Department of Computer Science and Engineering
The Chinese University of Hong Kong

LYU1204

A Mobile Assisted Localization Scheme for Augmented Reality

FYP Report Fall 2012

Supervised By

Prof. Michael R. Lyu

Written By:

Huen Shiu Fung 1155002688

Tsang Sze Hong 1155000577

Abstract

Today mobile devices such as smartphones and tablet computers are having great computational power, equipped with advance measurement sensor such as magnetometer and accelerometer to capture the environment around. Mobile devices are much more than for dialing and receiving calls.

In this project, we try to utilize the sensors of mobile device to measure the indoor distances and aspect ratio to generate the floor plan of the indoor. Then having the floor plan generated, the device can find position on the floor plan by using the camera and other measuring meters. On top of this localization method, we can extend to build augmented reality applications using this localization method.

We first study on the relevant topic about each of the above step. The product would be augmented reality application using this localization method running in Android platform.

Please read Chapter 1 for brief introduction about this report.

Table of content

1. Introduction

In chapter 1 of this report, we will talk about the background and insight that brought us the project idea. **Section 1.3 will be a brief overview of the project**. In chapter 2, we will come across the relevant research to our project we have studied. In chapter 3, we give you the overview of our project design. **Chapter 4 will be our detailed project design.** In chapter 5, we will show you the testing and comparing our current work completed. Chapter 6 will conclude what we have done and what is our future work.

1.1 Motivation

Smartphones, tablet computers these mobile devices are getting greater importance to our daily lives. They are now equipped with advanced measurement sensors like magnetometer, accelerometer, gyroscope and compass etc. All of these help to measure and detect the status of the device and environment around easily

Moreover, Global Positioning System(GPS) are also incorporated into the mobile devices. Receiving signal from the satellite in the space and obtain the device's current location information. It is useful for navigation use. GPS is advance and accurate enough for user widely use it as navigation. However, GPS is restricted to outdoor usage

Fig 1.1.2 Satellite used for GPS

Fig 1.1.2 GPS for navigation

only. Indoor area is difficult to receive the satellite signals and cannot be use for indoor navigation purposes. But with assisted GPS(A-GPS), mobiles phone uses network resources for location and capable for indoor positioning and navigation. Google maps 6.0 started to have indoor mapping and navigation function. Although, indoor positioning and navigation technology is evolving, the covering is still very small. Nokia indoor navigation system “Destination maps” included 4605 buildings. We can see that technology for indoor environment have still a way for improvement. So we want to do something about indoor.

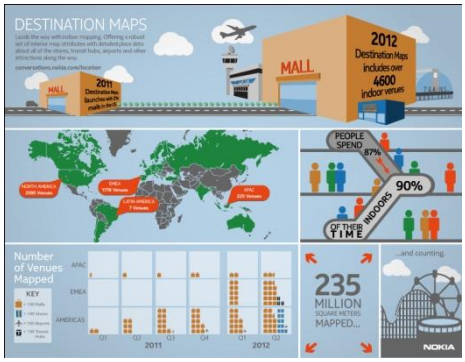


Fig 1.1.3 Nokia Destination Map

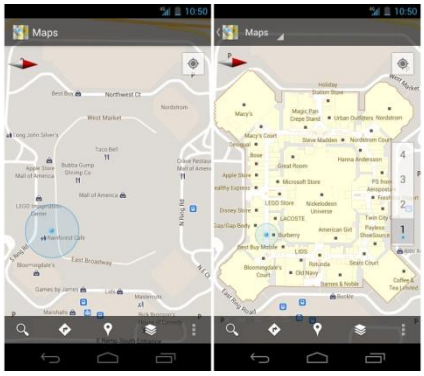


Fig 1.1.4Google Maps 6.0 with indoor positioning

Mobile devices are also having great computational power. Newer generation even have multi-core processor, gigabytes of memory which can already compete with personal computer. It is easy for these devices to render 3D scene. Many complex 3D games with beautiful graphics can be run on these devices.

Fig 1.1.5 GTA on Android Phone

From the above observation, mobile devices are definitely more than for making call and manipulating documents. Thus we want to first utilize the sensors of our mobile device to capture the certain degree of indoor environment and recognize the current position as a localization scheme. On top of this, use the great computational power of mobile devices to build an augmented reality application. This is how our idea comes from.

1.2 Background

There are some technology and give us insights to further extend our idea to become the project work. We have looked into some software and applications in the market. The followings would be a brief description about those software that brought us insight.

1.2.1 IKEA Home planner

IKEA Home planner is a 3D tools developed by the furniture company IKEA, running on PC or Mac computer browsers. This is a tool for costumers to design their home with new furniture put on virtually. It provides 2D floor plan view and 3D drawer functions for users. In the floor view, users can add actually measurement of features in the home such as wall length, door to draw the update the floor plan. The 3D view function gives costumer a more realistic plan for the home. User can draw and build a 3D view of the home and place the product of the company on it. The software finally records the product information and prices for costumer as reference.

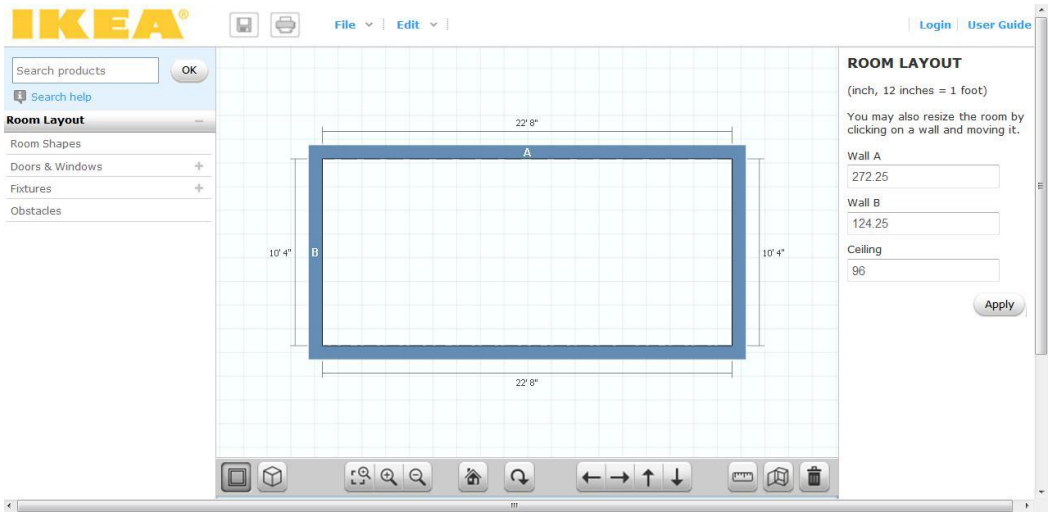


Figure 1.2.1.1 IKEA home planner 2D view needed to enter wall length

This is a quite useful software for users who move to new home but we still see some inconvenience in this software. First, it is not portable as it is on computer and the installation is not trivial. Having a version on mobile device would be more convenient. Also, user needs to enter every length to the program to generate the view which is rather troublesome. So if somehow we can get the ratio of different lengths automatically by using the mobile device to capture scenes from the indoor environment and enter one length for calibration, it would be much more convenient.

Fig 1.2.1.2 3D room planning on IKEA home planner

1.2.2 MagicPlan

MagicPlan is an iOS application for ipad on itunes. Its can measure user's room and build a floor plan by the using camera to obtain the indoor scene. Floor plan can be generated in different format such as PDF or JPG. It can also publish an interactive floor plan on web to let you build home virtual tour to walk around in your home. MagicPlan build floor plan by first starting the camera and let user to mark each corners in the home. When all corners are marked, the floor plan can be generated.

Fig 1.2.2.1MagicPlan

MagicPlan is a user friendly application which can be used without much limitation. It's user interface is easy to use. The measurement and floor plan generation function gives us insights on our project development. However, it is a commercial product but not an open source project, we cannot look at the implementation. We would use it as a comparison agent to our product. Also, it is an iOS application only and doesn't have an Android version.

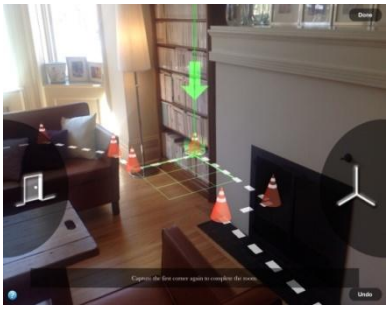


Fig 1.2.2.1MagicPlan Screenshot

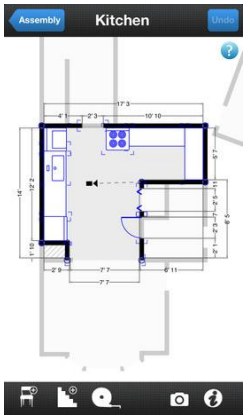


Fig 1.2.2.1MagicPlan Floor Plan Result

1.2.3 Floor Plan Creator

When it comes to Android platform, there is also an app named Floor Plan Creator which allow user to build a floor plan interactively. Users can choose to draw the floor plan on the blank page or interactively build it by marking the corner of the room like the MagicPlan.

Fig 1.2.3.1 Floor Plan Creator

However, this one is much more difficult to use than MagicPlan for iPad. It is not easy to creator a floor plan successfully and the app is always stuck on some point and cannot move on. Clearly the iOS version for floor plan creating application is more advanced and nicely built. Thus this would be part of the reason we choose Android as our development platform.

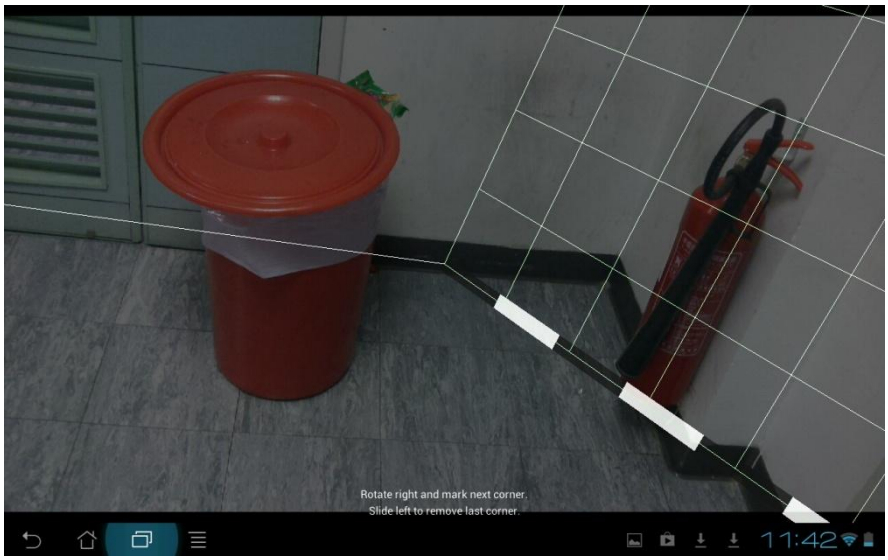


Fig 1.2.3.2 Floor Plan Creator stuck on the screen

1.2.4 Smart tools

Fig 1.2.3.1 Smart Tools

Besides floor plan generating applications, we have also looked into some apps for length measurement. Smart tools in popular in Google Play store with high rank and download times. It is a tool for different measurement as a virtual ruler or protractor. However, the mechanism is just printing markers on the screen like a real ruler. It is not

interactively measure the length. It do possess a distance measurement function interactively measure the distance from a object with the input of the height of the object. This is done by some geometric method which would be discussed in later section. We think that interactively measuring length and distance would be a convenient and interesting function.

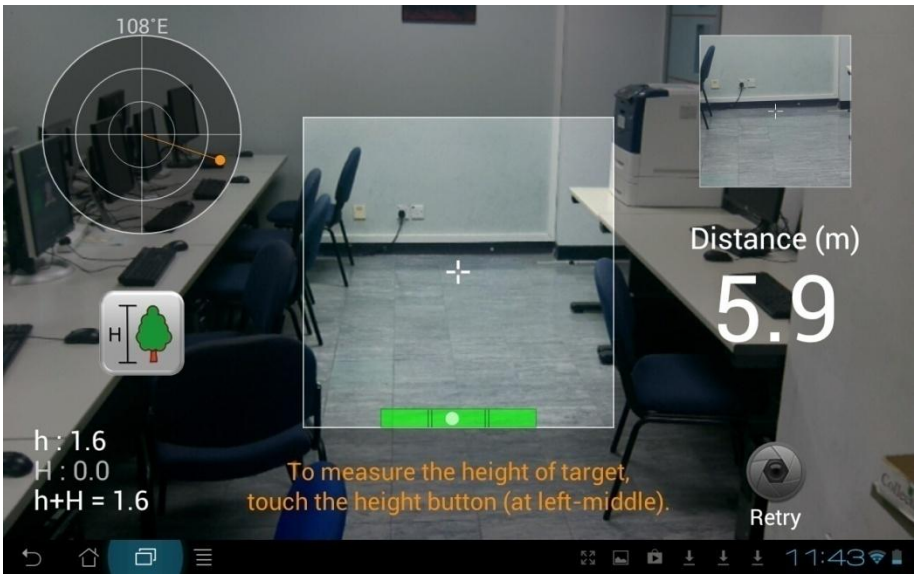


Fig 1.2.4.2 distance measurement function of smart tools

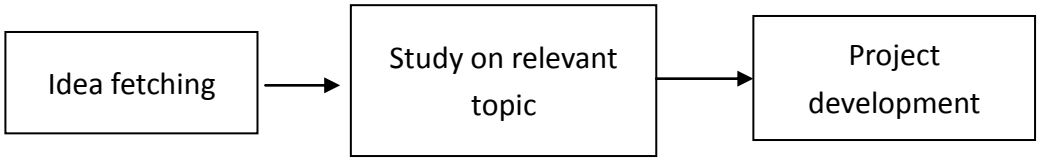
1.2.5 Advanced Ruler Pro

Advanced Ruler Pro also uses camera as a interactive ruler like the distance measurement function of smart tools. Basically it start with calibrating one of the length of a know object, which is also similar to the procedure of smart tools. The control of this app is quite complicated so there is some room for improvement.

