

**SMART NAVIGATION SYSTEM – MOBILE NAVIGATION CLIENT  
DEVELOPMENT**

**BY**

**KOH PENG HOW**

**A REPORT**

**SUBMITTED TO**

**Universiti Tunku Abdul Rahman**

**in partial fulfilment of the requirements**

**for the degree of**

**BACHELOR OF COMPUTER SCIENCE (HONS)**

**Faculty of Information and Communication Technology**

**(Perak Campus)**

**JAN 2015**

UNIVERSITI TUNKU ABDUL RAHMAN

**REPORT STATUS DECLARATION FORM**

**Title:** Smart Navigation System – Mobile Navigation Client Development

**Academic Session:** Jan 2015

I KOH PENG HOW  
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in  
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

\_\_\_\_\_  
(Author's signature)

\_\_\_\_\_  
(Supervisor's signature)

**Address:**

39, Jalan Setia 9/19  
Taman Setia Indah  
81100 Johor Bahru, Johor.

\_\_\_\_\_  
Supervisor's name

**Date:** \_\_\_\_\_

**Date:** \_\_\_\_\_

**SMART NAVIGATION SYSTEM – MOBILE NAVIGATION CLIENT  
DEVELOPMENT**

**BY**

**KOH PENG HOW**

**A REPORT**

**SUBMITTED TO**

**Universiti Tunku Abdul Rahman**

**in partial fulfilment of the requirements**

**for the degree of**

**BACHELOR OF COMPUTER SCIENCE (HONS)**

**Faculty of Information and Communication Technology**

**(Perak Campus)**

**JAN 2015**

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**SMART NAVIGATION SYSTEM – MOBILE NAVIGATION CLIENT DEVELOPMENT**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : \_\_\_\_\_

Name : \_\_\_\_\_

Date : \_\_\_\_\_

## **ACKNOWLEDGEMENT**

First and foremost, I would like to extend my heartfelt gratitude to all those who provided me the possibility of completing this project. A special acknowledgement I would like to express my gratitude towards my final year project supervisor, Dr. Ng Hui Fuang, who have willingly shared his precious time, experience and knowledge during this project. He have been inspiring me work greatly in this project as well as been showing me examples related to the project.

Furthermore, a special thanks goes to my teammate, Chin Chee Yang, who develop the server-side of the application and gave suggestions during the project discussion. Finally, an honorable gratitude goes to my beloved families and friends for understanding and supporting me in completing this project.

## ABSTRACT

Innovation of technology in our world has expanded dramatically. Today, mobile devices such as smartphones or tablet PCs have become a necessity in our daily life. Most of the people enjoy using mobile applications to solve their daily problems, one of the popular applications is the navigation application. However, most of the existing navigation systems are unable to provide precise routes and additional information of a building or a landmark within a region such as a campus, a shopping mall, a hospital and etc.

Hence, this project is aimed to develop an informative, reliable and precise guidance system in order to solve the problems face by Universiti Tunku Abdul Rahman (UTAR) freshman and visitors during the adaptation to the unfamiliar environment. In this project, we will develop a mobile application that consists of two modules, the “Building Recognition” and the “Campus Navigation”.

The “Building Recognition” is used for recognizing a building or landmark and for displaying recognition result and related information on the mobile device. The system will collect the user’s location data and the data of acquired image and is sent to the server for recognition process. After the recognition process is done, the server will return the recognition result and related information back to client-side. Once the client-side application received the result and information, it will overlay the result and information on the camera view.

The “Campus Navigation” is used for navigating the user from one place to another within the campus. The system will acquire the user’s location information and display the POIs around the user’s current location on the camera view by using augmented reality techniques. Once the user proceeds to routing process, the system will show the best route on the map according to the selected destination. To perform the navigation, the user’s location and route on the map will be updated in periodically.

## TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>DECLARATION OF ORIGINALITY</b>	<b>ii</b>
<b>ACKNOWLEDGEMENT</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENTS</b>	<b>v</b>
<b>LIST OF FIGURES</b>	<b>vii</b>
<b>LIST OF TABLES</b>	<b>ix</b>
<b>LIST OF ABBREVIATIONS</b>	<b>x</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1-1 Problem Statement	1
1-2 Background	1
1-3 Motivation	2
1-4 Project Objectives	2
1-5 Proposed Approach	3
1-6 Achievement	4
1-7 Report Organization	4
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>7</b>
2-1 Review on Similar Application	7
2-1-1 3D Mobile Augmented Reality in Urban Scenes	7
2-1-2 Staff Outdoor Positioning System	10
2-2 Location Tracking of Mobile Devices	11
2-2-1 GPS-based Positioning	11
2-2-2 Wi-Fi-based Positioning	12
2-3 Keypoints Extraction and Description	14
2-3-1 Scale Invariant Feature Transform (SIFT) algorithm	14
2-3-2 Oriented FAST and Rotated BRIEF (ORB) algorithm	17

2-4 Augmented Reality	19
2-5 Critical Remarks of Previous Works	20
<b>CHAPTER 3 SYSTEM DESIGN</b>	<b>22</b>
3-1 System Overview	22
3-1-1 Building Recognition	22
3-1-2 Campus Navigation	28
<b>CHAPTER 4 METHODOLOGY AND TOOLS</b>	<b>36</b>
4-1 Methodology	36
4-2 Tools to Use	37
4-2-1 Hardware	37
4-2-2 Software	37
4-3 Requirement	39
4-4 Verification Plan	40
<b>CHAPTER 5 IMPLEMENTATION AND TESTING</b>	<b>42</b>
5-1 Overview	42
5-2 The connection between client and server	42
5-3 The best route provided by the “Campus Navigation”	44
<b>CHAPTER 6 CONCLUSION</b>	<b>46</b>
6-1 Conclusion	46
6-2 Limitation	46
6-3 Future Work	47
<b>REFERENCE</b>	<b>49</b>



## LIST OF FIGURES

Figure 1-1 Smart Navigation System Design Diagram .....	4
Figure 2-1 Mobile Augmented Reality 3D offline processing system (Takacs, et al., 2011) .....	9
Figure 2-2 Mobile Augmented Reality 3D online processing system (Takacs, et al., 2011) .....	10
Figure 2-3 A screenshot of Mobile Augmented Reality 3D running with label augmentation and contour/plane detection (Takacs, et al., 2011).....	10
Figure 2-4 Architecture of the GUI campus map (Leow, et al., 2011).....	11
Figure 2-5 Overview of trilateration process (physics.org, 2014).....	12
Figure 2-6 Nearest Access Point technique (small triangle indicate user's position) (Shah & Shah, 2014).....	13
Figure 2-7 Fingerprint or Received Signal Strength Indication technique (Shah & Shah, 2014) .....	14
Figure 2-8 Trilateration technique (Shah & Shah, 2014) .....	14
Figure 2-9 A pixel (marked with X) is compared to its 26 neighbours in 3x3 areas at the current and adjacent scales (marked with circles), so that the local minima and maxima of DoG images can be detected. (Lowe, 2004).....	16
Figure 2-10 Keypoint descriptor using the gradient magnitude (Lowe, 2004).....	17
Figure 2-11 The result of comparing the intensity centroid (IC) method with another two gradient-based measures, MAX and BIN. (Rublee, et al., 2011) .....	19
Figure 2-12 Augmented Reality View on a Smartphone.....	19
Figure 3-1 Image captured from a real world scene .....	22
Figure 3-2 Directories dialog (Top) and uploaded image (Bottom) (Upload image process) .....	23
Figure 3-3 Convert the color image (Left) to grayscale image (Right) .....	24
Figure 3-4 Error message when location data is not obtainable .....	25
Figure 3-5 XML file contains user's location coordinate and image data.....	25
Figure 3-6 Connection timeout error message.....	26
Figure 3-7 Building not found screenshot (Recognition result) .....	27
Figure 3-8 Building recognized successfully screenshot (Recognition result).....	27
Figure 3-9 Coordinate does not exist in the server screenshot (Recognition result) ...	28
Figure 3-10 Building Recognition Module flow diagram .....	28
Figure 3-11 Entity Relational Diagram (ERD).....	29

Figure 3-12 POIs around the user's current location .....	31
Figure 3-13 Adjust the radar radius: before (Left) and after (Right) .....	31
Figure 3-14 Selected destination information dialog (Left) and travel modes dialog (Right) .....	32
Figure 3-15 Near to destination pop-up message (Left) and destination reached dialog (Right) .....	33
Figure 3-16 Routes details in JSON file format .....	34
Figure 3-17 Route drawn on google map .....	35
Figure 3-18 Campus Navigation Module flow diagram .....	35
Figure 4-1 Prototyping Methodology .....	36
Figure 6-1 Screenshot of 3-D arrow pointing to the route direction with the minimized map .....	47

