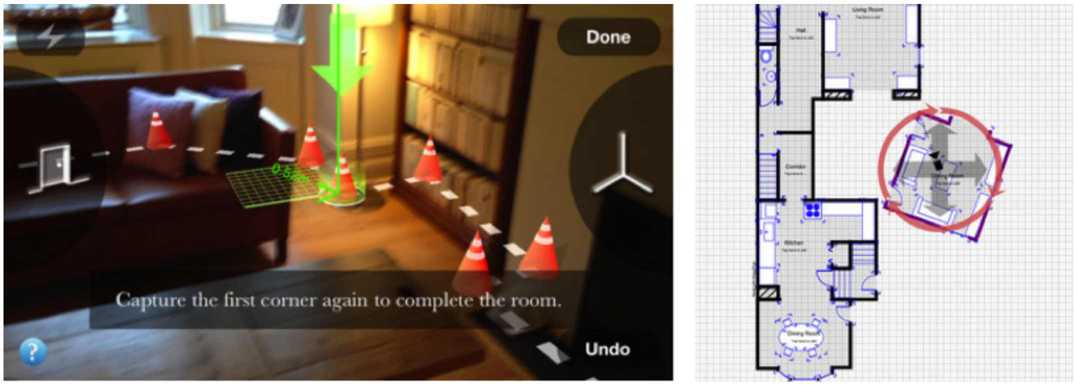
An alternative approach that may be used in our project to measure the room dimension is based on vision. In this approach, user can utilize the camera view to point his phone at a corner of his house. Based on the orientation API reading, we may estimate the length using simple trigonometry.

There are two mobile applications that aim to measure the length using similar approaches. One is called ”Magic Plan” on iOS and the other is called ”SmartMeasure” on Android [8].

2.2.3 Respective Limitations

For the INS approach, the main constraint comes from the low precision of smart phone sensors. Because double integration is involved, small error can accumulate and make the actual measure insignificant [9]. Therefore, whether the accuracy of the smart phone sensors will affect the feasibility of the INS approach remain questionable.

For the vision approach, there is a key parameter, height, which is unknown


Capture the first corner again to complete the room.

(a) Capture the dimensions.

(b) Layout the rooms.

Figure 2.3. Magic Plan

and can not be measured. Therefore, both the existing Apps mentioned above use estimation for these parameters. But this could potentially result in great inaccuracy.

2.3 Wi-Fi Map

As mentioned above, the second part of project is to estimate a Wi-Fi Strength Coverage map based on finite measurement points. Therefore, it is essential to understand the Wi-Fi strength property and its decay pattern.

2.3.1 Wi-Fi Strength Measurement

As all the typical radio frequency signals, Wi-Fi strength is measured in dBm. dBm is an abbreviation for the power ratio in decibels (dB) of the measured power referenced to one milliwatt (mW). It is used as a convenient measure of absolute power because of its capability to express both very large and very small values in a short form. Comparing dBW, which is referenced to one watt (1000 mW), the higher the dBm on a receiver, the stronger the signal strength is [10].

The range of -84 to -96 dBm on receivers is usually the threshold at which point the signal is not detectable any more. A lower rating means a more sensitive receiver, being able to receive and decode weaker signals. Typical range of wireless received signal power over a Wi-Fi network (802.11 variants) is in the range of -60 dBm to -90 dBm[1] [11].

2.3.2 Wi-Fi Strength Decay Pattern

When wireless signals leave a WiFi hotspot and travel to a computer or other wireless devices, the signal strength decays as it passes through air or solid objects [12]. This is why a computer may be unable to pick up the signal from a hotspot that is more than a couple of rooms away. The amount that the signal falls off depends on the medium it passes through; for example, whether it passes through a wall, door or window.

Both signal frequency and range between the end points of the medium affect the amount of attenuation. As either frequency or range increases, attenuation increases [1]. Unlike open outdoor applications based on straightforward free space loss formulas, attenuation for indoor systems is very complex to calculate. The main reason for this difficulty is that the indoor signals bounce off obstacles and penetrate a variety of materials that offer varying effects on attenuation.

---

[1] We majorly concern about signal strength in this project rather than bit rate. Because bit rate depends on more complex issues like data traffic protocols and transmission error rate. Therefore, signal strength is a more reliable metrices here.

Chapter 3

Methodology

In this chapter, I will proceed to identify the key challenges of this project and illustrate the methodology and implementation details.

As mentioned in previous chapters, the two main components of this project are to measure the dimension of a room using smart-phone sensors API calls and overlay the room with color-coded Wi-Fi strength map.

I will first explain the two algorithms I implemented and tested for measuring the room dimension. Followed by an assessment of the pros and cons of the each approach. Then I will explain the implementation of color-coded Wi-Fi map and the interpolation algorithm involved.

3.1 Room Dimension Measurement

The main task in this step is to get a measurement of the dimensions of each room in a house, so that the user can reassemble the rooms to form a house floor plan. The fundamental challenge in this step is ensuring the robustness[1] of the measurement using only the sensors in a smart-phone and with as little user input as possible.