Fig 1.2.5.1 Advanced Ruler Pro

1.3 Project overview

This section will give a report on work done and to-be done for our project. From the previous sections, we have talked about the insights and background. After we have studied some of the topic relevant to our brief idea, we can start our project development. Studied topic will be discussed on chapter 2.



Our project is to develop an Android application. The function design can be basically divided into core part and extension part.

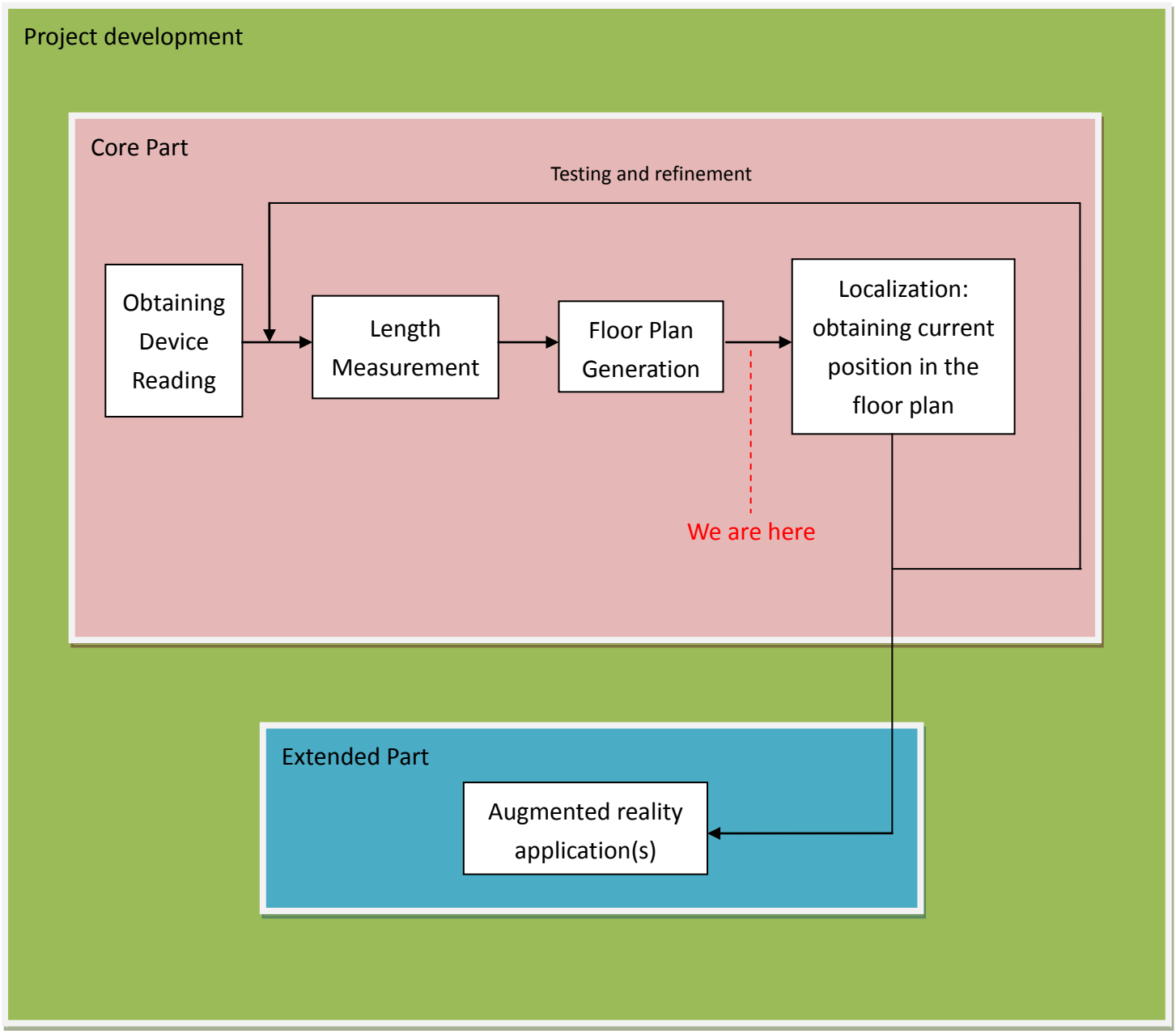
The core part consists of four functions. First, we obtain the device compass reading for obtaining device orientation relative to the environment. We need to get familiar with the development platform and API to be used in this stage. Second we would use the reading obtained to measure length of two points, with some estimation and assumption added for calculation. Third part would be our floor plan generation. We use the reading obtained to generate the aspect ratio of the floor plan and use the length calculated for scaling. In this part, the floor plan generated will have certain restriction because the algorithm used is based on some assumptions. Also we can mark the current position on the floor plan.

Our implementation progress of this semester is up to here due to the time constraint.

The forth part is the localization part, with the floor plan generated before, when user are at certain position of the room, using the camera and compass to recognize the current position of the floor plan. These are the function of the core part. The core part can be utilized to build different application

The extension part is to make use of the localization method for building augmented reality application. Start from simple, when the floor plan is generated, transform 2D floor plan to 3D room scene and allow user to put some 3D object to the room. This is one of the applications to be created.

Another idea is to build an interactive game using the core part. Say first transform the 2D floor plan to 3D room and put a virtual basketball stands in a position. A user in arbitrary position of the floor can shoot a ball to the basket.


Project development
Core Part
Testing and refinement
Obtaining Device Reading
Length Measurement
Floor Plan Generation
Localization: obtaining current position in the floor plan
We are here
Extended Part
Augmented reality application(s)

2. Study on relevant topic

We will explain briefly the material we have studied about when working on the project briefly.

2.1 Length measurement

We have studied on how to measure the length of an object in some approaches.

2.1.1 From prospective geometry to single view metrology

One way to do length measurement is using prospective geometry. It uses homogeneous coordinates for easier representation of homogenous coordinates. Using the parallel lines can define vanishing lines. Using a know height of object measure the height or length of another object through a know ratio.

Single view metrology is an algorithm summaries previous projective geometry study on measurement. From a view like a photo, it first recognizes some reference lines as axis of perspective and find out the vanishing plane points. Then enter a height or length of a reference object in the photo can obtain the length of the other object in the photo by some mathematics.

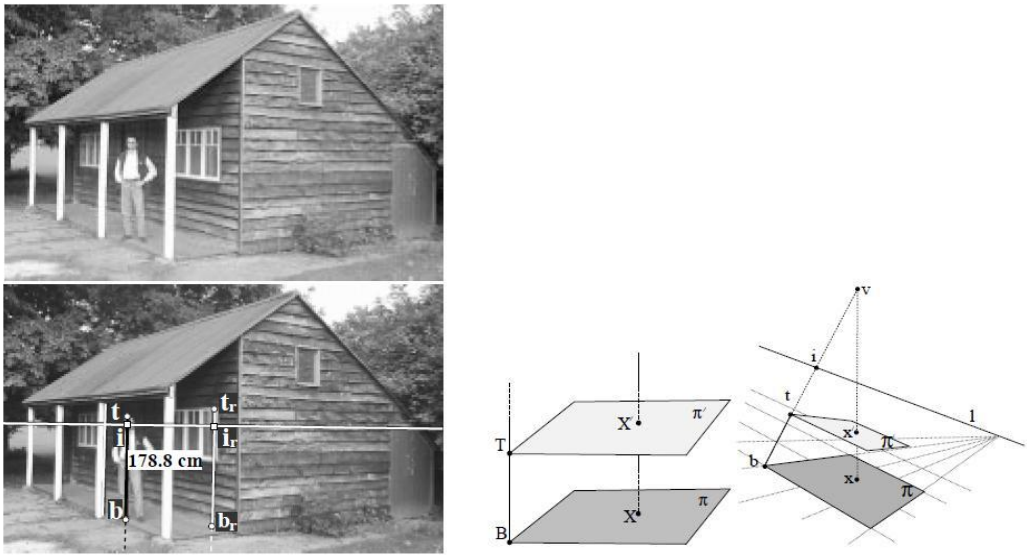


Fig 2.1.1.1 Single View Metrology and perspective geometry

Although the idea behind is not difficult to understand, the mathematics and implementations of this topic is quite complicated. Thus we choose not to implement our length measurement function using Single view metrology.

2.1.2 Smarts tools

In chapter 1, we have introduced the measurement tools “smart tools” on Google Play store. It has a distance measurement function. This app has given some diagram about how is works for us to interpret and understand the background mechanism of this function.

Computing Distance

With reference to fig 2.1.2.1.In order to compute the distance from a target, the system need to have the angle between the ray towards the target and the height of the device from the ground. The height is needed for the user to input. And the target distance can be computed by simple geometry:

$$AB = A'A * \tan(\Theta)$$

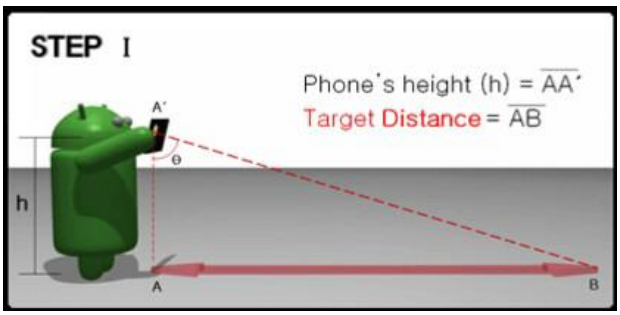


Fig 2.1.2.1 Diagram in smart tools for distance measurement

If user are standing inside a building and aiming at target out of the building, smart tools allow user to make an adjustment by inputting the height of the building.

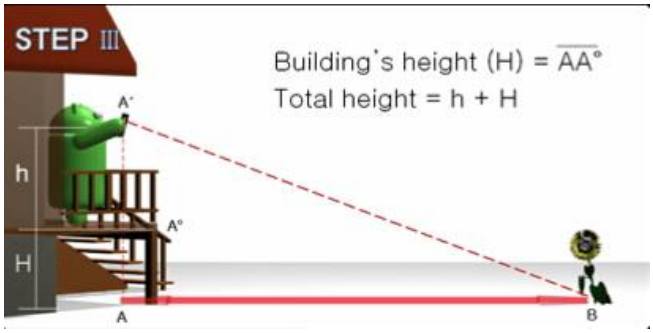


Fig 2.1.2.2 Diagram in smart tools for distance outside a building

Computing Height

Also by geometric method, the height of a target can be measured by getting the device angle of elevation and height of device from the ground which is inputted by user. With reference to fig 2.1.2.3

$$BB' = A'A + B'C$$

And $$B'C = \tan(\Theta) * AB$$

where AB could be obtained the step discussed previously.

$$\text{Thus, } BB' = A'A + AB * \tan(\Theta)$$

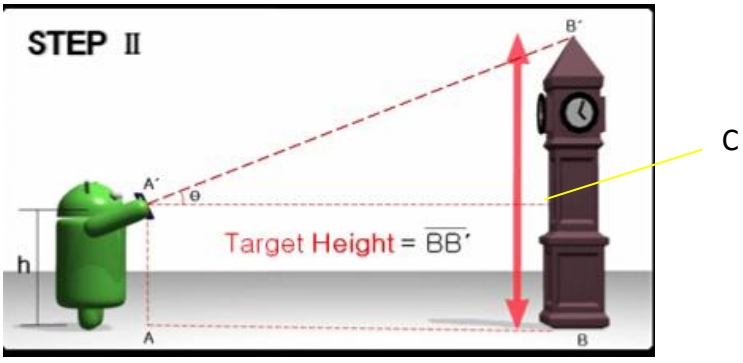


Fig 2.1.2.3 Diagram in smart tools for height measurement

Computing width

If user rotates along a vertical axis, the angle of rotation can be used to compute the width of an object. With the distance from the user to the left

point and the right point, the width of the object can be computed by cosine law:

$$BC^2 = AB^2 + AC^2 - 2*AB*AC*\cos(\Theta)$$

$$BC = \sqrt{(AB^2 + AC^2 - 2*AB*AC*\cos(\Theta))}$$

Where AB and AC can be measured by previous steps.