## LIST OF TABLES

Table 3-1 Data Dictionary .....	30
Table 4-1 Smartphone Hardware Specifications .....	37
Table 4-2 Personal Computer Hardware Specifications .....	37
Table 4-3 Verification P1.....	40
Table 4-4 Verification P2.....	41
Table 5-1 Connection between client and server testing results.....	42
Table 5-2 Best route provided by the “Campus Navigation” testing results .....	44

## LIST OF ABBREVIATIONS

<i>ADT</i>	Android Development Tools
<i>API</i>	Application Programming Interface
<i>AR</i>	Augmented Reality
<i>BRIEF</i>	Binary Robust Independent Elementary Features
<i>DoG</i>	Difference-of-Gaussian
<i>FAST</i>	Feature from Accelerated Segment Test
<i>FYP</i>	Final Year Project
<i>GPS</i>	Global Positioning System
<i>GUI</i>	Graphical User Interface
<i>IDE</i>	Integrated Development Environment
<i>IMEI</i>	International Mobile Equipment Identity
<i>JNI</i>	Java Native Interface
<i>JSON</i>	JavaScript Object Notation
<i>KM</i>	Kilometer(s)
<i>OpenCV</i>	Open Source Computer Vision Library
<i>ORB</i>	Oriented FAST and Rotated BRIEF
<i>OS</i>	Operating System
<i>PC</i>	Personal Computer
<i>PNG</i>	Portable Network Graphics
<i>POI</i>	Point of Interest
<i>QEMU</i>	Quick Emulator
<i>RSSI</i>	Received Signal Strength Indication
<i>SDK</i>	Software Development Kit
<i>SIFT</i>	Scale-Invariant Feature Transform
<i>TCP</i>	Transmission Control Protocol
<i>UI</i>	User Interface
<i>URL</i>	Uniform Resource Locator
<i>USB</i>	Universal Serial Bus
<i>UTAR</i>	Universiti Tunku Abdul Rahman
<i>XML</i>	Extensible Markup Language

## CHAPTER 1 INTRODUCTION

### 1-1 Problem Statement

Nowadays, most people feel inconvenient to search for what they need from the search engine by typing words, especially when they do not know the keywords that they should be using. For instance, a traveller facing a building, but he/she would like to know more about the building such as its history, point of interest (POI) within the building and etc. In this kind of situation, they would prefer an advanced searching function, which is able to search directly with images and return the related information.

In addition, there exist many advanced navigation system but most are unable to provide routes precisely as well as information of building within a region such as campus, shopping mall, hospital and etc.

Therefore, an informative, reliable and precise guidance system is very important in this technological era. This guidance system should be able to search the building information with the building's photo and return the information regarding the building. Furthermore, it should be able to navigate the user no matter the user is under the indoor or outdoor environment. The guidance system must be user-friendly and able to process data efficiently.

### 1-2 Background

Smart devices such as smartphones or tablet PCs have become necessities in our daily life, these smart devices are very powerful and affordable for majority of the people. There are many different categories of mobile application such as education, health & fitness, weather, business, travel, social networking, navigation and etc. People use such applications to solve their daily problem.

Moreover, people who likes to travel do not mean that they are familiar with the places around. This can be troublesome if they do not know the shortest path to their destinations causing the increase in travel cost. Therefore, a navigation system application would be very useful. However, there are people are not satisfied with the existing navigation system, because it does not provide any information about the destination, and thus they have to search for it separately. Hence, a system that can provide basic navigation functionality and information about the destinations would be very useful.

Furthermore, a lot of high specification smart phones are released recently. They do not only receive and make phone calls, voicemail, and text messages, but also have many different kinds of advanced functionality such as accessing the internet, access digital media, taking high quality of photo and etc. It has become a core piece of technology in the modern world.

### **1-3 Motivation**

Although there exist numerous robust outdoor navigation system such as Waze Social Global Positioning System (GPS) Maps & Traffic, Papago, Sygic and etc., which are mobile applications that are able to navigate the user base on the best route to selected destination and provides some extra information such as real time traffic conditions, road information and etc. However, even with our advance developments in navigation system, the system are still unable to provide precise routes as well as extra information of building such as history, various department and etc. within a region like campus, shopping mall, hospital and etc.

Moreover, there isn't a lot of application that provides image searching method. Most of the existing image searching application are only able to search a small product like books or other kinds of small object.

### **1-4 Project Objectives**

The general objective of this project is to provide UTAR freshmen and visitors a smart guidance system when they first visit the UTAR Perak campus. The specific objectives of this project are:-

- To develop a mobile application that is able to capture real time image and send it to server along with coordinate details.
- To develop a system that can recognize a building by using image and coordinate from client side and an image database.
- To develop a system that can provide building information accurately to the client side.
- To develop a system able to navigate user to the selected destination.

### **1-5 Proposed Approach**

The “Smart Navigation System” consist of two modules, which is the “Building Recognition” and the “Campus Navigation”. Due to the constraints of processing power and storage availability of mobile devices, the “Building Recognition” implements the client-server architecture with a single central server that will manage connections between clients. For client-side, there is a mobile application to interact with the user, the application is designed to have a simple and user-friendly interface. It will collaborate with the server-side by transmit the image data and user’s geolocation data to the server for further processing. For server-side, there is a server running a recognition program and it is also a database storage system. Once it has finished a recognition task, it will return the recognition result along with the information back to the client-side if the recognition result is positive; otherwise, it will just return the recognition result.

The final deliverable of this project is a client-side mobile application contains the “Building Recognition” and the “Campus Navigation”. The “Building Recognition” will acquire the image captured or uploaded by the user. In order to accelerate the data upload speed, the acquired image will undergo grayscale conversion and compression in the mobile device. Therefore, the data packet can be minimize. In the recognition task, the system will obtain the user’s location data and send along with the processed image data to the recognition server to proceed recognition process. Afterwards, the server will return the recognition results and related information back to the client device. In order to display those information to the user, the application will overlay those information on the camera view.

The “Campus Navigation” will acquire the user’s geolocation information and display the POIs around the user’s current location on the camera view by using augmented reality techniques. Once the user proceed to routing process, the system will display the best route on the map according to the selected destination and travel mode. To perform the navigation, the user’s location and route on the map will be updated in periodically. In addition, if the user near to the destination a pop-up message will show and propose user to switch to “Building Recognition” module.

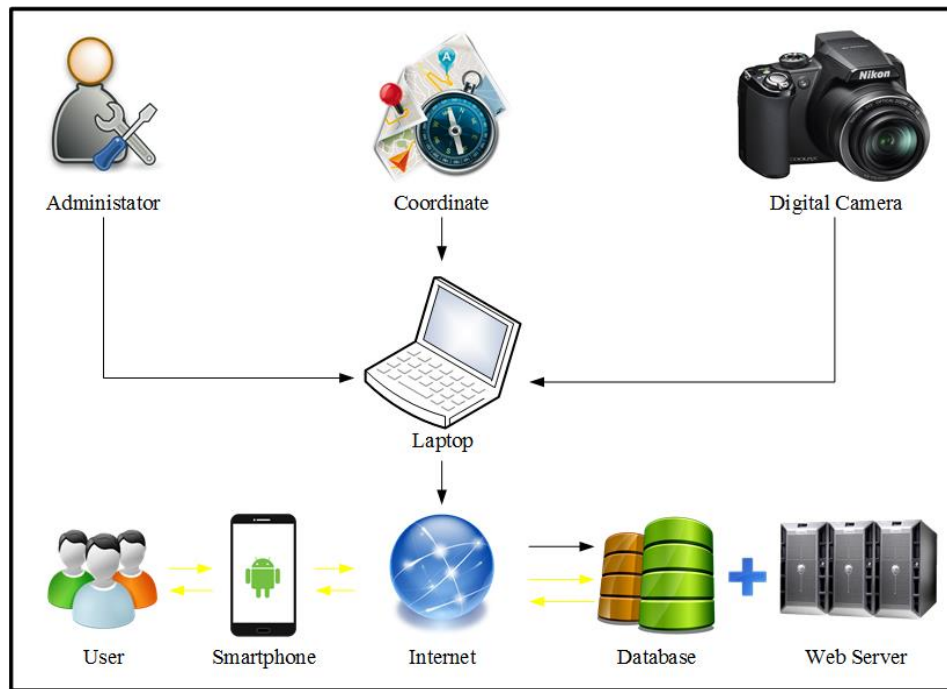


Figure 1-1 Smart Navigation System Design Diagram

### 1-6 Achievement

This project is able to assist the UTAR freshmen adapting the new campus environment. Furthermore, the visitors are also able to use the “Campus Navigation” in the “Smart Navigation System” application to do the navigation. They can also use the “Building Recognition” in this system to solve their problem of finding various departments within the building instead of asking others help.