To achieve high robustness, the algorithms and sensors we used should be insensitive to the environment and should propose fewer or simpler requirements on users.

In addition, we claim the absolute accuracy of the measurement is of less importance. Since the user of our application should not care too much about the exact room dimension but rather the Wi-Fi signal strength distribution.

3.1.1 Inertial Approach

In this algorithm, I exploit the existing ideas of inertial navigation from the aerospace field. The general idea is to use the measurement of the phone orientation to translate the acceleration of the phone into a fixed ground coordinate system. With these acceleration measurements, user can measure a distance [2] by moving the phone along a specific trajectory.

Fig 3.1 illustrates the idea graphically. The O_b coordinate is the body coordinate carried by the smart phone. Its orientation and position are fixed relatively to the phone but will change accordingly as the phone moves. The O_g denotes the ground coordinate that is fixed.

My goal is using the angular velocity $\omega = \begin{bmatrix} p & q & r \end{bmatrix}$ and acceleration measurement $a_b = \begin{bmatrix} a_{bx} & a_{by} & a_{bz} \end{bmatrix}$ from the body coordinate to calculate the absolute displacement in the ground coordinate $D = \begin{bmatrix} X_g & Y_g & Z_g \end{bmatrix}$ through a Projection

---

[1] Robustness here is referring to a consistent measurement (i.e. the variance of different measurement instances should be small).

[2] In this project, it will be the wall distance of a room.

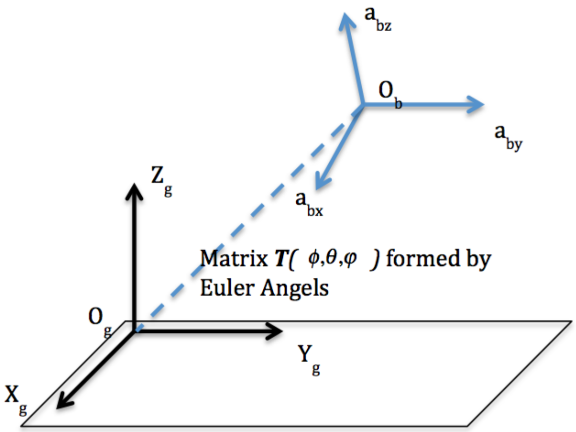


Figure 3.1. Coordinate System for Inertial Approach

Matrix T, which depends on the Euler angle $\begin{bmatrix} \phi & \theta & \psi \end{bmatrix}$. The API calls I utilized in this approach to measure the Euler angle and linear acceleration are orientation API call[3] and accelerometer API call.

The relationship between the quantities in discrete time domain is therefore:

$$D = \sum_{i=0}^{K} P\left(\phi, \theta, \psi\right) \cdot a_b\left(i\right) \cdot \Delta t^2$$

Similarly, the linear time invariant (LTI) system describing the dynamic between body velocity $\begin{bmatrix} u & v & w \end{bmatrix} = \sum a_b$ and the ground displacement vector D is summarized by the next three coupled ODEs, where the big matrix [4] is the projection matrix T.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - s_\psi c_\phi & c_\phi s_\theta c_\psi + s_\psi s_\phi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\psi c_\phi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Therefore, we had a complete structure of the initial measurement approach from the body coordinate acceleration to ground coordinate displacement, which is of our goal.

---

[3] The Android Open Source Community has developed a set of function calls that provide developers direct access to the Eular angle (instead of angular velocities) of the phone.

[4] Note that, the $s_{subscript}$ and $c_{subscript}$ are shorthand for $\sin(\cdot)$ and $\cos(\cdot)$ functions.

3.1.2 Inertial Navigation Approach Assessment

The inertial approach depicted above has a main advantage as it is basically functioning as a virtual ruler. Therefore, by simply moving the phone against the wall, one can theoretically capture the full profile of the wall regardless of any irregularity. That is, even if the wall is not a strait line or even the wall is curving, this algorithm should be able to capture the actual shape.

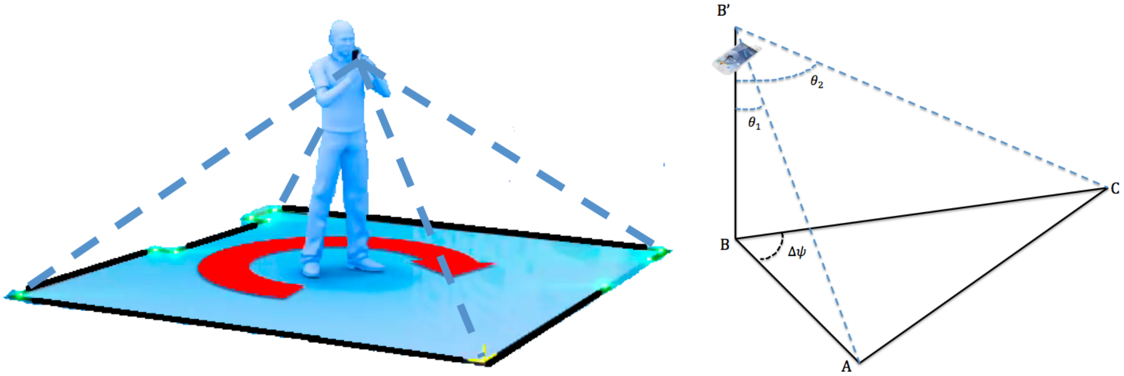
The main drawback of this approach is its measurement highly depends on the performance of the accelerometer. Because a double summation is involved in this algorithm, small noise can be amplified quite significantly. However, the accelerometers in most smartphones are usually quite lousy. The reading is usually sensitive towards the way users use it and the environment disturbance. Therefore, the robustness of the inertial approach can hardly be guaranteed. We will investigate this drawback further in the next section.