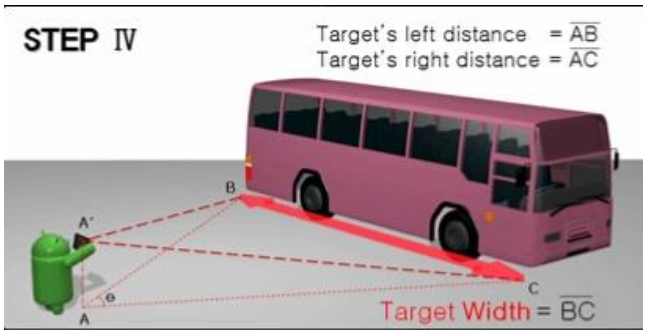


Fig 2.1.2.4 Diagram in smart tools for width measurement

Thus it can be seen that the distance measurement function of Smart tools works by geometric method.

2.2 Rendering Indoor Scene

Aditya Sankar from the University of has released a research paper on the topic “Capturing indoor scenes with Smartphones”. This is about a research on building smartphone application to capture, visualize and reconstruct indoor area. The application first obtains data from smartphone camera, accelerometer, gyroscope and magnetometer to visualize the indoor environment. Then the application can output a real time visual tour for user to explore the transaction of the room.

Interactive tour

User rotates 360 degree to capture a video of the indoor environment. When the capture is completed, the system generates a 360 degree panorama. The interactive play back of room tour is done by indexing the video by camera pose which is obtained from the reading of gyroscope sensor in the smartphone. The instant orientation of the smartphone obtained from the equation:

$$\Theta_t = \Theta_{t-1} + t * w$$

Where t is the change in time and Θ_{t-1} is the previous orientation and t is the change in time and w is the change in orientation. All in all real time interactive tour is done by spatial video indexing.

Floor Plan generation

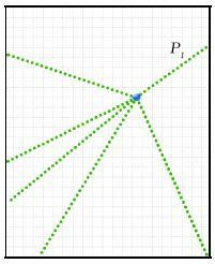
In playing back the interactive room tour, the system allow user to mark the corner. When the marking is completed, the system uses the angle of

orientation of the marker to generate the floor plan. For 2D floor plan generation, it has given the following algorithm:

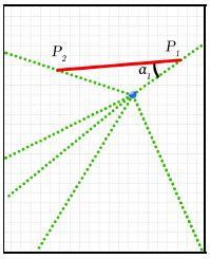
Algorithm 1 Calculating optimal wall configuration

P_0 = Origin
P_1 = Unit distance from P_0 along the first ray
N = Number of corners
$minDistance = MAX\_FLOAT$
for $\alpha = 0 \rightarrow 360$ step 0.5 **do**
    for $i = 2 \rightarrow N$ **do**
        θ_i = direction of i^{th} corner ▷ from (2)
        $P_i \leftarrow intersect(P_{i-1},\ \alpha + (i-1) \times 90,\ P_0,\ \theta_i)$
    end for
    $distance = getDistance(P_1, P_N)$
    if $distance < minDistance$ **then**
        $minDistance \leftarrow distance$
        $minAngle \leftarrow angle$
    end if
end for
Return $minAngle$

Using the marker orientation can obtain a series of angle, we can regard it as shooting different rays from the capturing position.



Choose an arbitrary angle alpha and start with first ray, connect a line with angle alpha to the first ray.



Then the following rays are intersected by making a 90 degree with the previous ray.

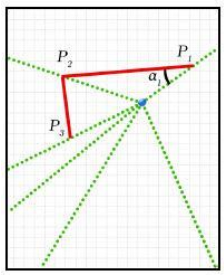


Fig 2.2.1 Steps of Sankar Algorithm

When the last ray meets with the first, the intersection point should be at

the starting point if the angle is chosen alpha is correct. However, in practical the two points do not intersect exactly. So the algorithm tries different values of alpha and chooses the alpha which result in having minimum distance between the last point and the starting point. If the distance d in figure 2.2.5 approaches zero, the first point P1 and the last point P7 merges.

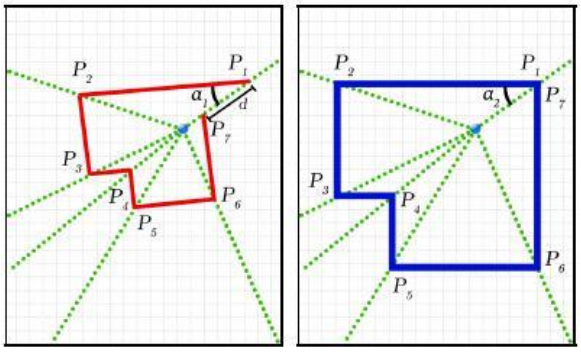


Fig 2.2.2 Steps of Sankar Algorithm

Furthermore, this algorithm is based on assumption that every corner is of degree 90. It is not applicable to the rooms with corners not of degree 90.

Aligning different rooms

After different rooms' floor plan is generated, the rooms are aligned to make the whole floor plan of the home. Floor plans of two different rooms can be connected with the supply of a door ray. Since the door ray of two different rooms are the same, so the two floor plans can be aligned by first multiplying the scaling matrix and then add a translation matrix to the coordinates.

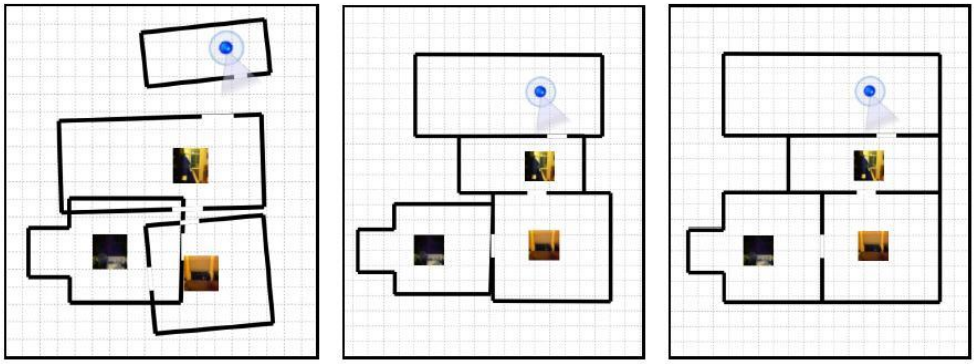


Fig 2.2.3Aligning different Floor Plans

Considering the complexity of the project and real time interactive visual video tour as it could require a lot of resources, we do not focus on the visual tour. Modeling indoor environment is only one of the tasks of our project. And the floor plan generating algorithm is useful to our floor plan generation function design. We have taken it as a reference to construct our algorithm. We first consider floor plan generation for a single room as our minimum target. If we have time to come back, we will try to include the function of aligning the floor plans of more than one room.

2.3 Augmented Reality

Augmented Reality is a view of the physical world with elements which is generated by computer. Most AR application can be are marker based. Usually are using the camera to analyze marker recognized or captured. Marker detection involves first converting the image obtained into binary image and then is to compute some points for recognition, called featured points. The last step is identifying the featured points.

Non-marker based AR application are those uses some features such as GPS to display some information augmented by the computer. There are many mobile AR applications and game available in the market.

Fig 2.3.1 AR shooting game on Google play store

3. Project setup

3.1 Target platform

The target platform of our project is on Android platform for tablet or smartphones.

Platform requirement:

OS	ANDROID version 2.3 or above
Hardware	(1) Sensors : Compass , Accelerometer , Gyroscope (Optional) (2) Camera
Device	Tablet or Mobile Phone

3.2 Development environment

We uses eclipse IDE with Android SDK plug-in as our development environment. Detailed version of the software used can be found in appendix. Implementation is in java in which we are quite familiar with. Of course we still need to read the API as some Android API are different with ordinary Android API.

3.3 Device used

Our development device is ASUS Eee Pad Transformer Prime TF201. The system currently runs smoothly without crashes on it. The application should also run in other Android devices with above requirement without problem. Transformer prime is a quad-core tablet with high computational power. Other weaker devices may run the application slower than Transformer prime. It gets slower when we have used opengl plug-in in our project. Generally, transformer prime computational power fits our program need in current stage.

4. Detailed design

In this section, we will explain our work in a detailed manner. Our project can be divided into a core part and an extension part. Section 4.1 to 4.6 is the core part and section 4.7 is about the extension part. **Section 1.3 gives a better overview of the work in brief idea.**

4.1 System Structure

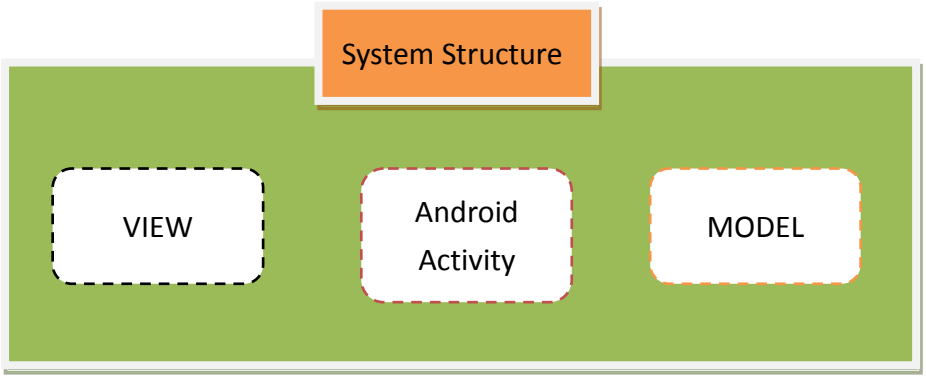


Fig 4.1.1 Structure of our system.

The structure of our system can be classified to three main categories, VIEW, Android activity and MODEL. VIEW contains what will be needed to display in the Android Activity. Activity communicates with view and model. MODEL process the main function of our application.

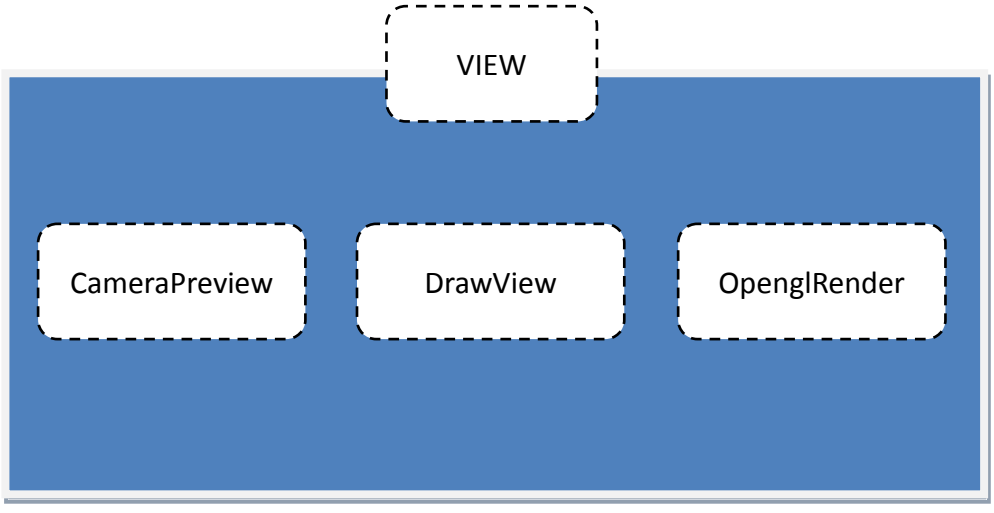


Fig 4.1.2 classes contained in VIEW

In each of the categories contains several modules with specific functions.

In VIEW, there are 3 classes: CameraPreview, DrawView and OpenglRender. CameraPreview is to control the camera and load the camera view to the screen. DrawView is for drawing basic information on the user interface including red corner marker for helping user to mark the corner, and some control buttons. OpenglRender aims to display the floor plan with opengl plug-in to the Android project.