It is able to reduce the problems of UTAR freshmen and visitors during the adaptation to the unfamiliar environment. With this application, they can easily travel around the campus without extra effort.

### 1-7 Report Organization

Project introduction was introduced in Chapter 1 and consisted of seven sub-chapters which is problem statement, background, motivation, project objectives, proposed approach, achievement and report organization. In the problem statement, the existing problem to be solved was stated. The existing technology of was discussed in the background. Third sub-chapter is the motivation ,the motivation of this project and why the project is required was stated. The following sub-chapter is project objectives, the aims and purpose if the project was described. In the proposed approach, it proposes the solution used to solve the problem that was illustrated in the previous sub-chapter.



The next sub-chapter is achievement, this sub-chapter highlights the accomplishment. The report outline and each chapter respectively was explained and described in the last sub-chapter which is report organization.

Chapter 2 is literature review, it consists of five sub-chapters in which was reviewed on similar application, location tracking of mobile devices, keypoints extraction and description, augmented reality and critical remarks of previous works. The review on similar application is about to review others similar project that had been developed. The next sub-chapter is the location tracking of mobile devices, this sub-chapter reviewed on existing location tracking technology. The following sub-chapter is the keypoints extraction and description, the two most mature algorithm, SIFT and SURF was reviewed. The next sub-chapter is augmented reality, this sub-chapter reviewed the current state-of-the-art of technology which is AR. Lastly, in the sub-chapter critical remarks of previous works, it described the weakness and strength of any previous work that are similar to this project as well as review of state-of-the-art technology to decide the most suitable technology to be implemented in this project.

Chapter 3 is system design, it illustrated and described the development details of the system design from top to bottom.

Chapter 4 is methodology and tools, it consists of four sub-chapters which is methodology, tools to use, requirement and verification plan. The first sub-chapter is methodology, in the methodology is a brief statement of the methodology for the realization of the project. The following sub-chapter is tools to use, it listed down all the software tools and hardware tools used in this project. The next sub-chapter is requirement, this sub-chapter discussed about the minimum requirements of this system. The last sub-chapter is verification plan, the verification plan analysed the details plan that use to verify this system.

Chapter 5 is implementation and testing, it consists of three sub-chapters which is overview, the connection between client and server and the best route provided by the “Campus Navigation”. The initial sub-chapter is overview, this addressed the overview of the testing scenarios. The following sub-chapter is the connection between client and server, in this sub-chapter is showed the test result of connection between client and server. The last sub-chapter is the best route provided by the “Campus Navigation”, in

this sub-chapter it has showed the test result of best route provided by the “Campus Navigation”.

Chapter 6 is the conclusion, it consists of three sub-chapters, which is conclusion, limitation and future work. The first sub-chapter is conclusion, in this sub-chapter the project was concluded and reviewed. The following sub-chapter is limitation, the existing limitation of this system will be listed. The last sub-chapter is future work, it is about the future improvements and solutions that can solve the limitation is currently faced by the system was proposed.

The reference is the last part of this report and it consist of the list of project references.

## CHAPTER 2 LITERATURE REVIEW

Campus navigator is a mobile application, a campus navigation system that students, employees and visitors to navigate from one place to another within the campus. Most of the navigation application like Papago, Sygic and Waze Social GPS Maps & Traffic and Yahoo Maps are unable to navigate or provide routes precisely within the campus. Hence, it is necessary to have a stable campus navigation system that enables users to obtain the precise route and to be more detailed than any other existing application. (Bangare, et al., 2014)

In the development of the guidance system, we know that providing accurate and comprehensive information is extremely important. In order to develop a robust program, there are some different technology and techniques will be discussed in this chapter.

### 2-1 Review on Similar Application

#### 2-1-1 3D Mobile Augmented Reality in Urban Scenes

In the year 2011, Takacs, Choubassi, Wu and Kozintsev proposed an advanced mobile augmented reality system that is able to recognize a building by using the mobile device's camera in real time and records this real time view with the 3D architecture of the buildings. This system consists of two primary components: online image matching, tracking and augmentation, and offline database processing. Initially, the database is created with 3D models and images. All the data with regards to urban scenes is obtained from a company called Earthmine. With a database server, or possibly a geographically limited subset pre-loaded on the device, the system merge information from the device's sensors and the real time frame from the device's camera. Then it would utilize the tracking and detection algorithms implemented to track and recognize buildings within the scene in addition to estimate the orientation of the mobile device. Accordingly, the system will augment the real time view of the mobile device along with details in the proper object location. The system components will elaborate at below:-

- Server Processing

As mentioned earlier, the data about urban scenes is obtained from Earthmine. Earthmine vehicle gathers images at subsequent GPS neighbourhood areas. All the data is corresponded to dense neighbourhood areas. All of the location will

have a depth map, a spherical panoramic RGB image, points cloud, its orientation and the GPS of the panorama center, a spherical panoramic mask image of planes' index and equations of the plane.

The real time images acquired by the device is not spherical like the data from Earthmine. Hence, the system have to unfold the spherical image panoramas and the spherical panoramic mask by projecting them to 2D image view (see the upper part of Figure 2-1). Moreover, the system will extract the interest points from the 2D image panoramas as mentioned by using the SURF algorithm. Next, the system will extract the contours from a planar segments. These extracted contours will be used in the future as augmentation content. Lastly, the augmentation content will be generated by geocoding (attaching a coordinate to a 2D image segment).

- On-Device Processing

Constrained subset of the database will be pre-loaded according to the data of GPS and orientation sensors and a local server will be set on the device (see the left part of Figure 2-2). The server will update incrementally through the internet to the main preprocessed database as a user's location changes. Additionally, the detection and tracking algorithms requires data from the sensor and real time images from the device (see the right part of Figure 2-2). Tracks and detects the building in the camera's view is the main objective of online building localization. Therefore, measuring the pose of the mobile device and augment corresponding information in the proper perspective.

The detection feature will estimate the orientation and location of the device from the sensor's data such as accelerometer, compass and location services sensor. Moreover, it will extract the features from the real time images by using SURF algorithm, and match those feature with the image feature pre-processed database.

The tracking feature will track the pose of the device by fusing camera inputs in real time and orientation sensor data. In order to track the real time content, the system will utilize an efficient motion estimation technique. This technique is based on a gradient-based strategy, iterative and multi-resolution. For the purpose of compensation for the drift and refine the tracking, the system will fuse the visual input and data from compass sensor and produce the tracking results.

In addition, the building tracking and detection modules requires different length of processing time. In order to coordinate them, the system have to implement multi-threaded framework. The framework also coordinates the output from both of the tracking thread and the building detection thread. The output is first detected, then the device's real time view is augmented with contents based on the updates of the tracking algorithm. It will be then updated to the screen with the recent result.

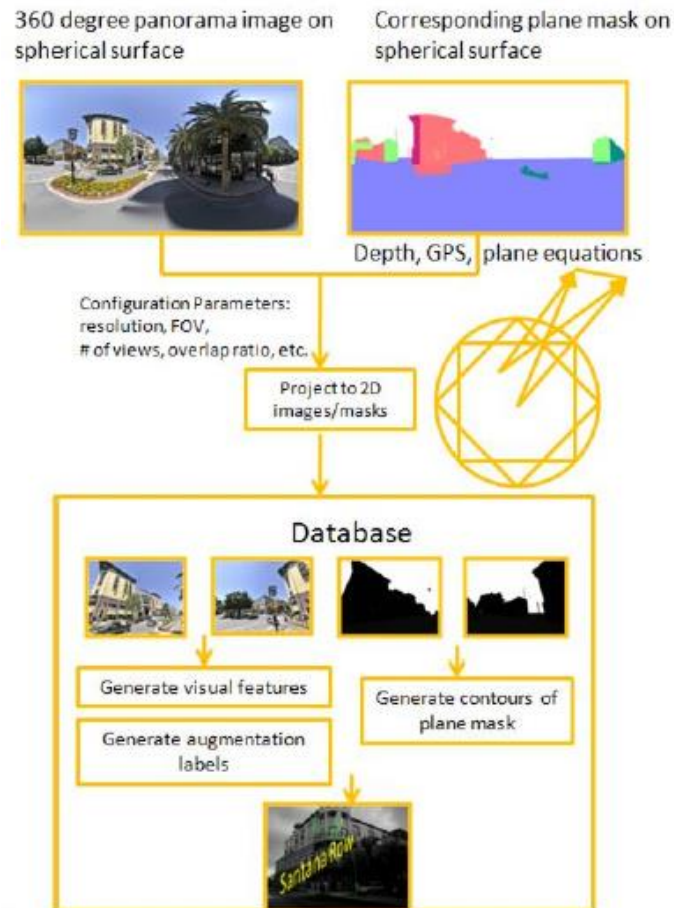


Figure 2-1 Mobile Augmented Reality 3D offline processing system (Takacs, et al., 2011)