3.1.3 Camera Approach

This approach utilizes the ideas from simple geometries. As illustrated in Fig 3.2(a), the user is asked to point the center of the phone (by pointing the center of a camera screen on the screen) to consecutive corners of the room. Then, based on the difference of Euler angle and the assumption of the height of the user we can calculate the wall distance.

To better explain the algorithm, we abstract Fig 3.2(a) to Fig 3.2(b). Length AC represents the wall we want to measure (A and C are the corners of the wall). Length BB' represents the height the user is holding the camera. AB and BC represent the distance from the standing point to the corners.

Then we ask the user to first point his phone to corner A and record down the



(a) Illustration (b) Abstraction

Figure 3.2. Camera Approach

orientation API reading (i.e. the Euler angle), where the θ channel measures the angle $\angle AB'B$ ($\angle\theta_1$). Secondly, we ask the user to rotate the phone to point to corner C. Similarly, record the θ channel measurement, which represents $\angle CB'B$ ($\angle\theta_2$). And during the rotation from first angle to second angle, record down the difference of ϕ channel, which represents $\angle ABC$ ($\angle\Delta\psi$).

Lastly, we assume we know the length BB'. Then we can calculate the wall distance AC as following:

$$AB = BB' \cdot \tan\left(\angle AB'B\right)$$

$$CB = BB' \cdot \tan\left(\angle CB'B\right)$$

$$AC = \sqrt{AB^2 + CB^2 - 2 \cdot AB \cdot CB \cdot \cos(\angle ABC)}$$

3.1.4 Camera Approach Assessment

The main advantage of this approach compared with the inertial approach is it gets rid of the heavy dependency on accelerometer and transfers the dependency mainly to orientation API, which is usually a relatively accurate and robust sensor.

But one major limitation of the algorithm is the measurement $AC = f\left(BB'\right)$, where BB' is a rough guess and can vary largely according to different users. In order to overcome this big limitation, I have implemented a calibration feature. In the calibration stage, we measure a known distance (i.e. we know AC ahead of time) and estimate the height $BB' = f^{-1}\left(AC\right)$. That is, the actual BB' is calculated as:

$$BB' = \left(\frac{AC^2}{\tan^2(\angle AB'B) + \tan^2(\angle CB'B) - 2\tan(\angle AB'B)\tan(\angle CB'B)\cos(\angle ABC)}\right)^{\frac{1}{2}}$$

where the three angles are provided by the orientation API call and the ground truth AC is provided by the user.

Another drawback of this approach is that only a distance between two points is measured. Any additional information between these two points is lost (e.g. curvature of the wall). However, as mentioned above, since the exact shape of the room is not the main concern of the project, this lost is not critical.

3.2 Color-coded Wi-Fi Map Generation

The second main step of this project is to measure the Wi-Fi strength and interpolate the measurement to a continuous color-coded map.

The main challenge in this step is to fill the continuous space with finite measurement. Therefore, certain algorithm is needed to interpolate the finite data points and estimate the rest space without actual measurement.

To address the challenge, we will first discretize the continuous room to square grids with about 0.5 meter in dimension. For example, a room measured with dimension of 10.2m × 7.9m should be discretized in to a 20 by 16 grids with cell size of 0.51m × 0.4938m. Then the user may go to the four corners of the room in the instructed order to gather more data. The program will then probe the Wi-Fi strength at those particular points and assign the measurement to that grid. With these measured data, the estimated map will be updated using following quadratic interpolation algorithm:

$$E_{i,j} = \sum_{k} \left[\left(\frac{1}{dis_{i,j,x,y}} \right)^2 \cdot M_{x,y} \Big/ \sum_{k} \left(\frac{1}{dis_{i,j,x,y}} \right)^2 \right]$$

where:

$E_{i,j}$ is the estimated Wi-Fi strength at location (i, j);

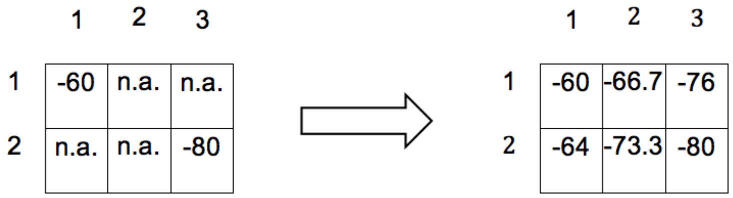
$M_{x,y}$ is the measured Wi-Fi strength at location (x, y);

Summation variable k is the total number of data points measured;

$dis_{i,j,x,y}$ is the Euclidian distance from measured location (x, y) to the estimation location (i, j) and is calculated as: $dis_{i,j,x,y} = \sqrt{(i-x)^2 + (j-y)^2}$.