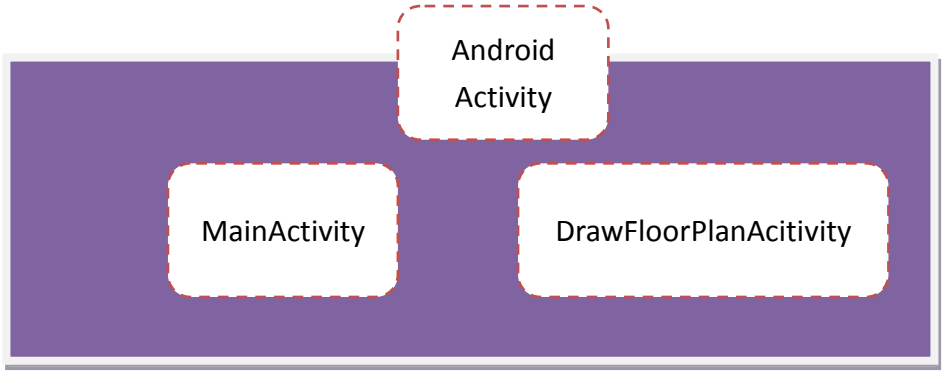


Fig 4.1.3 classes contained in Android Activity

In Android Activity, there are two modules: MainActivity and DrawFloorPlanActivity. MainActivity is loaded at the start of the program and load the user interface. DrawFloorPlanActivity is started when floor plan generation function is performed. It starts the openGL plug-in.

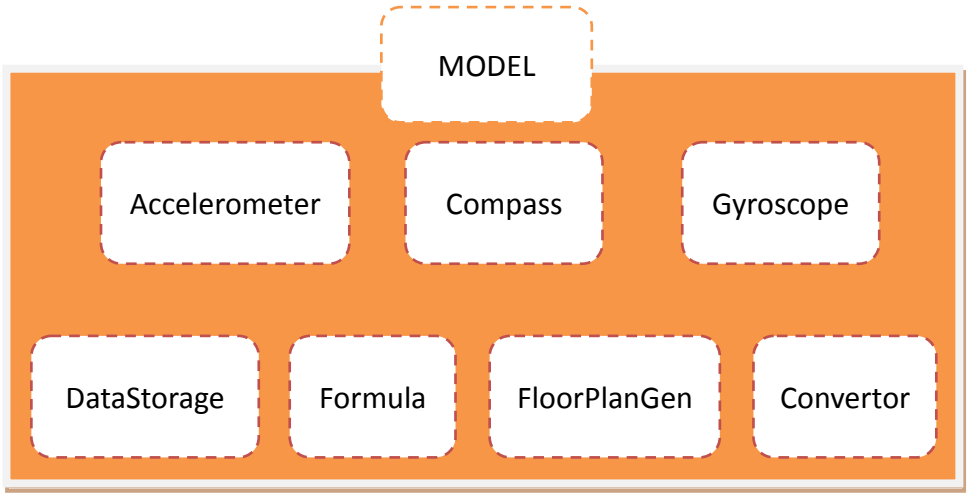


Fig 4.1.4 classes contained in MODEL

As seen in Fig 4.1.4, MODEL contains seven classes. Three of them are sensor class Accelerometer, Compass and Gyroscope. These classes obtain the reading from the sensors of the device. Data Storage module is for storing the data needed for communication between modules. Convertor module is for converting data for in the DataStorage for other classes to use. Formula module stores the method to measure length and FloorPlanGen module is the algorithm to generate the floor plan. They will be explained in section 4.4 and 4.5 respectively.

4.2 User Interface

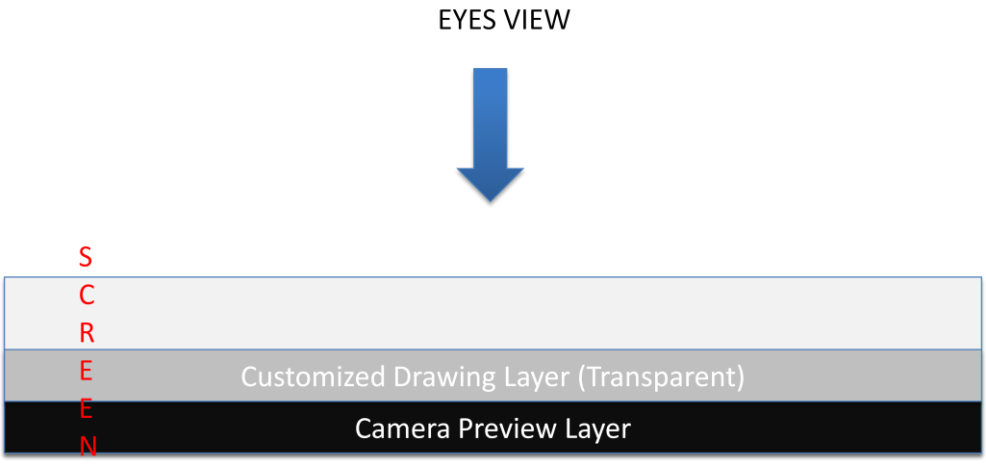


Fig 4.2.1 the user interface layer of our program

In our UI design, we have separated the camera and draw layers. The camera preview layer is in the bottom in which it displays the camera view instantly. On top of it, there is a customized drawing layer which allows us to display information and control on the UI. The red corner marker in the middle of the camera view is drawn for user to capture the corner easily. And control buttons are drawn for controlling the functions of the system. “Capture” button capture compass value, “GEN” button generates the floor plan from the data recorded. “CLEAR DATA” button clear previous data stored.

Fig 4.2.2 UI of our system

When the enough data is captured and the GEN is pressed, the UI switch to openGL to show the floor plan output. The colored square represents the floor plan of the room and the white cross is the position of the person standing when capturing the scene.

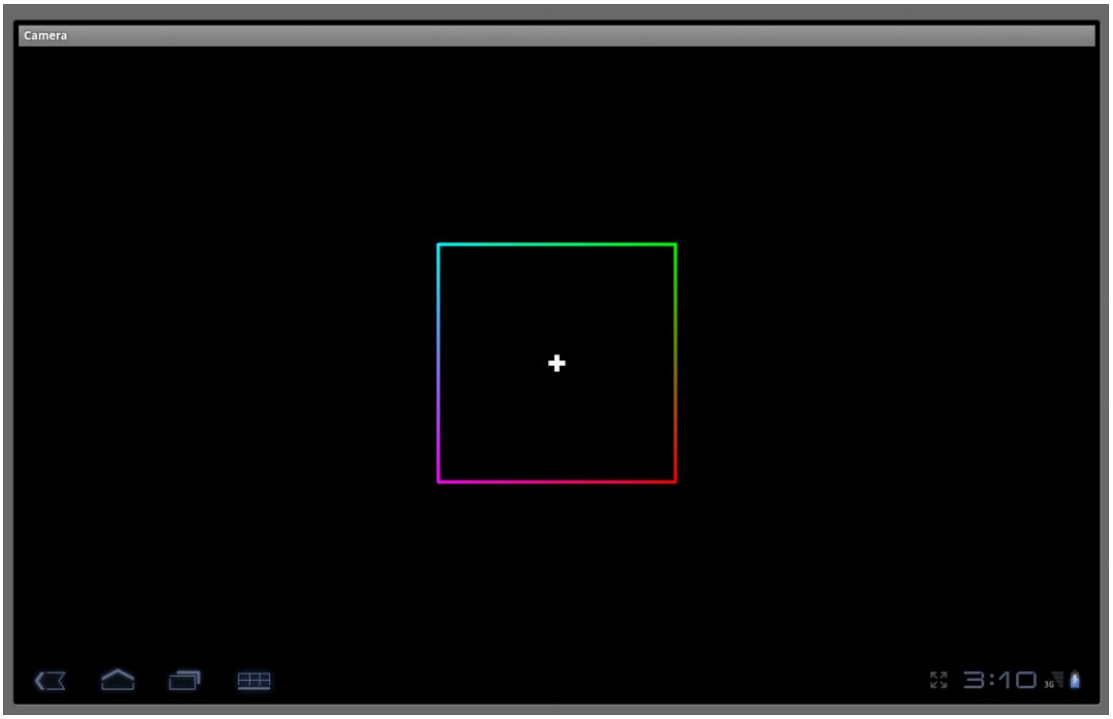


Fig 4.2.3Floor plan output UI

4.3 Obtaining pose

We need to obtain the device orientation before carrying out the measurement. Accelerometer class gets the acceleration of the device. The parameter is in 3 dimensions x, y, z. The compass class reading obtained is in azimuth, pitch and roll. The gyroscope class gets the value based on the principle of angular momentum.

Fig 4.3.1 Pitch, yaw and roll

At the current stage, we use mainly the compass reading. See figure 4.3.2 the value α can be obtained from the compass directly, it is called PITCH. Also, the value β can be obtained from the compass indirectly, it is called AZIMUTH. The readings are obtained using the getOrientation (float[] R , float[] result) function. Result[0] is azimuth and Result[1] is pitch and Result[2] is roll. They are useful in later on computation.

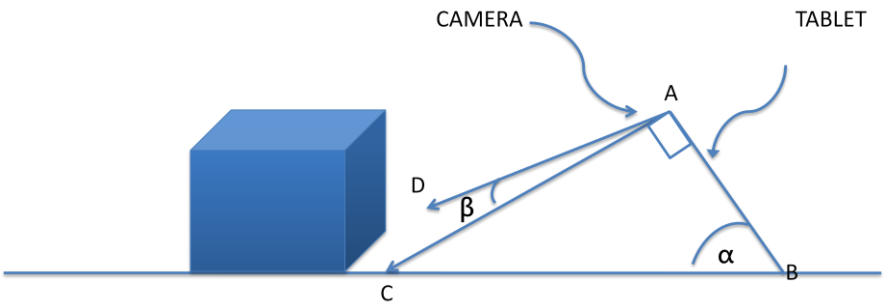


Fig 4.3.2 tablet orientation

4.4 Measurement

Next we measure the length between two corners in a room. User can use the red corner marker on the user interface to point at a corner of the room.

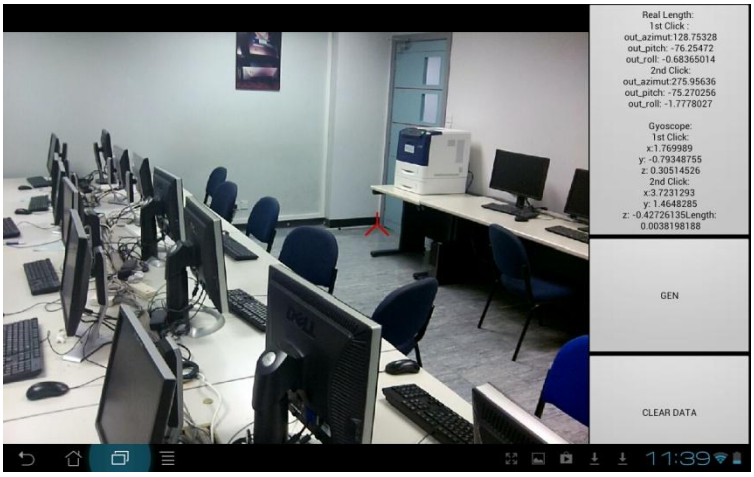


Fig 4.4.1 using red marker to capture the corner

When user touches on the capture button, the system will record the compass reading of the device. And the reading will be shown on the screen. Next, user uses the red corner marker to point on the next corner, and touches on the capture button again. Then the measured length will be shown on the screen at the upper right corner.

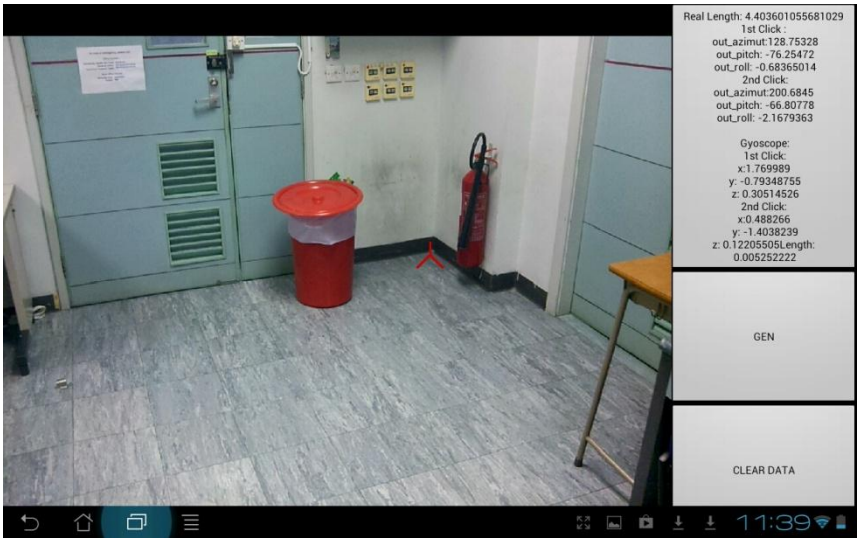


Fig 4.4.2 marking the second corner and the result of measurement is shown

We use geometric method to compute the length. The step is as follows.

Step 1:

When user capture the first corner, the orientation of the device is obtained. We record the angle$\theta$the device makes to the horizontal.