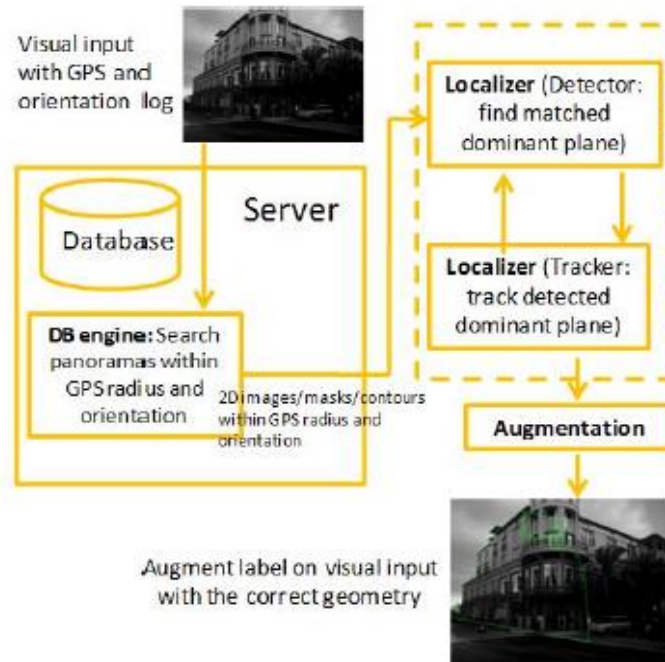


Figure 2-2 Mobile Augmented Reality 3D online processing system (Takacs, et al., 2011)

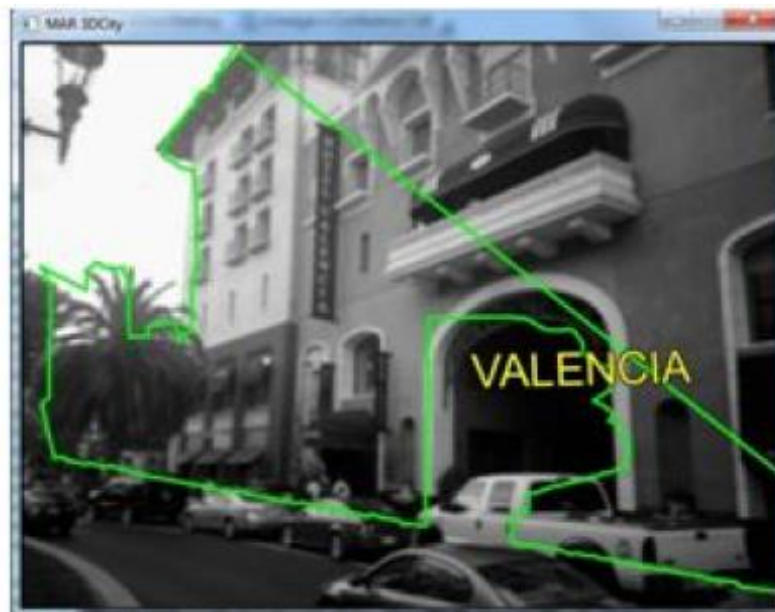


Figure 2-3 A screenshot of Mobile Augmented Reality 3D running with label augmentation and contour/plane detection (Takacs, et al., 2011)

### 2-1-2 Staff Outdoor Positioning System

A positioning system using the GPS-based positioning technique, Google Map and Mobile Network had proposed by Leow, Lau, Le Grange, Laxman and Anantharsekaran (2011). Their proposed solution of this paper is to use a GPS enabled devices to track the device coordinate, and uses the mobile application to send the

information packet. This packet includes the device coordinate and the device International Mobile Equipment Identity (IMEI), in order to make this packet uniquely identified. Next is the graphical user interface (GUI) campus map, they overlay the campus map on the actual Google Maps, and this will be served as the interface to the user. Moreover, the location of the device also can be view on the map. The structure of the GUI campus map is shown in Figure 2-4.

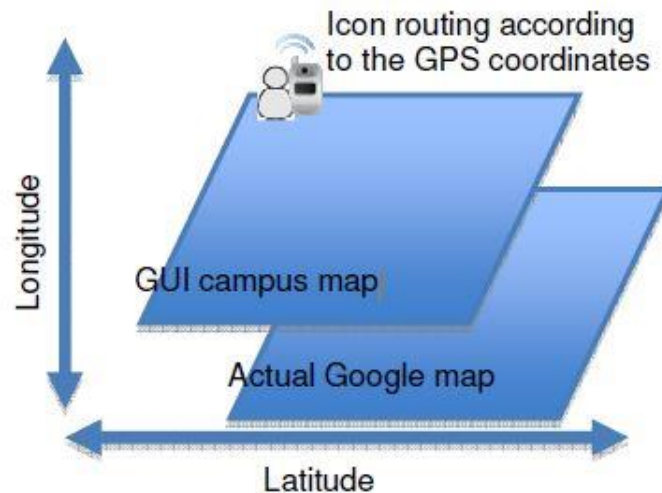


Figure 2-4 Architecture of the GUI campus map (Leow, et al., 2011)

There is a problem such that when the device is under an indoor environment, it will cause the device to lose GPS signal, thus there is no position data send to the server. In order to solve this problem, the authors propose to use the GPS foot printing to induce the last whereabouts of the devices.

## 2-2 Location Tracking of Mobile Devices

### 2-2-1 GPS-based Positioning

The most mature and free positioning system, which is the Global Positioning System (GPS). It is a leading technology to determine user position on the smart device. GPS is a freely accessible system, the device can obtain the location information via satellite positioning services. The signals will transmit from the GPS satellites to the devices that built-in with the GPS receiver on the ground. Majority of the smart devices on the market have built-in GPS receiver, so it will have the capability to receive the GPS signals. To calculate the position of the GPS receiver would require signals from at least four or more satellites. Furthermore, the data transmitted by the satellite consist of position data and also the current time. The process of calculating the position of the

receiver is called trilateration. As illustrate in Figure 2-5, if the receiver know how far the distance from it to satellite A, then it will know that it must be located someplace on the red circle. Same operations are performed on satellites B and C, then it would calculate the location by intersecting the three circles. The GPS receiver uses overlapping the spheres rather than using the circles. The fourth satellite is to eliminate any clock errors between the satellite's clock and the GPS receiver clock, the minor differences of the clock can lead to large errors in determination of location.

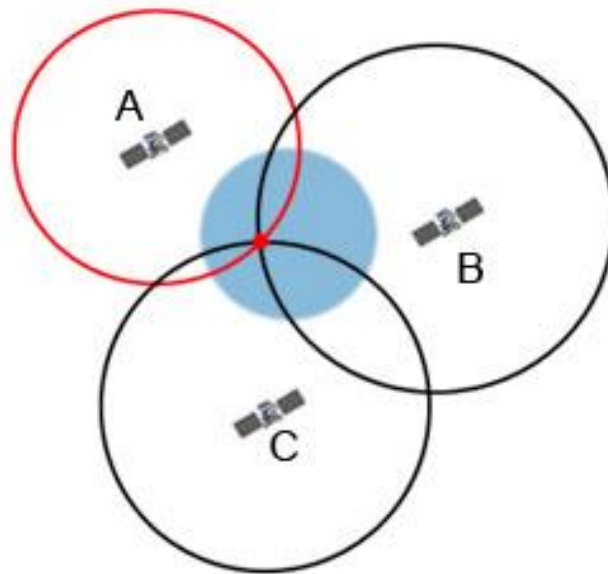


Figure 2-5 Overview of trilateration process (physics.org, 2014)

The position accuracy determined by the GPS depends on the type of receiver. Majority of the consumer receiver have an accuracy of about  $\pm 10$  meters. The limitation of the GPS is that it only works under outdoor environment. (Griffin, 2011)

### 2-2-2 Wi-Fi-based Positioning

Although the GPS positioning method works excellent under outdoor environments, but it is not usable under indoor environment such as underground subway (tunnel), building and etc. Acquiring the user's location information under the indoor environment is still important, because it enables the navigation system to keep track the user in any building or underground subways. Furthermore, the smart device are commonly equipped with Wi-Fi sensors and almost all the modern buildings have Wi-Fi access points. Therefore, it is possible to realize the indoor positioning system. (Shah & Shah, 2014)



To determine the user's indoor location, there are several Wi-Fi positioning techniques as below:-

- Nearest Access Point

This is the easiest way to determine the user's location based on the nearest access point that the user is connected. This technique do not require complex calculations and operations. However, its rate of accuracy is low. This technique illustrate in Figure 2-6.