As a simple numeric example, assume we are in a 2 by 3 discretized room, and the location $(1, 1)$ and $(2, 3)$ have measured Wi-Fi strength of -60 dBm and -80 dBm respectively as shown in the left table of Fig 3.3. Then the estimated Wi-Fi strength at location $(1, 2)$ is:

$$E_{1,2} = \left[-60 \cdot \left(\frac{1}{1} \right)^2 - 80 \cdot \left(\frac{1}{\sqrt{2}} \right)^2 \right] \Big/ \left[\left(\frac{1}{1} \right)^2 + \left(\frac{1}{\sqrt{2}} \right)^2 \right] = -66.7$$



	1	2	3
1	-60	n.a.	n.a.
2	n.a.	n.a.	-80

	1	2	3
1	-60	-66.7	-76
2	-64	-73.3	-80

Figure 3.3. Numeric Example of Wi-Fi Interpolation Algorithm.

Chapter 4

Discussion

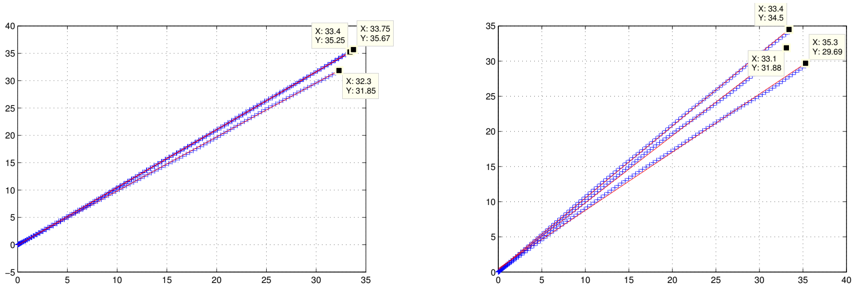
4.1 Floor Plan Measurement

In this section, I will first present and discuss the simulation results using the Inertial Navigation Approach in measuring the floor plan, together with the comparison with actual experimental data. Then I will proceed to discuss the results obtained using Camera Approach and the performance improvement of adding the calibration feature.

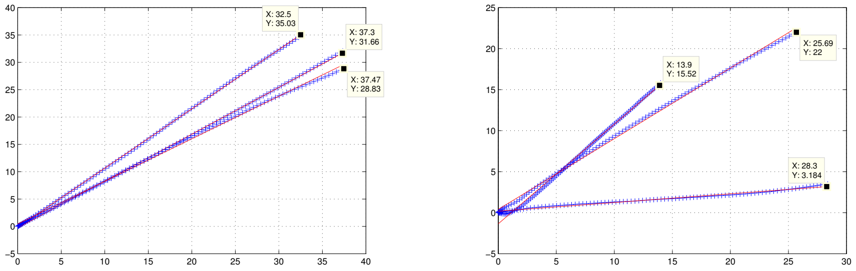
4.1.1 Inertial Navigation Approach Simulation

Simulation is run in Matlab and the input are generated using *randn* function with different means and standard deviations. The mean value is used to represent the general direction of movement and orientation of the phone. The standard deviation of the randomness is used to represent the lousiness of the sensors. I measured the sampling rate provided by the sensor APIs are very close to 0.2 sec (5Hz), so I set $\Delta t = 0.2 sec$. And for a better visualization, the mean direction of movement is set to 45°. The first 20 out of the total pseudo 100 data points [1] of acceleration have a mean of $0.5 m/s^2$ and represent the user speeding up the phone. The rest of the acceleration has a zero mean and represent the user moving the phone in a constant speed. I then varied the sensor noise level (by setting σ_{acc} [2] and σ_{oren}[3]) to see the performance of the algorithm.

Firstly, I fixed the noise level of the accelerometer API and vary the noise level of orientation API. As we can see from Fig **??**, the algorithm is indeed very robust with respect to orientation API noise. Even when the standard deviation increased to $\sigma_{oren} = 10$, the output is still quite reasonable. Significant distortion is only seen when σ_{oren} increases to multiple of tens (as shown in Fig 4.1(d) [4]).



(a) $\sigma_{acc} = 0.1$ and $\sigma_{oren} = 0.1$ (b) $\sigma_{acc} = 0.1$ and $\sigma_{oren} = 1$



(c) $\sigma_{acc} = 0.1$ and $\sigma_{oren} = 10$ (d) $\sigma_{acc} = 0.1$ and $\sigma_{oren} = 50$

Figure 4.1. Varying the Noise Level of Orientation API.

---

[1] This is indicating a 20 sec (100 sample times 0.2sec/sample) of simulation duration.