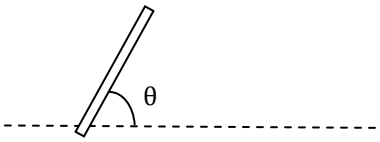


Fig 4.4.3Recording the angle that device makes with horizontal

Then we estimate the height of the device above the ground is as 1.4m. With this assumption we can formulate the vertical distance from the device to the target corner as follows:

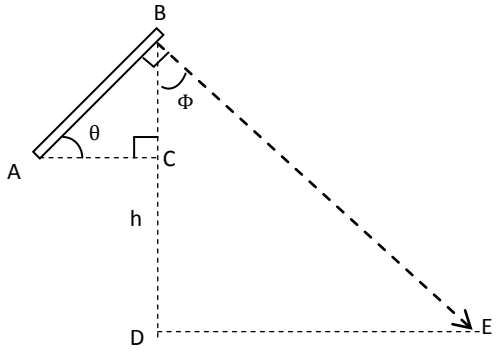


Fig 4.4.4 formulating distance from device to target corner

AB represents the device holding by the user and E represents the corner of the room the user pointed at.

$$\Phi = \angle ABE - \angle ABC$$

$$= 90^{o} - \angle ABC$$

$$= 90^{o} - (180^{o} - 90^{o} - \theta)$$ Since (Triangle ABC is right-angled)

Since θ have obtained already and h as estimated as 1.4, we can compute the distance from the corner as:

$$DE = h * \tan(\Phi)$$

$$= h * \tan(\theta)$$

Step 2:

When the user points at another angle can press capture, the system record another set of compass reading for to get the device orientation. Moreover, the angle β is recorded.

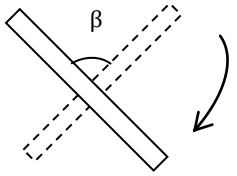


Fig 4.4.5 top view of rotating the device to point at another corner

With β the length the first captured corner to the second corner can be computed as follows:

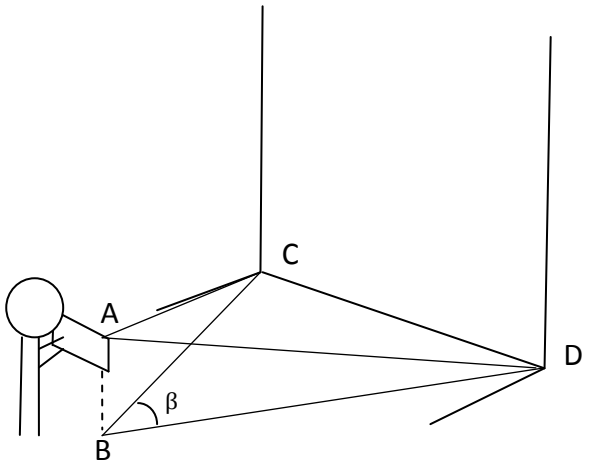
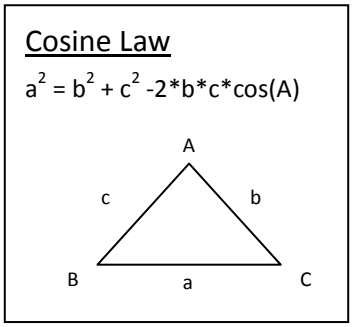


Fig 4.4.6measuring distance between two corners

The user wants to measure the length between the corner C and corner D of

his room. He first aim at C to capture the first corner and rotate to aim at D. The angle β is assumed to be the same as angle A. Then length between CD can be found:

$$CD = \sqrt{(BC^2 + BD^2 - 2*BC*BD*\cos(\beta))}$$



BC and BD is obtained with the method explained in step 1.

There is a minor assumption that the angle obtained is roughly equal to β since it should be the angle $\angle CAD$ instead of $\angle CBD$.

4.5 Floor Plan Generation

Next part is a floor plan generation function. User stands on a point inside a room and use the red corner marker on the user interface to target the corner of the room. Then the user presses the capture button to capture a corner. Then the user rotates clockwisely to target at the next corner to capture it. The steps are repeated until all the corners are captured. Then the floor plan can be shown on the screen by pressing the gen button. The output floor plan is drawn in openGL plug-in.

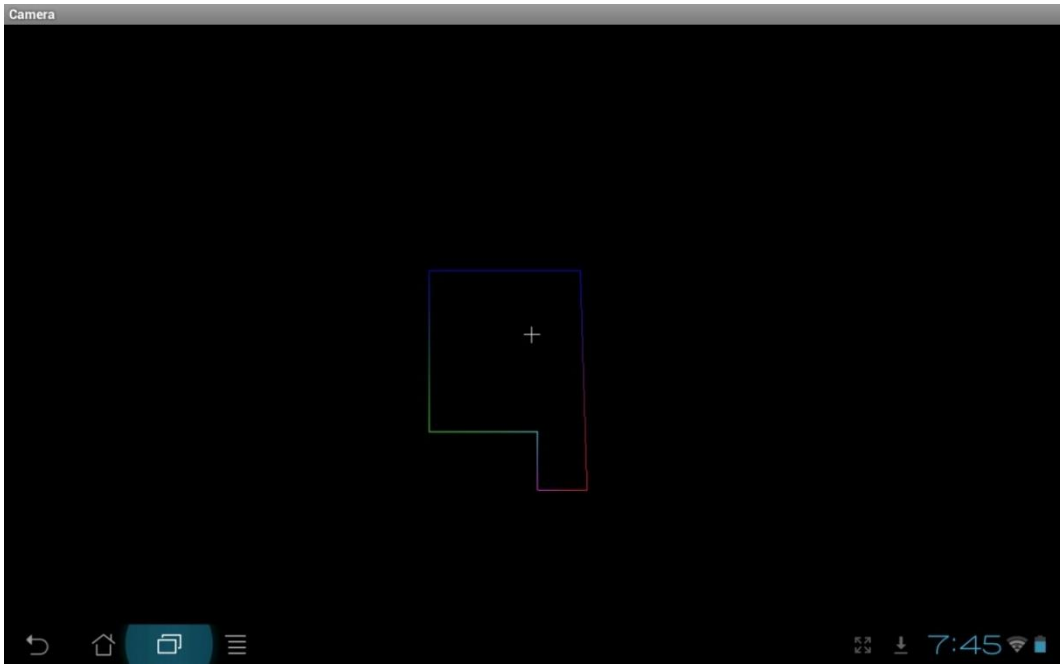


Fig 4.5.1 Floor Plan Display using openGL

We have designed our own algorithm for floor plan generation taken reference to the algorithm of Sankar on Chapter 2.2. The pseudo code of our algorithm is as follows:

Floor Plan Generation Algorithm Pseudo Code

```
minDist←minimum distance from the first to last point
angles[] ← input angle array
output[] ← output array storing the points
for(θ  = 0 → 360 step 0.1){
        sumAngle←θ
        P1X ←tan(θ)/√(1+tan²(θ))
        P1Y ← P1X* tan(θ)
        for(i=2 → no of corner+1 step 1){
                sumAngle← sum of previous angle in angles[]
                if(i%2 == 0){
                        PiX←P(i-1)X
                        PiY←PiX / tan(sumAngle)
                }else{
                        PiY←P(i-1)Y
                        PiX←PiY * tan(sumAngle)
                }
        }
        dist←calDistance (Pi+1, P1)
        if(dist<minDist){
                output←PiX, PiY
        }
}
```

The algorithm inputs an array of angle recorded and outputs an array of coordinates of the generated floor plan. The array is then pass to openGL rendering module.

Our algorithm starts from calculating the first point on the coordinate plane. The line connects from the original to the first point makes an arbitrary angle θwith the horizontal axis. The value of θ will be adjusted by iteration.

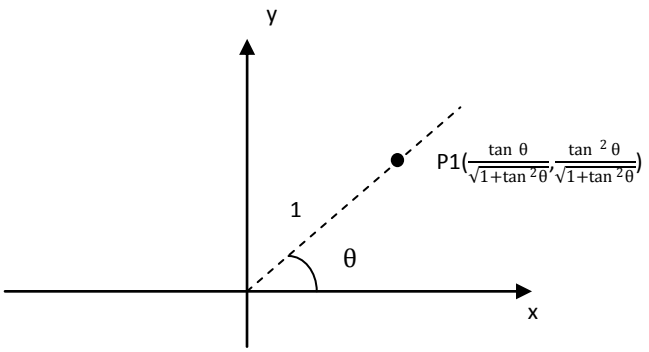


Fig 4.5.2the first point is computed from an arbitrary angle θ

Next add a ray which makes the angle from input array with the previous line. The next point is the intersection between the horizontal line passing through the previous point and the added ray. As seen in the following figure:

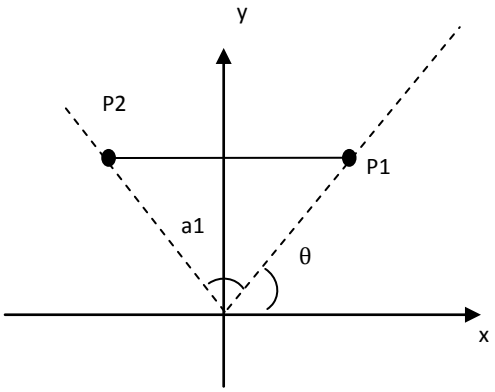


Fig 4.5.3 find the second point by intersecting

Repeat the steps by replacing the horizontal line and vertical line alternatively, i.e. if the ith iteration intersects a horizontal line with the new ray, then the i+1th iteration intersects a vertical line with the new ray. This assumes the angles of all corners in the room are of 90 degree.

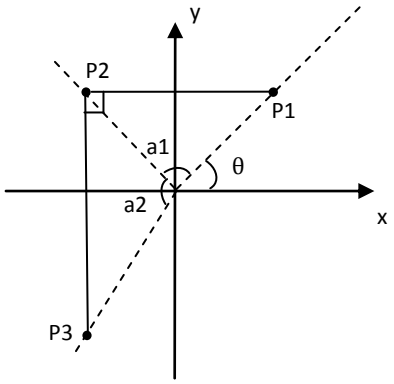


Fig 4.5.4Assume angle is 90^o to find the next point

Then the steps are repeated until the new added point is on the first ray.

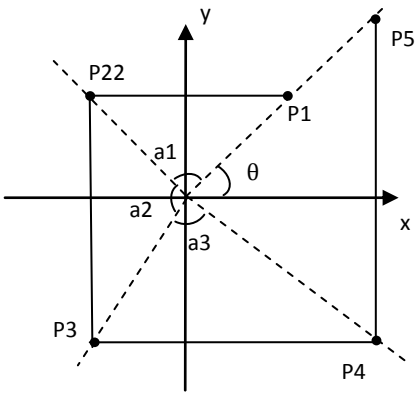


Fig 4.5.5 if unsuitable θ is chosen, the first and last point don't meet

Trying different values of θ in order to minimize the distance between the first and the last point so that they merge together. The floor plan with the minimum distance will outputted.

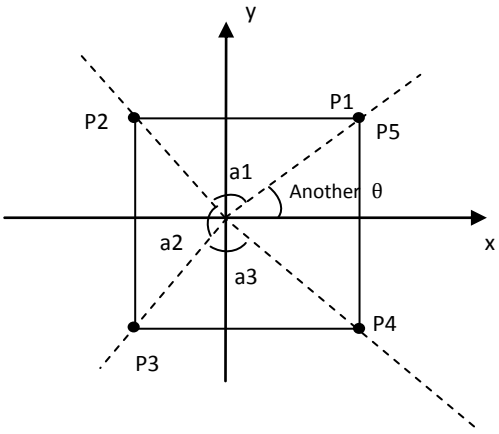


Fig 4.5.6With suitable values of θ, first point P1 and last point P5 merges

Of course this algorithm is also based on the assumption that angles of all corners in the room are of 90 degree. It is not applicable for rooms with corner of different angles. Another assumption is that users are capturing the corners inside the room. Testing on this algorithm is in chapter 5.

Our progress on implementation is till here. The next section is our design the other functions of the application.

4.6 Localization

Next we will do the last function of the core part. With the floor plan generated, user stand on a arbitrary point inside the room can know where he is on from the floor plan. The UI show the user where he is standing by the red cross shown on the floor plan.