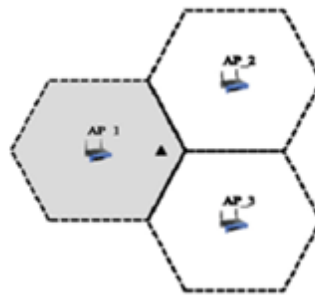


Figure 2-6 Nearest Access Point technique (small triangle indicate user's position)  
(Shah & Shah, 2014)

- Received Signal Strength Indication (RSSI)

This technique estimate the user's location by measuring the distance from a sensor to the nearby access point using the relationship of distance to signal strength. The RSSI can use to deduce the distance to the access points. To calculate absolute loss of the signal strength, it will use the outgoing power level of the Wi-Fi access point and the strength of signal received by the user. Moreover, the distance to a particular access point can be calculated by using the free space path loss equation.

Gather the absolute position of three or more access point and the distance to them, the location of the user can be calculated by using the trilateration algorithm.

This technique illustrated in Figure 2-7.

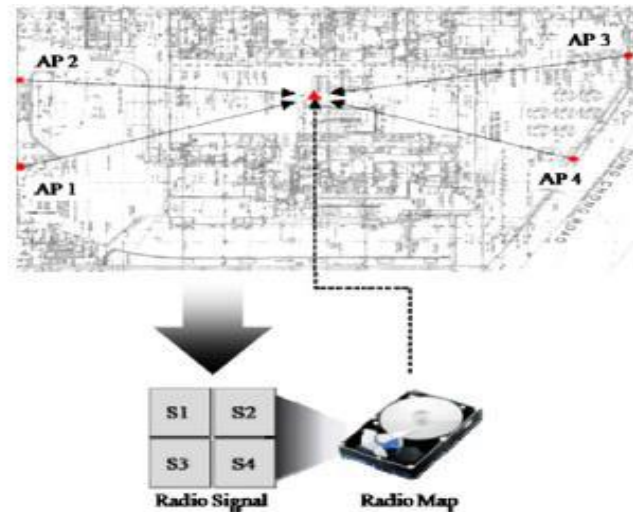


Figure 2-7 Fingerprint or Received Signal Strength Indication technique (Shah & Shah, 2014)

- Trilateration

Trilateration is similar technique to the GPS-based positioning, it calculates a node's location by intersect the three circles. The location of the others three references nodes have to be known and as well as the distance to each of them. This technique illustrate in Figure 2-8.

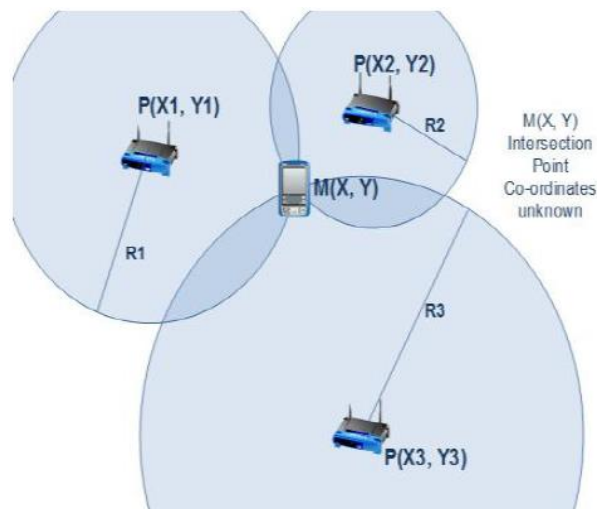


Figure 2-8 Trilateration technique (Shah & Shah, 2014)

## 2-3 Keypoints Extraction and Description

### 2-3-1 Scale Invariant Feature Transform (SIFT) algorithm

In the year 2004, David G. Lowe proposed a feature matching algorithm known as SIFT (Scale Invariant Feature Transform). The SIFT algorithm is widely used in computer

vision applications. It is because the SIFT algorithm are invariant to image rotation, scale, and translation. Furthermore, it provides robust matching across a substantial range of related transformation, addition of noise, change in illumination and change in 3D viewpoint. Therefore, SIFT algorithm are able to detect the object image robust and accurate.

The SIFT algorithm is dependent on the feature spotting in scale space. There are four major stages used to calculate the SIFT descriptors, these four major stages of this algorithm are:-

- Scale-space Extrema Detection

The first stage of computation is to search over both scales and image locations. Using different-of-Gaussian (DoG) function to detect stable keypoint that are invariant to both orientation and scale. The scale-space of a 2D image is defined as a function as shown below:-

$$\mathcal{L}(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

\* represents convolution operation,  $I(x, y)$  is the input image, and  $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$  is the scale-invariant Gaussian function. The parameter  $\sigma$  is the standard deviation of the Gaussian function and is also the scale of the feature point.

The potential feature points location in scale space is detected by using the DoG function. The DoG function,  $\mathcal{D}(x, y, \sigma)$  is calculated by using the difference between an image and another identical image with scale  $k$  times. The DoG function is defined as follow:-

$$\begin{aligned} \mathcal{D}(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= \mathcal{L}(x, y, k\sigma) - \mathcal{L}(x, y, \sigma) \end{aligned}$$

Compared a pixel to its' eight neighbours at the current scales, and its' nine neighbours top-scale and bottom-scale, so that the local minima and maxima  $\mathcal{D}(x, y, \sigma)$  can be detected. If this value is the maximum or minimum among the other pixel then this pixel is an extrema. It will be used as a SIFT feature point. The illustration will shows in Figure 2-9.

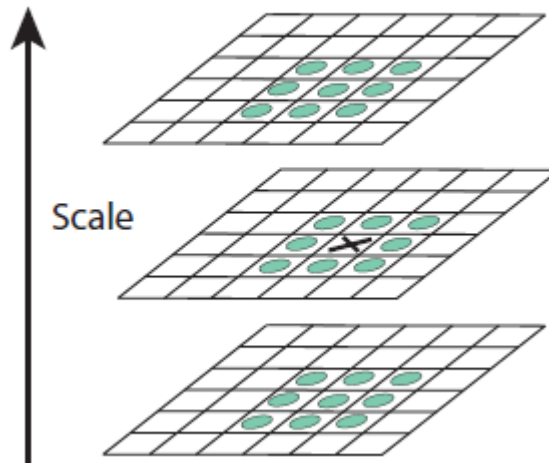


Figure 2-9 A pixel (marked with X) is compared to its 26 neighbours in 3x3 areas at the current and adjacent scales (marked with circles), so that the local minima and maxima of DoG images can be detected. (Lowe, 2004)

- **Keypoint Localization**

To filter up keypoint candidate by eliminating low contrast points and edge points, in order to improve the accuracy of feature matching and enhance the algorithm's anti-noise ability. Hence, this stage will calculate the Laplacian value for each keypoint candidate. The location of the extremum,  $\hat{x}$ , can be determined by using the following formula:-

$$\hat{x} = -\frac{\partial^2 D^{-1} \partial D}{\partial x^2 \partial x}$$

- **Orientation Assignment**

In order to make this algorithm invariance to rotation, the orientation of each keypoint have to calculate using the equation. The of orientation,  $\mu$ , and gradient magnitude,  $m$ , of  $(x, y)$  can be determined by using the following equation:-

$$\mu(x, y) = \tan^{-1}((\mathcal{L}(x, y + 1) - \mathcal{L}(x, y - 1)) / (\mathcal{L}(x + 1, y) - \mathcal{L}(x - 1, y)))$$

$$m(x, y) = \sqrt{(\mathcal{L}(x + 1, y) - \mathcal{L}(x - 1, y))^2 + (\mathcal{L}(x, y + 1) - \mathcal{L}(x, y - 1))^2}$$

- **The Keypoint Descriptor**

The keypoint descriptor generally use a 16 x 16 neighbourhood around the keypoint and divide them into 16 sub-blocks of 4x4 size. Each of them with one

8 bin orientation histogram. This will produce a feature vectors that containing 128 elements. The keypoint descriptor shows in Figure 2-10.