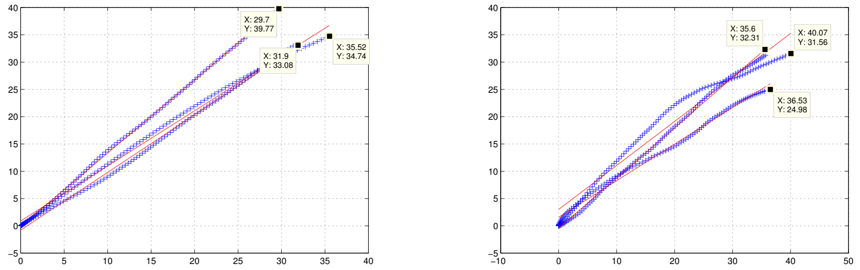
[2] Representing the sensor noise of accelerometer API.

[3] Representing the sensor noise from orientation API.

[4] All units are in meters. For example, 4.1(a) represents a 40 meter by 35 meter squre space.

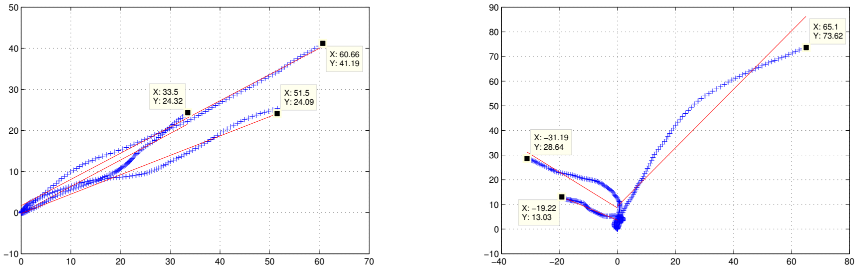
Notice, since the simulation is based on random generated data characterized by its mean and variance, there's no ground truth to compare with. Therefore, the simulation result should be compared to zero variance case, which is an ideal 45 degree line goes from (0, 0) to (36.2, 36.2).

Next, I fixed the noise level of the orientation API and vary the noise level of the accelerometer API. As we can see from Fig 4.2, the algorithm is very sensitive to even relatively small increase of σ_{acc}. When the noise level increases to 0.8, the results began to deviate quite significantly. When I further increase it to 1.5, the results were totally unusable as shown in Fig 4.2(d).



(a) $\sigma_{acc} = 0.2$ and $\sigma_{oren} = 0.1$

(b) $\sigma_{acc} = 0.5$ and $\sigma_{oren} = 0.1$



(c) $\sigma_{acc} = 0.8$ and $\sigma_{oren} = 0.1$

(d) $\sigma_{acc} = 1.5$ and $\sigma_{oren} = 0.1$

Figure 4.2. Varying the Noise Level of Accelerometer API.

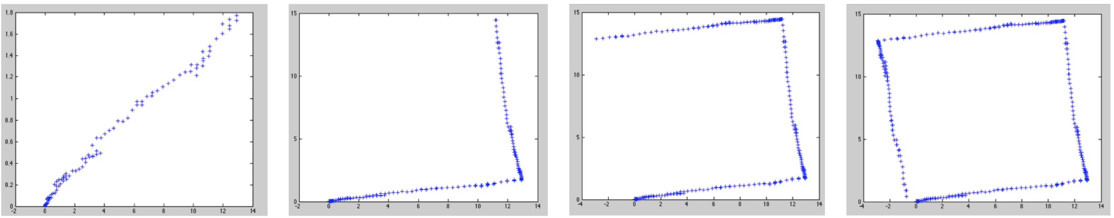
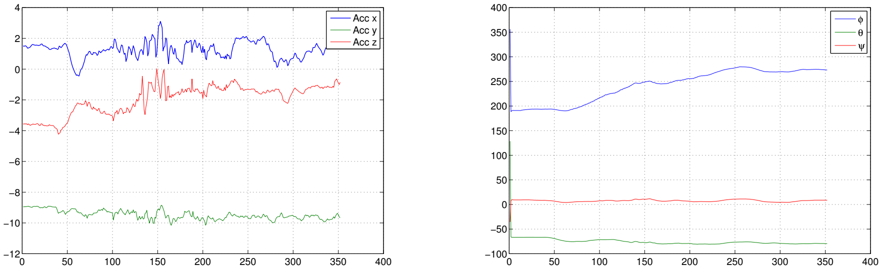


Figure 4.3. Best Simulation Result in Genrating a Room.

4.1.2 Inertial Navigation Approach Experimental Results

Therefore, we can conclude that the algorithm will only work properly given the condition of low accelerometer API noise. Ideally, we should be able to measure the room dimension as shown in Fig 4.3. However, the actual collected data shows a quite low variation from the orientation API but quite lousy readings from the accelerometer API, as shown in Fig 4.4. As a result of feeding the actual data into the algorithm, the outcome is not very satisfying. As Fig 4.5 shows, the actual movement of the phone is a straight line along the vertical direction. However, the algorithm outputs has quite a big deviation.



(a) Accelerometer API Channel

(b) Orientation API Channel

Figure 4.4. Actual Data Collected from Phone Sensors.