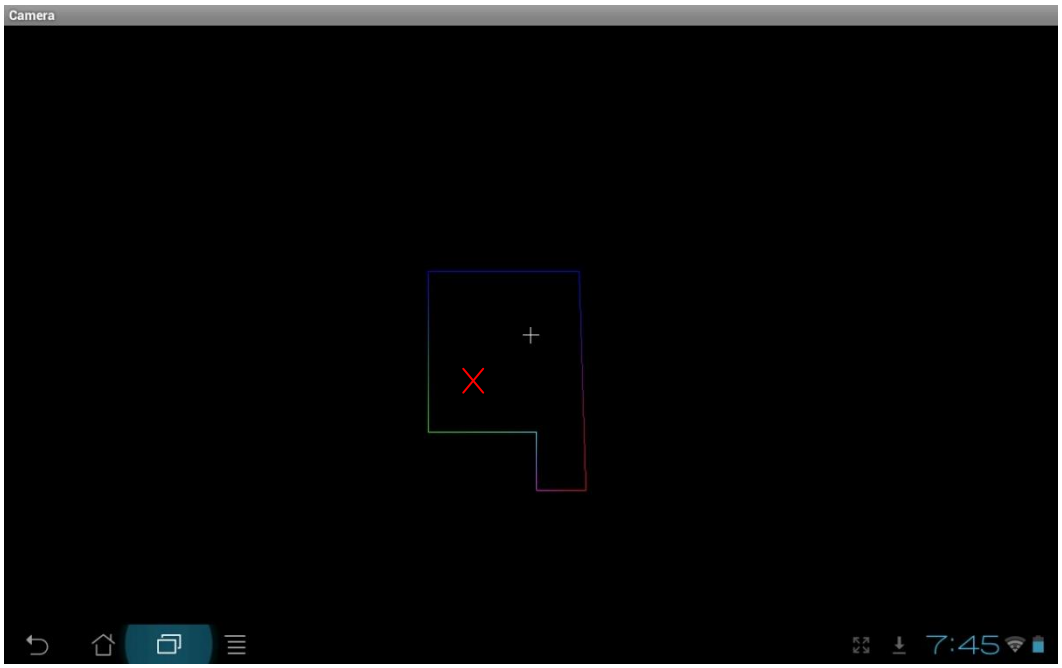


Fig 4.6.1 red cross telling current position

First we need to save our floor plan generated and open it again like file IO. This can be done by saving all the coordinates points of the corners in the floor plan and should not be a difficult task. Next we need to store all the compass reading for each corner during the capturing step of floor plan generation. We need to modify a bit the previous part in capturing corner of the room. Just associate and save the compass reading together with the coordinates of the corner of the floor plan.

When user wants to find his location, he faces at one corner, capturing a set of current compass reading. Then the new reading is compared with the set

associated with the floor plan. Then the corner that user is facing can be located. With the distance measured from the user to the corner, the position of the user on the floor plan can be located.

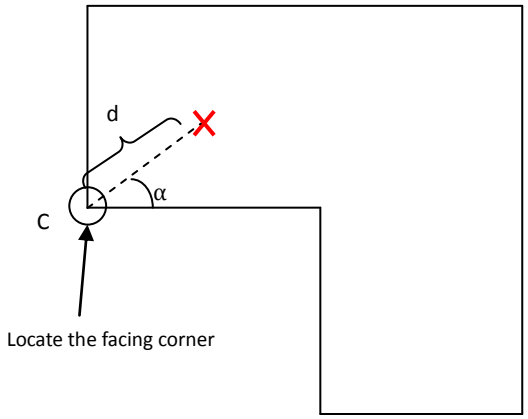


Fig 4.6.1 Concept diagram for localization

Look at Figure 4.6.1, when the user faces and capture a new corner. The system first recognizes it to be the corner C by comparing current compass reading and saved data. The orientation of the device towards the corner should also be found i.e. α is to be determined. With the measurement method in section 4.4, d can be determine. Thus, the location of the red cross marker. i.e. the position of the user, can be spotted out.

These functions contribute to our core part. With the core part, we can already capture a degree of actual indoor scene and have a localization scheme in which we can locate our position on the indoor floor plan we generated. With this core functions, we can build our application on top of that. This brought us to the extension part of this project.

4.7 Augmented Reality Application

This is an extension part of our project. We aim to build 2 applications using the core part. Here is the design of the 2 applications.

Start from easy, the first application would be an interactive room design application. With the 2D floor plan generated from the capturing the indoor room, we first transform it to a 3D room. And allow user to interactively add different 3D object into the 3D room to create a real room structure with computer generated decoration. Objective of this application is to let user to design their dream room with their creative mind.

Fig 4.7.1 3D room planning by opengl

The second application would utilize the core part to build a game. Using the localization scheme and floor plan generation, we turn the room to a basketball court by putting computer generated virtual basketball stand in the 3D transformed floor plan. Then, using the localization function, one can locate himself in the basketball court and notice the position of the

basketball stand. Then player a shoot a ball towards the basketball stand to score marks. This uses all functions in the core part. We also need to physics engine plug-in into the game in order to have real world physical behavior of basketball. We plan to use bullet-physics engine for android to complete the task.

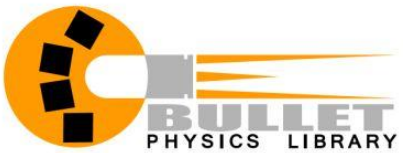


Fig 4.7.2. Bullet Physics Engine

Fig 4.7.3 Bullet Physics Game on Android

5. Experiments and testing

We have done various experiments to test our program and compare with other applications. The experimental data can be found in appendix.

5.1 Reading of sensors– Pitch

Introduction

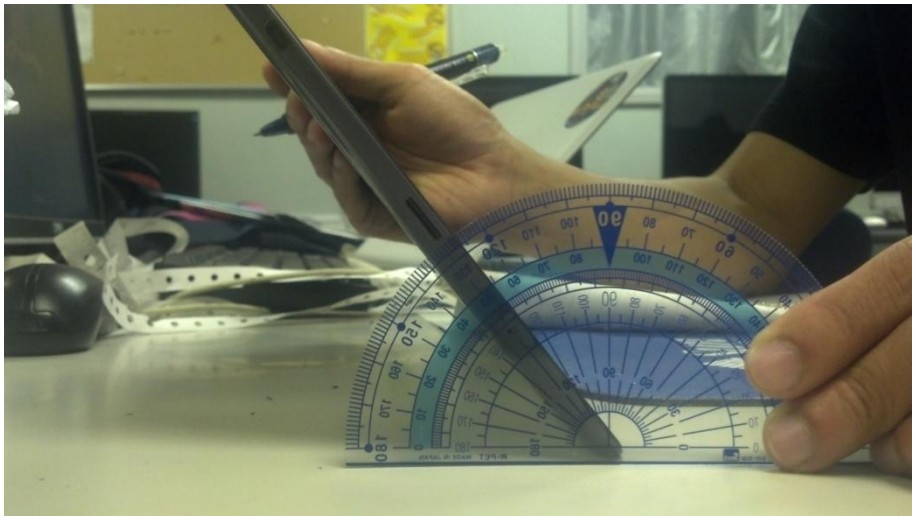
The Pitch of the device orientation is needed to compute the distance from the user to the corner of a room. We need to know whether our reading from device is accurate for further computation of distance.

Objective

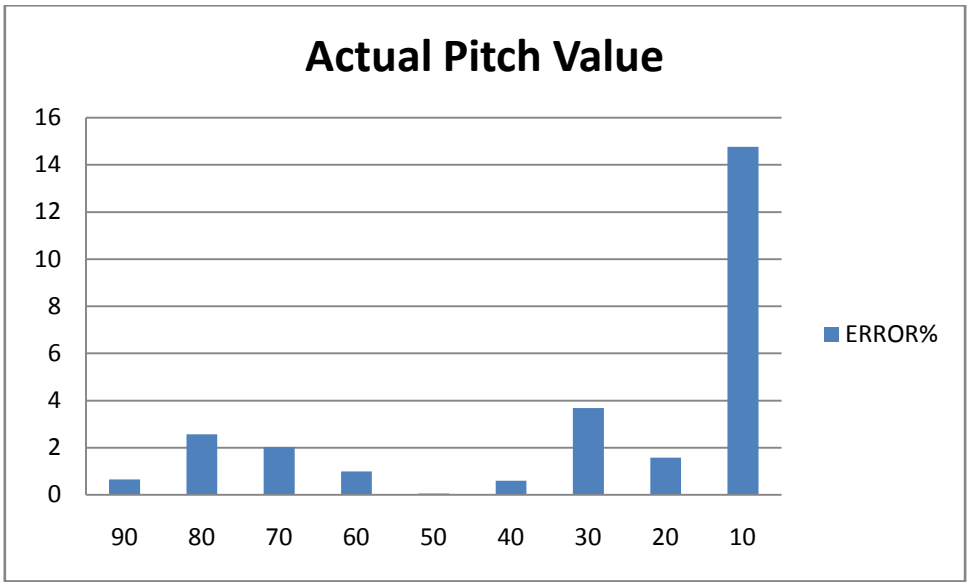
To understand how large error is of pitch angle of the device obtained from the compass reading.

Setup

Measure the pitch angle for step of 10 degree. Use protractor to measure the 10 degree and see the difference from our measurement function and the actual measurement by protractor.



Result



Conclusion

The error of the pitch angle record is remained quite small for angle larger than 10 degree. Normally we will not face a case with very small pitch angle orientation. Thus, the pitch from compass reading can be trusted for further computation on length measurement.

5.2 Reading of sensors –Azimuth

Introduction

Difference in Azimuth between two corners is needed to compute the length between two corners. We need to know whether our reading from device is accurate.

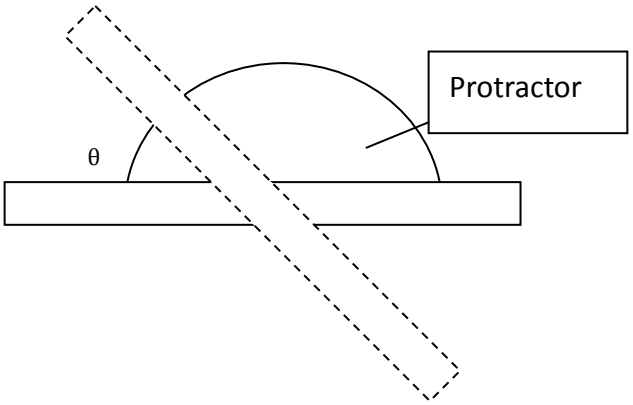
Objective

To find out the error of differences in azimuth of two device orientations. i.e. the angle of horizontal rotation when the user turns from pointing on one corner to another corner.

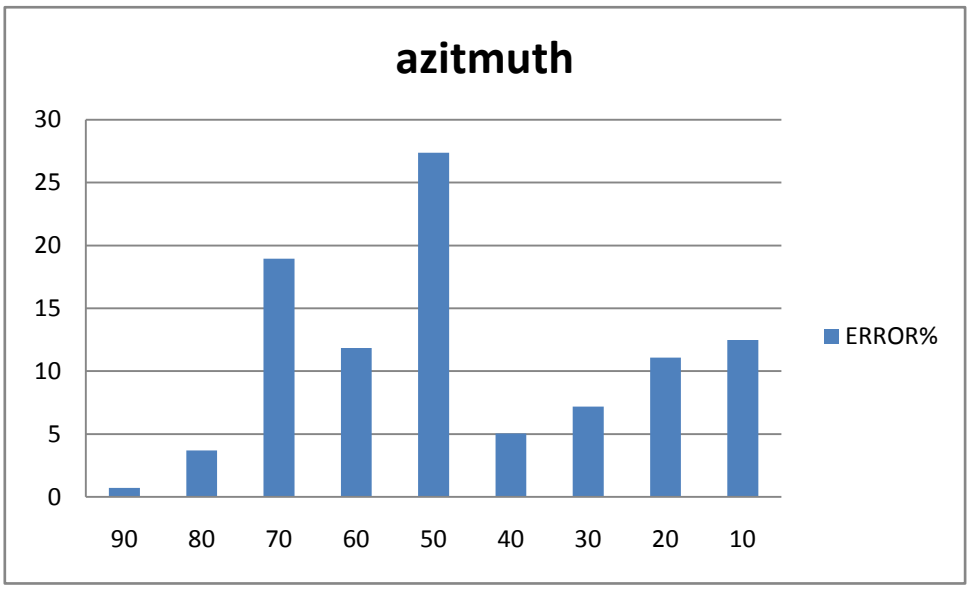
Setup

Place a protractor on table. Put the device vertical up right to make a 90 degree with the protractor. Capture one reading from the azimuth and rotate according to the marking on the protractor. Then capture another

Top view is :



Result



Conclusion

The difference in azimuth is quite large which may propagate and cause a large error for length measurement in later stage. And also the floor plan generation function maybe affected. There is a need to figure out a solution to obtain a more reliable reading or use alternative method to find out the horizontal rotation of the device.

5.3 Distance measurement

Introduction

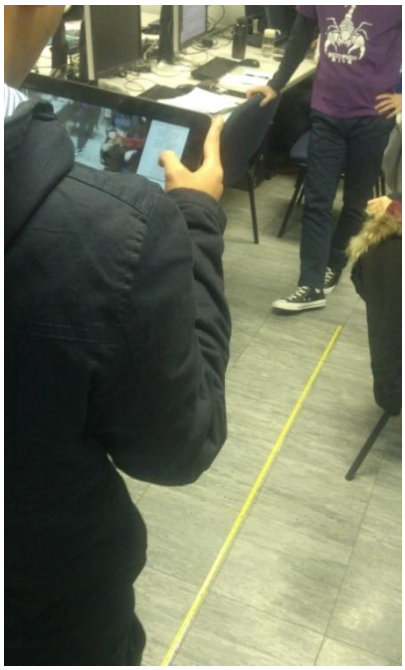
In our application, distance measurement function is an intermediate step of finding the length between the devices. We need to know whether this step is accurate for us to proceed to the length measurement.