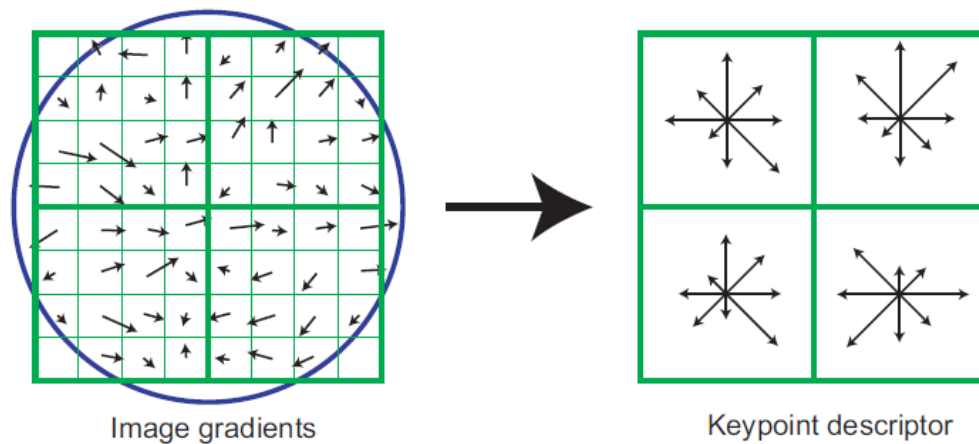


Figure 2-10 Keypoint descriptor using the gradient magnitude (Lowe, 2004)

### 2-3-2 Oriented FAST and Rotated BRIEF (ORB) algorithm

Rublee, Rabaud, Konolige and Bradski (2011) had proposed a very fast binary descriptor based on BRIEF, called Oriented FAST and Rotated BRIEF (ORB), which is resistant to noise and rotation invariant. ORB is a fusion of the BRIEF (Binary Robust Independent Elementart Features) keypoint descriptor and FAST (Features from Accelerated Segment Test) keypoint detector. One of the problem of FAST keypoint detector is that it does not calculate the keypoint's orientation element. The authors proposed the Oriented FAST keypoint detector by adding an accurate orientation component to FAST keypoint detector. Furthermore, the BRIEF descriptor performs poorly with rotation. Thus, the authors first present a steered BRIEF keypoint descriptor, according to the orientation of keypoints to “steer” the BRIEF. In order to retrieve the missing variance in steered BRIEF, and to minimize correlation between the binary test, they present a learning step to discover less correlated binary tests producing the better descriptor rBRIEF. (Rublee, et al., 2011)

In addition, computational resources are limited in mobile devices such as battery life and computation performance. In order to produce a reliable and efficient algorithm, the authors use the well-known FAST keypoint detector. FAST keypoint detector originally proposed by Edward Rosten and Tom Drummond (Rosten & Drummond, 2006), it is to detect and find the feature point from the image. However, FAST keypoint detector do not calculate the keypoints’ orientation element. Therefore, the authors proposed to use the intensity centroid (Rosin, 1999) to measure the orientation

of the keypoint. By assuming the corner's intensity is the offset from its center, and the vector can be used to impute an orientation. Rosin defining the moments as:-

$$\mathcal{M}_{pq} = \sum_{x,y} x^p y^q I(x,y)$$

according to these moments to determine the centroid as:-

$$\mathcal{C} = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

The authors define the keypoint's center as the origin "O" and construct a vector from its center, and use the origin to the centroid,  $\overrightarrow{OC}$ . So, the orientation of the patch is:-

$$\theta = \text{atan2}(m_{01}, m_{10})$$

The atan2 is the quadrant-aware version of arctan. The authors have to confirm that moments are calculated with  $x$  and  $y$  remaining within the radius  $r$  region, in order to improve the rotation invariance of that measure. They choose the  $r$  as the patch size, so that  $x$  and  $y$  run from  $[-r, r]$ . If the  $|\mathcal{C}|$  close to 0, the measure will become unstable. But in the FAST features, they found that it is rarely to happen.

They also compare this centroid method with another two gradient-based measure, which is the MAX and BIN. With both of the cases,  $\mathcal{X}$  and  $\mathcal{Y}$  gradients are computed on a smoothed image. MAX will choose the largest gradient within the keypoint patch; BIN will form a histogram of gradient directions at 10 degree intervals, and it will choose the highest bin. The test result is shown in Figure 2-11, the test is performed on recover the dataset, which is rotated and added noise. The intensity centroid performed is the best compared to histogram (BIN) and MAX method.

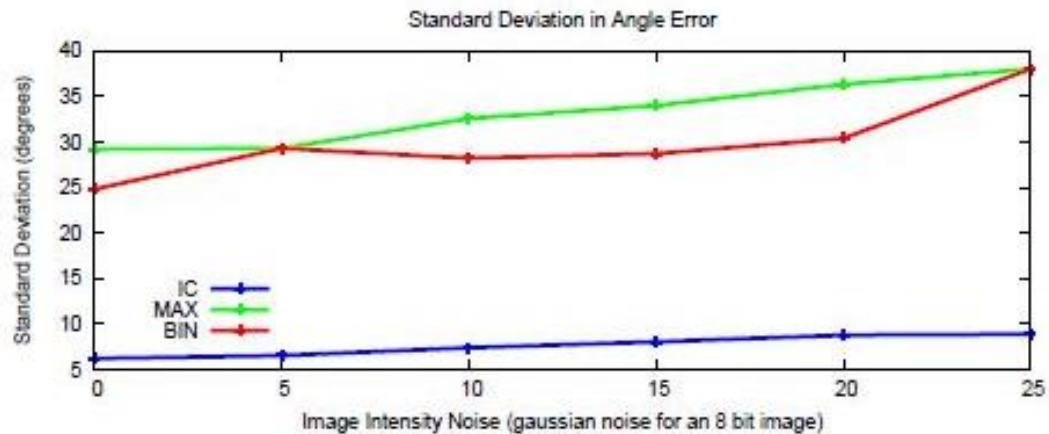


Figure 2-11 The result of comparing the intensity centroid (IC) method with another two gradient-based measures, MAX and BIN. (Ruble, et al., 2011)

## 2-4 Augmented Reality

Augmented Reality (AR) is a type of virtual reality. Virtual scene is generated with extra information together with real scene, what the users see are the composite view generated by the augmented reality system. This virtual scene is to enhance user's perception of the virtual world. The goal of augmented reality is to have a system where the virtual augmented is indifferent of the real world. The augmented reality view on a smartphone shows in Figure 2-12.



Figure 2-12 Augmented Reality View on a Smartphone

The augmented reality is required to combine multiple techniques. It requires a device equipped with a camera in order to display the real scene viewed by the user. In order to determine the orientation and location of the device, this would require several sensors of the device. The compass sensor and accelerometer is used to determine the orientation of the device and the GPS receiver of the device is used for determination

of the current geolocation. If the information is stored in separate server, an internet connection is required for the retrieval of the information data from the server.

Augmented reality applications are not only for mobile devices, it is also able to work on any other devices as long as the device has a camera attached to it. Today augmented reality is widely used in military training, robotics, medical, entertainment, education, manufacturing and many other industries. [ (Wikipedia, 2014); (webopedia, 2014)]

## **2-5 Critical Remarks of Previous Works**

### Keypoints Extraction and Description

The bottleneck of SIFT algorithm is its large amount of calculation on the entire source image, this increases the amount of time required.

The ORB algorithm is not scale invariant. If the scaling difference between the object image and input image too big, it will affect the effect of the detection. Thus, it only works well within a specific distance.

By comparing the ORB algorithm with the SIFT algorithm, although the ORB algorithm is significantly faster than the SIFT algorithm, but the matching result of the ORB algorithm is not as accurate as the SIFT algorithm. Hence, there require several improvement and testing on those algorithm before decision making process.

### Location Tracking of Mobile Devices

The Wi-Fi-based positioning is required connect to a Wi-Fi hotspots or wireless access point. In addition, the accuracy of this positioning technique is not as accurate as the GPS-based positioning.

The GPS-based positioning is unable to work under the indoor environment, because the GPS signals are unable to reach the mobile device if the device is inside a building or underground walkway. Moreover, the high battery consumption is another bottleneck of this positioning technique.

In conclusion, both of the positioning techniques have their own advantages and disadvantages. Therefore, the decision making in the development process is important since it may affect the robustness of the application.