X-Axis
X: -0.61073
Y-Axis
Y: 0.7149133
Z-Axis
Z: 9.782457

(a) Screenshot of Data Collection

(b) Experimental Outcome

Figure 4.5. Inertial Navigation Approach with Actual Data

4.1.3 Camera Approach Results

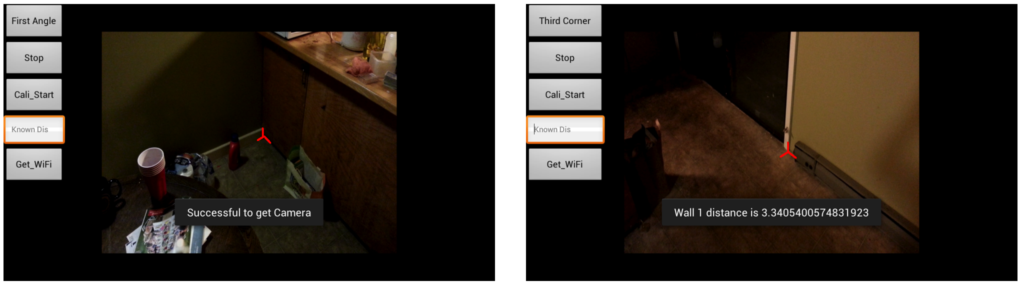
Since the high noise level of the accelerometer API, the Inertial Navigation Approach was finally abandoned. But by observing the low noise level of the orientation API, we can expect a good outcome using the Camera Approach.

Following the discussion in 3.1.3, I implemented the Camera Approach. Initially, the method assumes a fixed BB' (the height the user is holding the camera) and calculate the geometry using that assumption. However, the actual BB' varies from person to person significantly (usually ranges from 1.40m to 1.70m), and even the same person may hold the camera at some different hight from time to time. Therefore, the results without calibrating the assumption is not very satisfying. However, with the calibration feature added, I significantly improved the accuracy of the assumption and hence the accuracy of the whole approach. As measured in a numbers of experiments, the error percentile ($e = \frac{L_{measured} - L_{actual}}{L_{actual}}$) can be controlled below 10% [5]. Fig 4.6 shows the screenshot taken during the steps of using the Camera Approach. We points the center of the screen to each 4 corners of a room, as shown

---

[5] Since the Camera Approach requires the user to point the center of the screen to the corner, the error is measured based on the assumption that the user is able to do the pointing action reasonablly accurate.

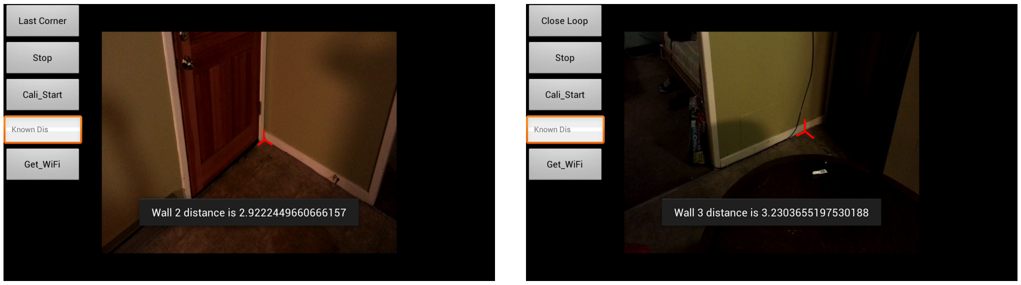
from Fig 4.6(a) to Fig 4.6(d). Then we point it back to the first corner to close the loop, as shown in Fig 4.6(e). Lastly, we press the *DisplayRoom* button and the room dimension is displayed as shown in Fig 4.6(f) [6].



Successful to get Camera

Wall 1 distance is 3.3405400574831923

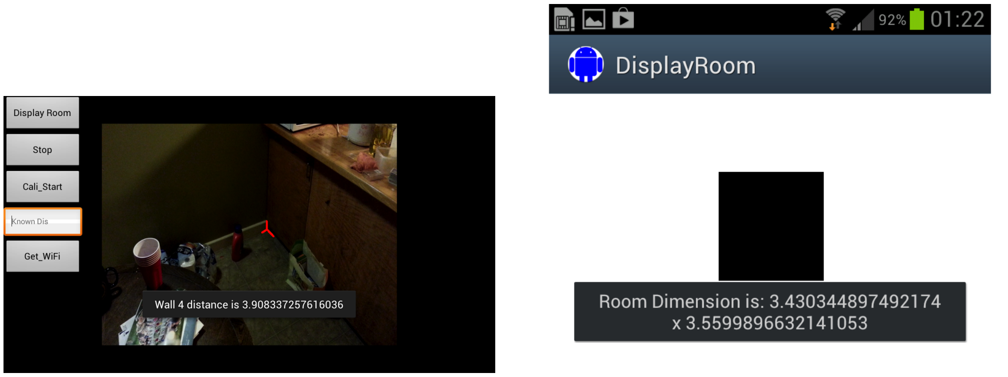
(a) Meausring First Corner (b) Meausring First Corner



Wall 2 distance is 2.9222449660666157

Wall 3 distance is 3.2303655197530188

(c) Meausring First Corner (d) Meausring First Corner



Wall 4 distance is 3.908337257616036

Room Dimension is: 3.430344897492174 x 3.5599896632141053

(e) Experimental Outcome (f) Experimental Outcome

Figure 4.6. Camera Approach Floor Plan Generation

---

[6] Note that, this black box shown in Fig 4.6(f) is to be covered by the estimated Wi-Fi Map.