Objective

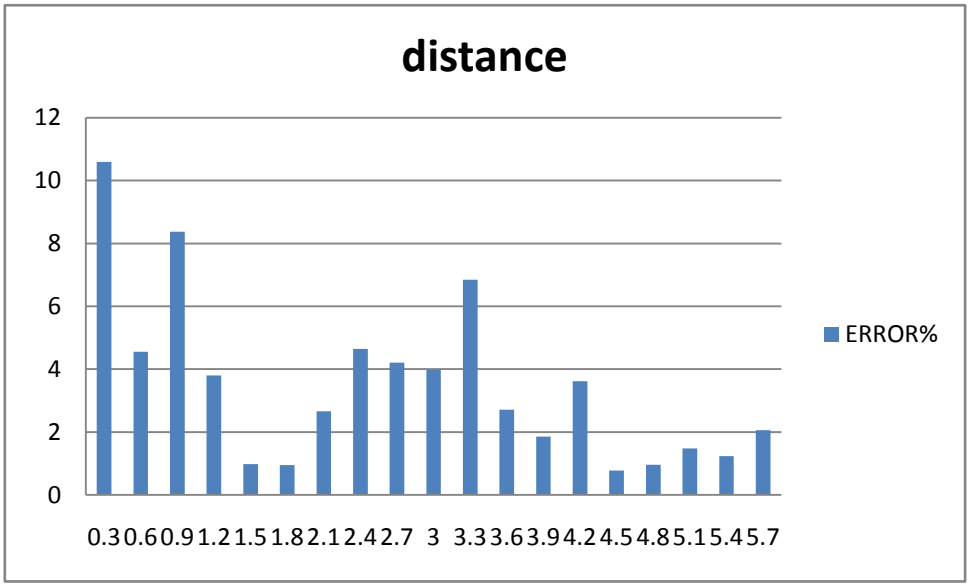
In this experiment we want to find the error in our distance measurement step.

Setup

Stick a measurement tape on the ground to mark the length. Use our device to point on some markings of the tap to measure the distance. The output of the measurement should meet with the corresponding marking of the tape. Check the difference and find out the error.



Result



Conclusion

The error found is quite small and acceptable although there are some inevitable fluctuations of the error. This distance measurement for length measurement can be trusted.

5.4 Comparison with Smart Tools

Introduction

Smart Tools in Google Play store has a distance measurement function as introduced in chapter 1. Our length measurement function has an intermediate step of finding the distance between the device and the corner. The two programs can be compared.

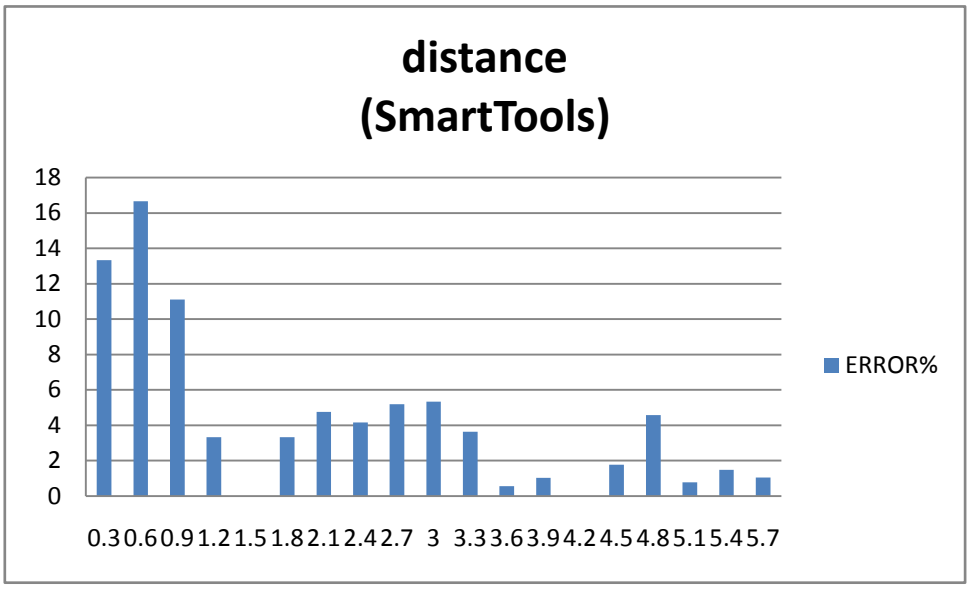
Objective

In this experiment we want to find a comparison with smart tools about which implementation is more accurate.

Setup

Repeat the step in section 5.3 with Smart Tools distance measurement function to measure the distance.

Result



Conclusion

The error of the Smart Tools measurement is similar to our program. It is also fluctuating error from near point to farther point which maybe a evidence for fluctuating digital sensors reading. And this error is vital. Thus smart tools distance measurement function is also quite accurate measurement and it is comparable to our program.

5.5 Testing on length measurement

Introduction

Length measurement function between the two corners is one of the major functions of our program. In this part, we do experiments on measuring length between two corners of a room.

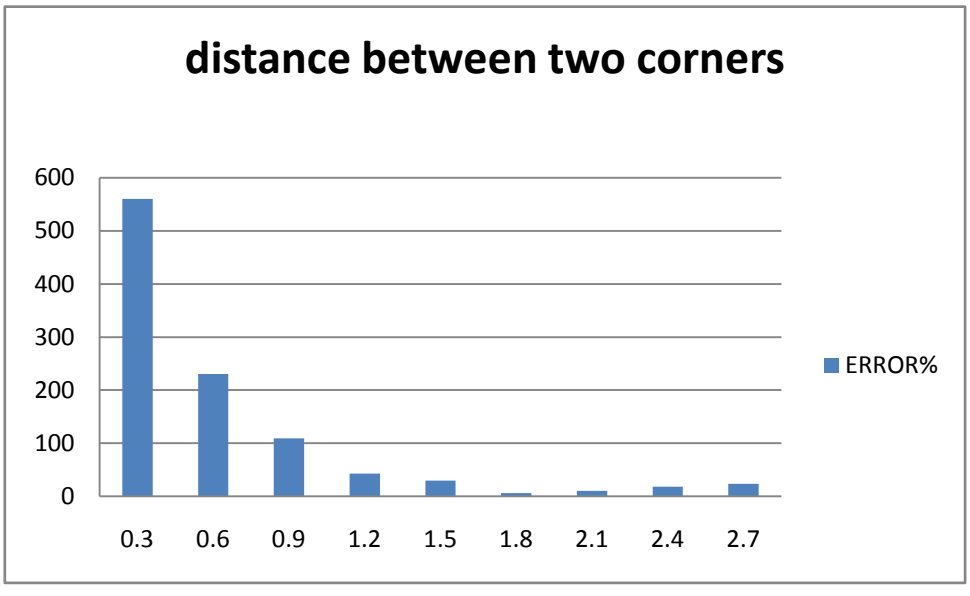
Objective

Find out the error of our length measurement function.

Setup

Point our device with two horizontal points with known length measured by ruler. Then use the length measurement function of our application to measure the length and record the difference from the actually measurement.

Result



Conclusion

The error of length measurement is very large and quite disappointing. This is most probably because of the error propagated from the azimuth since the error found in section 5.2 is quite large. The error is magnified after using the azimuth value in computing the length. There are also other possible mistakes in the assumption in formulating the geometric method to find the length.

5.6 Testing on Floor Plan Generation Algorithm

Introduction

Our floor plan generation algorithm can generate a set of points of a floor plan by inputting an array of angle. The most important factor for the accuracy is whether the distance between the first point and last point generated is very small. In this test, we want know whether the two points merges together.

Objective

To test whether our floor plan generation algorithm can work normally as expected.

Setup

Using command line on PC, we write a simple java program and enter some different degrees of angle input to see distance between the last point and

the first point is it small enough. We have entered 10 sets of different angles of square rooms to generate the coordinates of the corner of the floor plan

Result

On average, the minimum distance between the first point and the last point generated = **0.00656** ,

which can be considered as small. Thus, the difference between the first and the last point is quite small that can be ignored when the angles array are appropriately inputted(meeting the assumption that the person is inside the room to capture the corners).

Conclusion

Our floor plan generation algorithm works fine if the input angles array obey the assumption we have made such as the user is standing in the room to capture is scene. It crashes when four corner of the room is in front of the user but this case is impossible if the use is standing inside a room.

5.7 Testing on Floor Plan Generation

Introduction

Floor plan generation is another major function of our application. In floor plan generation, the aspect ratio of the sides of a floor plan is the major factor determining the accuracy of our function.

Objective

To see whether our application perform floor plan generation function properly.

Setup

We have chosen SHB 102 fun room as our experiment venue. We do testing on floor plan generation inside the room. We first measured the length and width with physical instrument and calculate the ratio. Then we use our application to create a floor plan can measure the aspect ratio and compare with the actual measurement.

Result

Average error after 7 trials = 2.80086293%

The error is surprising small and is wield

Conclusion

The result is quite wield since the error is even less than the error found in section 5.2. Thus there maybe other factors affected the accuracy.

5.8 Comparison with MagicPlan

Introduction

MagicPlan is interactive indoor design application for iPad. As discussed in chapter 1, it has a floor plan creation function. We can compare the floor plan generated from our application with it.

Objective

To compare the accuracy on the aspect ratio of floor plan generated between MagicPlan on iPad and our application.

Setup

Repeat the step in test 5.7 by using MagicPlan on iPad instead of our application.

Result

Average Error = 9.12571225%

Unexpected license error was occurred during the testing. The MagicPlan stopped and cannot do further testing after 3 trails. We can only use in-completed data to analyze.

Conclusion

Although there is no enough data to complete the test, we see that the MagicPlan on iPad is quite accurate in some times. A possible reason maybe that iPad has a more accurate and stable compass reading than android devices. Also, of course it maybe because they use a good implementation

since MagicPlan is a commercial product. We cannot tell which one is better since we don't have enough data to make a valid conclusion.

Yet MagicPlan plan still have things good for us to learn. Their UI is simple to use and UI is interesting in which we also wish to include these good-mobile-application elements to our app so that it become more user friendly.

6. Conclusion

6.1 Difficulties

We have faced many difficulties during our project development in this semester. The first semester is rather short. We had hard times in meeting the tight schedule to solve these challenges.

Android development

Although Android project development uses Java language, there are quite a lot of different from writing ordinary Java programs. We have to read and investigate a lot to achieve what we want to do. For example, using openGL plug-in for Android project is different.

Fluctuating sensor reading

In measurement part, the reading of compass is fluctuating all the time that we cannot obtained a steady record. This brought the problem of inaccuracy of the result obtained. We need to do a lot more testing and try possible solution such as get a set of value and take the mean. This require a lot of time on testing and experimenting what we have done. This is a big challenge and still requires hard work to be done.

Limited computing resources

Although mobile devices have high computing power compare to previous generation of mobile phone and PDA, we still need to take careful control of

resources. The program runs very slow when we add an openGL activity to the program and often crashes when switching between main activity and openGL activity. Although the program runs fairly well in current stage, we expect it would get worse when we move on. Thus we have to put some effort on smoothing our program runs.

Reading research material

At the beginning understanding research level materials are not comfortable to us. It is not so easy to utilize previous released research work to develop our own project. Time is needed to train to read research paper effectively and efficiently.

6.2 Summary

To sum up, in semester one, we have first outlined our project idea on mobile indoor localization application from the mobile technology background and different applications observed in the market. After that, we have studied topics related to our project development. They include interactive length measurement, capturing indoor scene and information and augmented reality. We have taken reference from these topics in our project development.

Then we designed our project work. Our project consists of a core part and an extension part. The core part has functions include capturing sensor readings, interactive length measurement, floor plan generation and localization scheme. The extension part will be using our core part to build augmented reality function on it.

We have chosen ASUS transformer prime TF201 as our building and testing device. Our program is built on Android platform. Our implementation progress has completed the first four functions of the core part of our project. We have built a basic user interface with functions including capturing indoor scene from sensor reading, length measurement and floor plan generation.

After implementing the functions, we have done testing and experiments to understand the accuracy and error of our program. We have been modifying our algorithm to reduce the error after testing. Although some experimental results are still quite disappointing, we will try our best to find a solution to minimize the problem. This requires more future work to work on.

There are also other difficulties we faced and have to overcome in doing this project. Some are difficult to solve and need more time on finding better solution such as the error in length measure.

6.3 Future work

We still have a long way to complete rest of the parts in our project design. In the coming semester, we have to first complete the localization function in our core part which is recognizing current position on a pre-generated floor plan.

Then we will build 2 applications example using the core part as our extension part. The first one is a construction of 3D environment from the 2D floor plan. And add different 3D objects to the room.