### 3D Mobile Augmented Reality in Urban Scenes

The application “3D Mobile Augmented Reality in Urban Scenes” is implemented with client-server architecture. However, the limitation of this architecture is the speed of the mobile network. If the mobile network is not fast enough or not available within a certain region, this may affect the robustness and performance of this application. In order to solve this limitation, the main pre-processed database can put into the local device.

### Staff Outdoor Positioning System

The “Staff Outdoor Positioning System” is only use the GPS-based positioning technique to determine the staff’s location, but the limitation of the GPS-based positioning is it unable work under the indoor environment. In order to solve this limitation, this system will utilize both GPS-based and wireless network positioning technique in future work, the wireless network positioning technique will be used to determine the user’s indoor location.

## CHAPTER 3 SYSTEM DESIGN

### 3-1 System Overview

The “Smart Navigation System” consists of two modules, which is the “Building Recognition” and the “Campus Navigation”.

#### 3-1-1 Building Recognition

The “Building Recognition” flow diagram shows in Figure 3-10. There are multiple processes in this module. These includes image acquisition, grayscale conversion and image compression, export user’s coordinate and image data to Extensible Markup Language (XML) format, send XML file to server, receive JavaScript Object Notation (JSON) file from server, read building information from JSON file and overlay information on the camera view. The description of each process is given below:-

- Image acquisition

In the image acquisition process, the system will acquire image from user. The image is captured from real world scene (as shown in Figure 3-1) or uploaded using existing image (as shown in Figure 3-2), the resolution of the capture image will be 640 pixels \* 480 pixels (Width \* Height). Moreover, if the uploaded image is not within the predefined resolution, the system will resize the image according its proportion. The image must contain the building that need to recognize.



Figure 3-1 Image captured from a real world scene

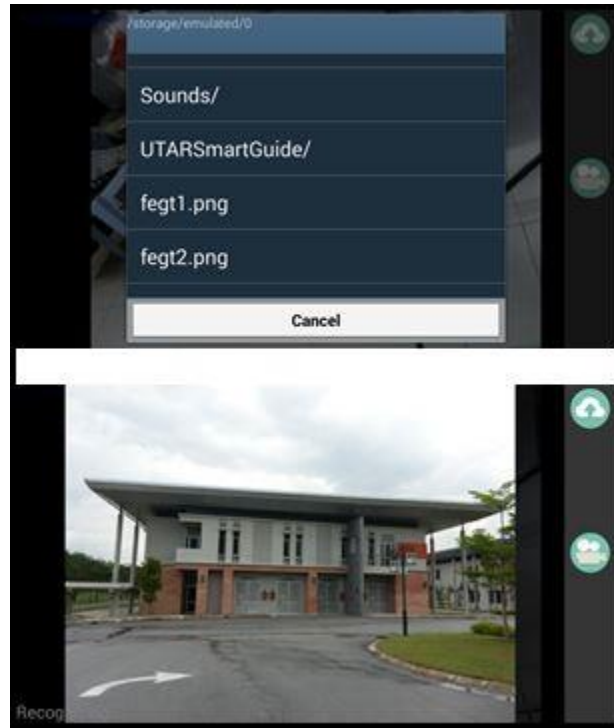


Figure 3-2 Directories dialog (Top) and uploaded image (Bottom) (Upload image process)

- Grayscale conversion and image compression

As image data is required to be sent over the internet, the size of the packet data must be minimized. In order to accelerate the data upload speed, the acquired image will undergo grayscale conversion as shown in Figure 3-3 and compression in the mobile device. Basically, the image uploaded or captured by the user consists of 4 color channels image, which contains Red, Green, Blue and Alpha (RGBA) components in each pixel and there are 8-bits per channels. Therefore, the image data is very large.

First of all, the system converts the image to grayscale because grayscale images has only 1 channel image. Hence, it would 3 times reduced in data size. After conversion, the converted image is then compressed to Portable Network Graphics (PNG) format. The reason of using PNG format is because it is lossless compression which would be able to perfectly reproduce the original data from the compressed data.



Figure 3-3 Convert the color image (Left) to grayscale image (Right)

- Export user's coordinate and image data to XML format

For the purpose of efficient recognition, user's location data will be written into XML file and is sent to server. The location data of user is retrieved by utilizing the method provided in "Google Play Service API", it uses either GPS-based positioning method or wireless-based positioning method. If a user only enables a particular positioning method, then the system will use the method to retrieve location data if the data is obtainable. However, if the location data is not available at that time, the system will prompt a message as shown in Figure 3-4 to notify the user. The user's location data would increase the efficiency of recognition as the recognition server uses the data to filter up the image for matching. This significantly increases the speed of recognition.

In the same XML file, the processed image data is appended (as shown in Figure 3-5). The reason of using XML file format is because the export method provided by Open Source Computer Vision Library (OpenCV) library and it is faster than other self-made method. However, the method is not provided in OpenCV library for android but it exists in OpenCV C++. Thus, the system have to use JNI framework in order to use the OpenCV C++.



Figure 3-4 Error message when location data is not obtainable

```

<opencv_storage>
  <latitude>4.3421757000000003e+00</latitude>
  <longitude>1.0113993080000000e+02</longitude>
  <buffer>
    137 80 78 71 13 10 26 10 0 0 0 13 73 72
    68 82 0 0 2 128 0 0 1 224 8 0 0 0 0 16
    186 131 56 0 0 32 0 73 68 65 84 120 1
    236 193 121 148 94 119 97 230 249 239
    115 239 125 151 218 85 218 101 107 177
    36 219 178 36 175 194 27 216 198 11 54
    216 128 1 7 66 135 4 146 166 195 9 233
    116 146 147 94 166 123 122 230 204 204
    233 76 58 51 57 167 147 201 116 103 235
    14 1 146 0 29 118 48 54 193 88 224 85
    222 101 217 150 45 107 183 246 82 45 82
    237 245 238 247 153 223 125 223 42 89 54
    9 51 39 255 76 74 122 63 31 221 149 203
    255 203 220 255 190 230 127 41 30 174
    109 232 144 157 86 71 134 60 250 248 85
  
```

Figure 3-5 XML file contains user's location coordinate and image data

- Send XML file to server

After the XML file is ready, the system will send the XML file over Transmission Control Protocol (TCP) socket. The reason of using TCP socket is because the TCP ensures the data to reach its destination correctly and safely. If the server socket is unable to connect after 5 seconds, the system will prompt a message as shown in Figure 3-6 to notify the user that he/she is unable to connect to the server. Once the sending process has finished, the system will start to listen for incoming message from the server.

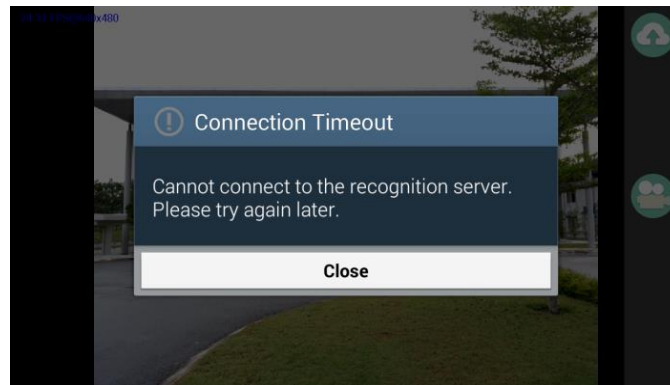


Figure 3-6 Connection timeout error message

- Receive JSON file from server

Once the server finished the recognition task, it will return the recognition results and related information back to the client-side. All the data is written in JSON file format because JSON is a light weight data exchange format and it is very easy to generate, parse and read. The received JSON file will be stored into a string variable and passed to JSON parser.

- Read building information from JSON file

If the recognition process is successful, the JSON parser method will receive an input string and it contains the name of the building and other information like the coordinate of a particular department and its name. The building name will be displayed at the bottom left of the device screen. Furthermore, the coordinate of the each department will be stored into a global object array list, this array list will be used in overlay process.

If the recognition process is unsuccessful, the JSON parse method will also receive an input string, but it contains error message received from the recognition server. This error message will be displayed at bottom left of the device screen.

In order to prevent the process from blocking the User Interface (UI) process, the process above will run in a background thread to allow the UI to be usable while the process is running. Moreover, the background thread will run every 5 seconds in order to prevent the server from being flooded. This background process includes the process of grayscale conversion and image compression, exporting user's coordinate and image data to XML format, sending XML file to the server, receiving JSON file from the server and parse JSON data.