4.2 Wi-Fi Map

Strictly following the quadratic interpolation algorithm described in last chapter, I measured the Wi-Fi strength at the four corners and a relatively middle point of a measured room and carried out the interpolation. The strengths measured at the corners are -60dBm, -80dBm, -55dBm and -48dBm from the upper-left corner clockwise. The strength in the middle is about -50dBm. Then a standard RGB color coding scheme is applied on the interpolated map. Result is shown in Fig 4.7.

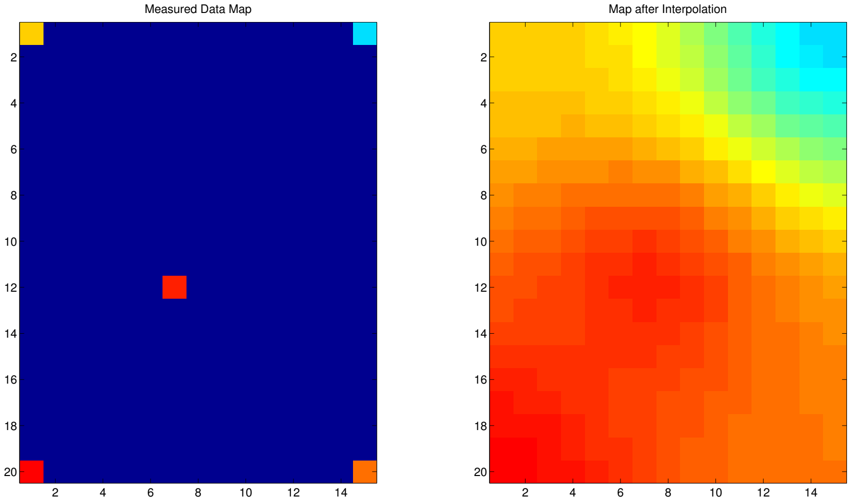
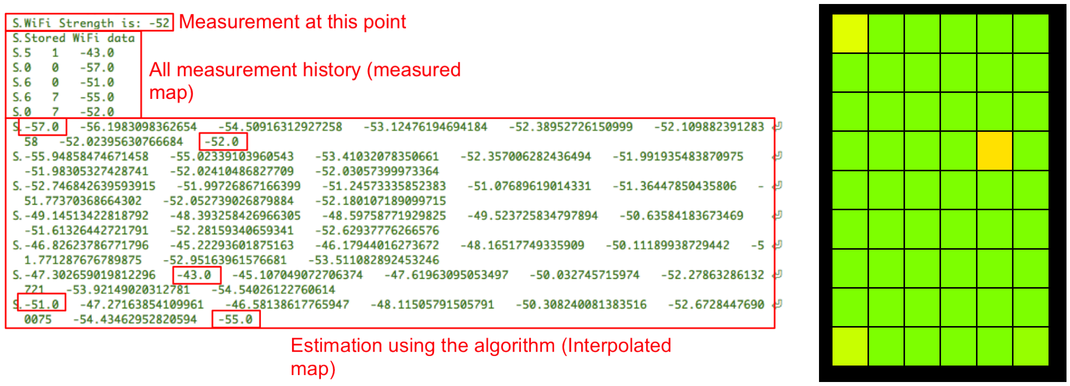


Figure 4.7. Interpolated Wi-Fi Map Using RGB Color-Coding.

I have also carried out actual experiments with this algorithm. As shown in Fig 4.8(a), the first red box is the current measured Wi-Fi strength. The second red box shows all previously measured data (the first two numbers represent the cell index. The last number is the measured Wi-Fi strength corresponding to that index cell.). This measurement history forms the ”measured map” as shown in Fig 3.3. The bottom big red box is the interpolated Wi-Fi map based on the actual data (The desired ”interpolated map” as in Fig 3.3). As we can see, at the indexes where actual data are available, the algorithm just purely trust the actual data. At indexes where there is no actual data, the cell is filled using the interpolation algorithm described in section 3.2.