The second application is an interactive AR game using the core part. Put an augmented basketball stand on one of the position of the generated floor plan. And then user can arbitrary position in the room can shoot a ball to the basket to score marks.

Our **ultimate goal** of this project is not restricted our core localization function for ourselves to build application only. We want to restructure our core part to a **library** which is open to other developers to use. Thus, everyone can use this localization scheme to build the extension application with their own creative ideas. The development of this localization scheme will not be restricted.

And at the same time, since we are quite unsatisfied with our measurement result, we will work on some optimization to our program so as to improve the accuracy of our work. Also, try to optimize the running time of our

application.

All in all, we will complete our project development core part and extension part, refining and optimizing our measurement function and accuracy of our program functions. And our final goal is to restructure the core part to a library open to all with creative ideas.

6.4 Reflection

The time in semester 1 is rather short. In this semester, we have tried our best to complete the work. During the project development, we not only have learnt technical skills on mobile development but also acquire different soft-skills. Especially in time management, it is challenging to complete a large project while keeping our

We still have a lot of work to do. We are prepared to work harder in the next semester. We will manage our tight schedule better to avoid unexpected bugs appearance which delays our schedule. We will allocate more time on testing and experiment so as to do better on refining our work.

7. Acknowledgment

We would like to express our sincere thanks to our supervisor Prof. Michael Lyu for spending his precious time to give us useful advice. Prof. Lyu arranges 1-hour meeting on weekly basis for us throughout the project development and helps us by giving invaluable advice, guiding our project to the right direction and keep tracking our work progress.

Moreover, we would like to thank Mr. Edward Yau in VIEW lab for giving us technical support and providing us insights and suggestions to the difficulties we faced during the project.

8. References

[1]Paul A. Zandbergen and Sean J. Barbeau (2011). *Positional Accuracy of Assisted GPS Data from HighSensitivity GPSenabled Mobile Phones.* Journal of Navigation, 64, pp 381399 doi:10.1017/ S0373463311000051

[2]Günther Retscher (2007). *Test and Integration of Location Sensors for a Multisensor Personal Navigator*. Journal of Navigation, 60, pp 107117 doi:10.1017/S037346330700402X

[3]Adutya Sankar(2012) *Capturing Indoor Scene with Smartphones* 2012 ACM 978-1-4503-1580-7/12/10

[4]A. Criminisi(1999) *Single View Metrology* University of Oxford Oxford, UK, OX1 3PJ

[5]. Google maps 6.0
http://googleblog.blogspot.hk/2011/11/new-frontier-for-google-maps-mapping.html
Access on 25/11/2012

[6]. Nokia Destination maps
http://conversations.nokia.com/2012/07/16/nokia-leads-the-way-with-indoor-mapping/
Access on 25/11/2012

[7]Android API
Developer http://developer.android.com/reference/packages.html
Access on 23/10/2012

[8.] Lauren Darcey and Shane Conder *Augmented Reality GettingS Start On Android.*
http://mobile.tutsplus.com/tutorials/android/android_augmented-reality/

[9.] Lauren Darcey and Shane Conder *Android Barometer Logger: Acquiring Sensor Data*
http://mobile.tutsplus.com/tutorials/android/android-barometer-logger-acquiring-sensor-data/

[10] "openGL ES" http://www.khronos.org/opengles/

[11] Ng Ka Hung, Chan Hing Fat (2011) *Digital Interactive Game Interface Table Apps for ipad*

[12] Guanghui Wang (2005) *Single view metrology from scene constraints* Image and Vision Computing 23 (2005) 831–840

Appendix

Development Environment

Type	Software	Version
IDE	Eclipse	3.7.1.M20110909-1335
Software	Android DDMS	18.0.0.v201203301601-306762
Software	Android Development Tools	18.0.0.v201203301601-306762
Software	Android Hierarchy Viewer	18.0.0.v201203301601-306762
Software	Android Traceview	18.0.0.v201203301601-306762
Software	Eclipse XML Editors and Tools	1.1.201.v201108151912
Software	NVIDIA Debug Manager for Android NDK	14.0.1.201202211602
Software	Structured Source Editor	1.3.1.v201108191312
Software	Structured Source Model	1.1.601.v201108151912
SDK	Android SDK	2.2
SDK	Android SDK	2.3.1
SDK	Android SDK	2.3.3
SDK	Android SDK	3.0

SDK	Android SDK	3.2
SDK	Android SDK	4.0
SDK	Android SDK	4.03

Experimental Data

Section 5.1

Actual	Record 1	Record 2	Record 3	Record 4	Record 5	Average Record	ERROR %
90	-91.278564	-91.4812	-89.43698	-89.481865	-91.278546	-90.591431	0.657145556
80	-81.466225	-81.99104	-82.4332	-82.66409	-81.736465	-82.058204	2.572755
70	-72.03988	-71.82651	-71.31792	-70.98081	-70.934074	-71.4198388	2.028341143
60	-59.843685	-61.54993	-59.99825	-60.81079	-60.783038	-60.5971386	0.995231
50	-51.240208	-49.82069	-49.82069	-48.646282	-50.336697	-49.9729134	-0.0541732
40	-38.60473	-40.13329	-39.5383	-41.037655	-39.47822	-39.758439	-0.6039025
30	-28.744356	-28.960182	-29.184547	-29.512732	-28.069708	-28.894305	-3.68565
20	-21.032894	-19.035744	-19.832718	-19.707102	-18.822702	-19.686232	-1.56884
10	-8.268817	-7.836118	-7.836118	-10.287818	-8.389249	-8.523624	-14.76376
0	-1.286808	-1.2550343	-1.36802	-1.2899705	-1.428162	-1.32559896	#DIV/0!

Section 5.2

Actual	Record 1	Record 2	Record 3	Record 4	Record 5	Average Record	ERROR %
90	85.056836	88.475105	88.291138	95.020697	89.921018	89.3529588	-0.718934667
80	82.26912	85.16892	55.90677	101.40787	60.52647	77.05583	-3.6802125
70	51.04996	61.51413	49.47695	50.27954	71.37964	56.740044	-18.94279429
60	68.86748	59.49368	38.85028	51.54046	45.72612	52.895604	-11.84066
50	46.9615924	32.94207	26.061656	37.92486	37.65034	36.30810368	-27.38379264
40	37.45199	37.71032	37.9109	37.79885	39.02487	37.979386	-5.051535
30	29.178415	27.78319	26.05632	27.69775	28.495	27.842135	-7.192883333
20	17.680055	18.56778	16.244725	17.45257	18.965976	17.7822212	-11.088894
10	9.231094	8.13953	9.157315	8.395985	8.83277	8.7513388	-12.486612
0	0.002684	0.00087	0.01078	0.00012	0.00231	0.0033528	#DIV/0!

Section 5.3

Actual	Record 1	Record 2	Record 3	Record 4	Record 5	Average Record	ERROR %
0.3	0.27447743	0.270648	0.26472333	0.26560369	0.26569253	0.268228997	-10.59033423
0.6	0.56114966	0.56114966	0.57431161	0.58688098	0.5800244	0.572703261	-4.5494565
0.9	0.80952196	0.82780427	0.8326009	0.83455482	0.8187226	0.82464091	-8.373232192
1.2	1.24811884	1.23171178	1.24555855	1.25029752	1.2525796	1.245653257	3.804438106
1.5	1.4725956	1.47449823	1.48695626	1.50142058	1.49091395	1.485276924	-0.981538429
1.8	1.7837371	1.7867709	1.79207884	1.77848482	1.77344074	1.782902482	-0.949862099
2.1	2.03002886	2.03803247	2.0569866	2.05588819	2.03971068	2.044129359	-2.660506697
2.4	2.28134634	2.28037899	2.289521	2.30392534	2.28753167	2.288540667	-4.644138891
2.7	2.54932023	2.58238568	2.60852372	2.58500495	2.60629697	2.58630631	-4.210877403
3	2.87144616	2.89117773	2.89715403	2.87687063	2.86756442	2.880842593	-3.97191355
3.3	3.09662118	3.08937443	3.07293255	3.02364999	3.08801265	3.074118158	-6.844904296
3.6	3.49731529	3.51026568	3.50425008	3.47946518	3.52018301	3.502295849	-2.714004193
3.9	3.84369555	3.81319717	3.81764473	3.82597999	3.83761769	3.827627026	-1.855717284
4.2	4.00903571	4.03014963	4.09578206	4.07096962	4.03537367	4.048262136	-3.612806275
4.5	4.45562787	4.45025831	4.45909708	4.46114022	4.49971804	4.465168304	-0.774037688
4.8	4.77634989	4.77531098	4.7956938	4.72722218	4.69663645	4.754242662	-0.953277883
5.1	5.10577111	5.08383602	5.00424073	5.00898852	4.92078767	5.024724808	-1.475984165
5.4	5.31873449	5.36534792	5.34451715	5.33070232	5.30675093	5.333210562	-1.236841453
5.7	5.61239333	5.55672978	5.5293446	5.51603494	5.69925014	5.582750557	-2.057007765

Section 5.4

Actual	Record 1	Record 2	Record 3	Record 4	Record 5	Average Record	ERROR %
0.3	0.4	0.3	0.3	0.3	0.4	0.34	13.33333333
0.6	0.7	0.7	0.7	0.7	0.7	0.7	16.66666667
0.9	1	1	1	1	1	1	11.11111111
1.2	1.2	1.3	1.3	1.2	1.2	1.24	3.333333333
1.5	1.5	1.5	1.5	1.5	1.5	1.5	0
1.8	1.7	1.8	1.7	1.7	1.8	1.74	-3.333333333
2.1	2	2	2	2	2	2	-4.761904762
2.4	2.3	2.3	2.3	2.3	2.3	2.3	-4.166666667
2.7	2.5	2.6	2.5	2.6	2.6	2.56	-5.185185185
3	2.8	2.8	2.9	2.8	2.9	2.84	-5.333333333
3.3	3.2	3.1	3.2	3.2	3.2	3.18	-3.636363636
3.6	3.6	3.5	3.6	3.6	3.6	3.58	-0.555555556
3.9	3.8	3.9	3.8	3.9	3.9	3.86	-1.025641026
4.2	4.2	4.1	4.2	4.3	4.2	4.2	0
4.5	4.4	4.4	4.5	4.4	4.4	4.42	-1.777777778
4.8	4.6	4.6	4.6	4.6	4.5	4.58	-4.583333333
5.1	5	5	5	5.1	5.2	5.06	-0.784313725
5.4	5.3	5.2	5.4	5.4	5.3	5.32	-1.481481481
5.7	5.7	5.8	5.6	5.6	5.5	5.64	-1.052631579

Section 5.5

Actual	Record 1	Record 2	Record 3	Record 4	Record 5	Average Record	ERROR %
0.3	2.23747121	1.91649386	1.91512093	1.92933772	1.90406776	1.980498295	560.1660984
0.6	2.23747121	1.91649386	1.91512093	1.92933772	1.90406776	1.980498295	230.0830492
0.9	1.86982232	1.88233829	1.89335026	1.87981909	1.88235987	1.881537966	109.059774
1.2	1.77422749	1.70867672	1.69790864	1.68565029	1.69879098	1.713050825	42.75423544
1.5	1.80608189	1.98794887	1.97977916	1.98055541	1.98035696	1.946944457	29.79629712
1.8	1.99124896	1.87430449	1.8909458	1.90314318	1.8907942	1.910087324	6.115962433
2.1	1.90310504	1.91443825	1.86234659	1.86951013	1.87161493	1.884202987	-10.27604824
2.4	1.86739647	1.9977977	1.98174069	1.97663081	1.97558366	1.959829864	-18.34042235
2.7	2.03442071	2.07570268	2.03936342	2.07029917	2.08762958	2.061483113	-23.64877359

Section 5.6

Input Angle 1	Input Angle 2	Input Angle 3	Input Angle 4	minDist
90	90	90	90	4.93E-14
120	90	120	50	5.63E-15
150	30	120	60	0.011545283

20	160	20	160	1.16E-13
144	67	123	26	0.007744714
133	56	120	51	0.007828804
88	128	76	68	0.007305098
110	70	115	65	0.00413989
70	120	70	100	6.56E-14
170	40	115	35	8.14E-04

Section 5.7

Acutual	width	length	width/length	ERROR %	AVG ERROR %
	480	760	0.63157895	0	0
Measured	6	9	0.66666667	5.55555556	2.80086293
	3.4	5.4	0.62962963	0.30864198	
	3.2	5.3	0.60377358	4.40251572	
	3.2	5.3	0.60377358	4.40251572	
	3.2	5.1	0.62745098	0.65359477	
	3.4	5.2	0.65384615	3.52564103	
	3.5	5.5	0.63636364	0.75757576	

Section 5.8

Acutual	width	length	width/length	ERROR %	AVG ERROR %
	480	760	0.63157895	0	0
Measured					
	7.9	12	0.65833333	4.23611111	9.12571225
	7.9	13	0.60769231	3.78205128	
	9.8	13	0.75384615	19.3589744	