Fig 4.8(b) shows the representation of an actual measurement in GUI[7]. Notice, the current RGB coding implemented in GUI is quite coarse. It only changes color when the WiFi strength differs more than 10dBm. My teammate Zeying may discuss this in detail in her report focusing on GUI part of the project.



```
S.WiFi Strength is: -52
S.Stored WiFi data
S.5   1   -43.0
S.0   0   -57.0
S.6   0   -51.0
S.6   7   -55.0
S.0   7   -52.0
S.-57.0   -56.1983098362654   -54.50916312927258   -53.12476194694184   -52.38952726150999   -52.109882391283 ↵
  58   -52.02395630766684   -52.0
S.-55.94858474671458   -55.02339103960543   -53.41032078350661   -52.357006282436494   -51.991935483870975   ↵
  -51.98305327428741   -52.02410486827709   -52.03057399973364
S.-52.746842639593915   -51.99726867166399   -51.24573335852383   -51.07689619014331   -51.36447850435806   - ↵
  51.77370368664302   -52.052739026879884   -52.180107189099715
S.-49.14513422818792   -48.393258426966305   -48.59758771929825   -49.523725834797894   -50.63584183673469   ↵
  -51.61326442721791   -52.28159340659341   -52.62937776266576
S.-46.82623786771796   -45.22293601875163   -46.17944016273672   -48.16517749335909   -50.11189938729442   -5 ↵
  1.771287676789875   -52.95163961576681   -53.511082892453246
S.-47.302659019812296   -43.0   -45.107049072706374   -47.61963095053497   -50.032745715974   -52.27863286132 ↵
  721   -53.92149020312781   -54.54026122760614
S.-51.0   -47.27163854109961   -46.58138617765947   -48.11505791505791   -50.308240081383516   -52.6728447690 ↵
  0075   -54.43462952820594   -55.0
```

Measurement at this point

All measurement history (measured map)

Estimation using the algorithm (Interpolated map)

(a) Experimental Log

(b) Actual Map

Figure 4.8. Wi-Fi Map Actual Results

---

[7] This map is taken in a diffrenet experiment to the log discussed above.

Chapter 5

Conclusion and Future Work

In this project, we have successfully developed an Android application that measures the floor plan of a house with an overlaid color-coded Wi-Fi strength map. Being the first application in the market that capable of drawing indoor Wi-Fi strength map, our application provides a more economic and efficient method for householders and Internet Service Providers to measure and diagnose the Wi-Fi coverage problems, which may further help them to optimally place the routers. The application may also be useful for business place (e.g. shopping malls and restaurants) owners to provide their visitors with information of where to have a better Wi-Fi access by providing the Wi-Fi map.

Technology wise, I have addressed the two challenging problems of measuring the floor plan and estimate the Wi-Fi map. Both the Inertial Navigation Approach and the Camera Approach were investigated. The Inertial Navigation Approach was finally abandoned due to the poor performance of the accelerometer in cellphones. The Camera Approach was finally chosen. Together with the calibration feature, I am able to keep the error rate very low and achieve a satisfactory performance. Lastly, a quadratic interpolation algorithm was implemented to estimate the Wi-Fi map from finite measurements.

In terms of future work, one may further investigate the interpolation algorithm and probably come up with a more accurate estimation approach to layout the Wi-Fi map. In addition, the current Camera Approach in measuring floor plan doesn't allow any curvature boundary of a room. One may develop a new algorithm that will capture an arbitrary shaped rooms. Lastly, one may also connect this application to cloud servers so that users can get help from others in optimally placing their routers by uploading the resulting maps. Also, user may download the Wi-Fi map of a public place before visiting there.

References

[1] M. Afanasyev, T. Chen, G. M. Voelker, and A. C. Snoeren, "Usage patterns in an urban wifi network," *Networking, IEEE/ACM Transactions on*, vol. 18, no. 5, pp. 1359–1372, 2010.

[2] S. Perez, "With 58% market share, android will top ios in smartphone app downloads this year, but apple will win on tablets," http://techcrunch.com/2013/03/04/abi-with-58-market-share-android-will-top-ios-in-smartphone-app-downloads-this-year-but-apple-will-win-on-tablets/, Mar 2013.

[3] R. Nazarian, "Android continues to dominate with 53% of the market share; ios takes 50% of application revenues," May 2011, retrieved 12 30, 2012, from Talk Android.

[4] Asymco, "Are android app developers more prolific but less persistent than ios developers," http://www.androiddevelopmenttalk.com/android-apps/are-android-app-developers-more-prolific-but-less-persistent-than-ios-developers/, July 2011, retrieved 12 30, 2012, from Android Development Talk.

[5] W. von Braun, "Concluding remarks by dr. wernher von braun about mode selection for the lunar landing program." *Lunar Orbit Rendezvous File*, July 1962.

[6] E. Nebot, S. Sukkarieh, and H. Durrant-Whyte, "Inertial navigation aided with gps information," in *Mechatronics and Machine Vision in Practice, 1997. Proceedings., Fourth Annual Conference on.* IEEE, 1997, pp. 169–174.

[7] Y. Xiaoxia and H. Yi, "Systematic calibration method for the laser gyro strapdown inertial navigation system," in *Control Conference, 2007. CCC 2007. Chinese.* IEEE, 2007, pp. 478–483.

[8] "Magicplan & floorplanner partner up," http://ww.floorplanner.com/weblog/magic-plan-integration, Floorplanner, May 2012.

[9] K. R. Britting, *Inertial navigation systems analysis*, 2010.

[10] J. Carr, *RF components and circuits.* Newnes, 2002.

[11] http://en.wikipedia.org/wiki/DBm, April 2013, retrieved 4 10, 2013 from Wikipedia.

[12] J. Geier, "Beating signal loss in wlans," http://www.education.com/science-fair/article/wifi-signals/, July 2002.