- Overlay information on the camera view

This process is the drawing process, it will draw the box as shown in Figure 3-8 based on the data retrieved camera view by utilizing “*Core.putText()*” method and “*Core.line()*” method, both methods are provided by the OpenCV library for android. This method will automatically detect whether the global array list is empty, this array list contains the object data needed to draw on device camera view. If the global array list is not empty, the data like name of department and other information will be displayed on the camera view according to the coordinate provided in the object array list. User are able to see the information regarding the building through the device’s camera view.

This process is run on UI thread.



Figure 3-7 Building not found screenshot (Recognition result)

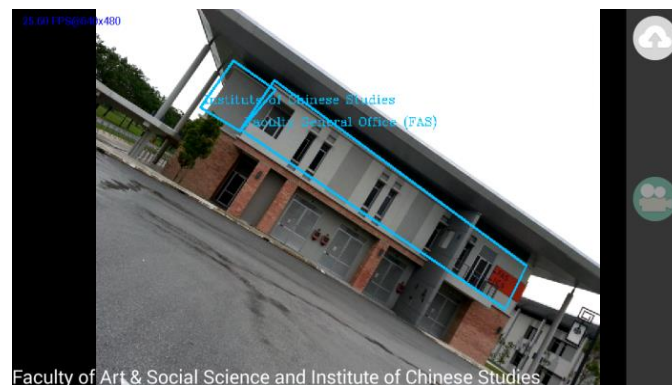


Figure 3-8 Building recognized successfully screenshot (Recognition result)



Figure 3-9 Coordinate does not exist in the server screenshot (Recognition result)

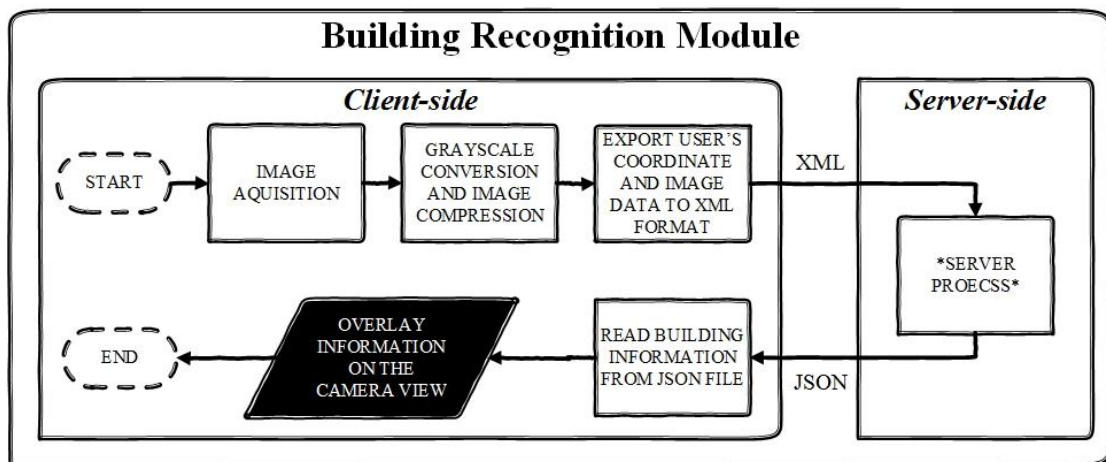


Figure 3-10 Building Recognition Module flow diagram

### 3-1-2 Campus Navigation

The “Campus Navigation” flow diagram is as shown in Figure 3-18. This module has high demand on the device resources because several of the sensors have to be queried during runtime and many of the virtual object must be drawn on the screen of the mobile device. The Metaio Android SDK is used to develop the location-based AR. There are multiple processes in this module. These includes acquiring user’s location information, retrieving POIs data from local database, displaying the POIs around the user’s current location using augmented reality techniques, retrieving the selected destination information and travel mode, switching to map route activity, get new directions Uniform Resource Locator (URL) according to current position, downloading route data from directions URL, reads route data from JSON file, draw polyline on google map, and start “Building Recognition” module. The description of each process is given below:-



- Acquire user's location information

In the initial stage of this module, the Metaio SDK native library is loaded, multiple sensor components will be created and registered and the device camera is started. After the location sensor is initialized, the system will acquire the user's location information from location provider such as GPS or wireless network. This process is crucial as this module is unable to work as expected if user's location information retrieval has failed.

- Retrieve POIs data from local database

The data of the POIs will be loaded from local SQLite database before being displayed to the user. This database is a very light weight database, the data stored includes the POIs information, names of related images and its GPS coordinate. The Entity Relational Diagram (ERD) and data dictionary of the database are as shown in Figure 3-11 and Table 3-1 respectively.

As creating the database during runtime would be an expensive process because there is large bulk of data. Therefore, the system will utilize a preloaded database and load it during the system first run. To create the preloaded database, a program called "SqliteBrowser" to create and input data was used. After the database is created, the database file is copied to assets folder and the database is loaded during the system's first run. The loading process is handled by the "SQLiteHelper" class. In order to update the database, the latest database would be required to replace the previous database in assets folder and named differently. After replacing the database in the assets folder, the database name in "SQLiteHelper" also need to be changed according to the latest database name. In order to prevent the preloaded database from loading every single time, the system uses "SharedPreferences" to achieve the "first run" check.

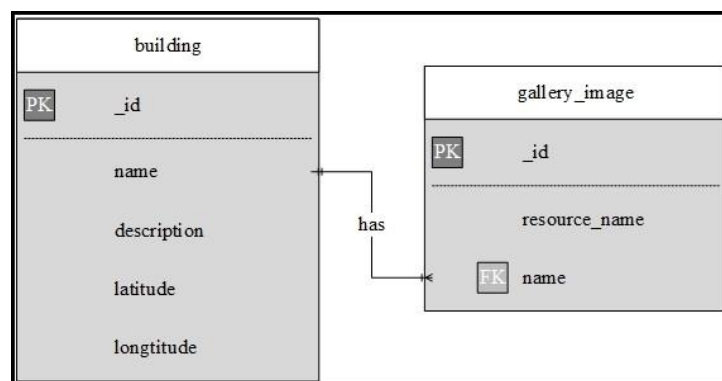


Figure 3-11 Entity Relational Diagram (ERD)

Table 3-1 Data Dictionary

Table Name	Attribute Name	Content	Type	Format
building	id	Building id	Integer(5)	99999
	name	Building name	Varchar(45)	xxxxxx
	description	Building description	Varchar(500)	xxxxxx
	latitude	Building latitude	Double(10, 6)	99.999999
	longitude	Building longitude	Double(10, 6)	99.999999
gallery_image	id	Gallery image id	Integer(5)	99999
	resource_name	Gallery image resource name	Varchar(45)	xxxxxx
	name	Building name	Varchar(45)	xxxxxx

- Display the POIs around the user's current location using augmented reality techniques

Once the user's current location information is received, the system will overlay POIs geometry around the user's current location on device's camera view as shown in Figure 3-12. In order to enhance the user's sensory perception, multiple sensor components will be utilized. For instance, the accelerometer, the compass sensor and the location receiver is use to determine the rotation, orientation and location of the device. These sensors' data is to ensure the POIs is within the field of view.

Moreover, the radar on the top right is clickable. If the radar is clicked, a seek bar will appear, user can adjust the radius of radar by moving the point of the seek bar (as shown in Figure 3-13). POI will be filtered according to the distance set by the user on the seek bar. The default radius of radar is 10 kilometers (KM) around the users.

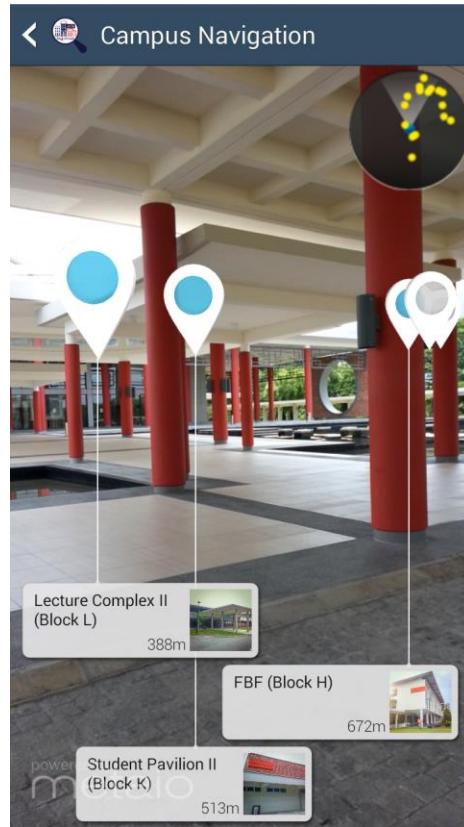


Figure 3-12 POIs around the user's current location

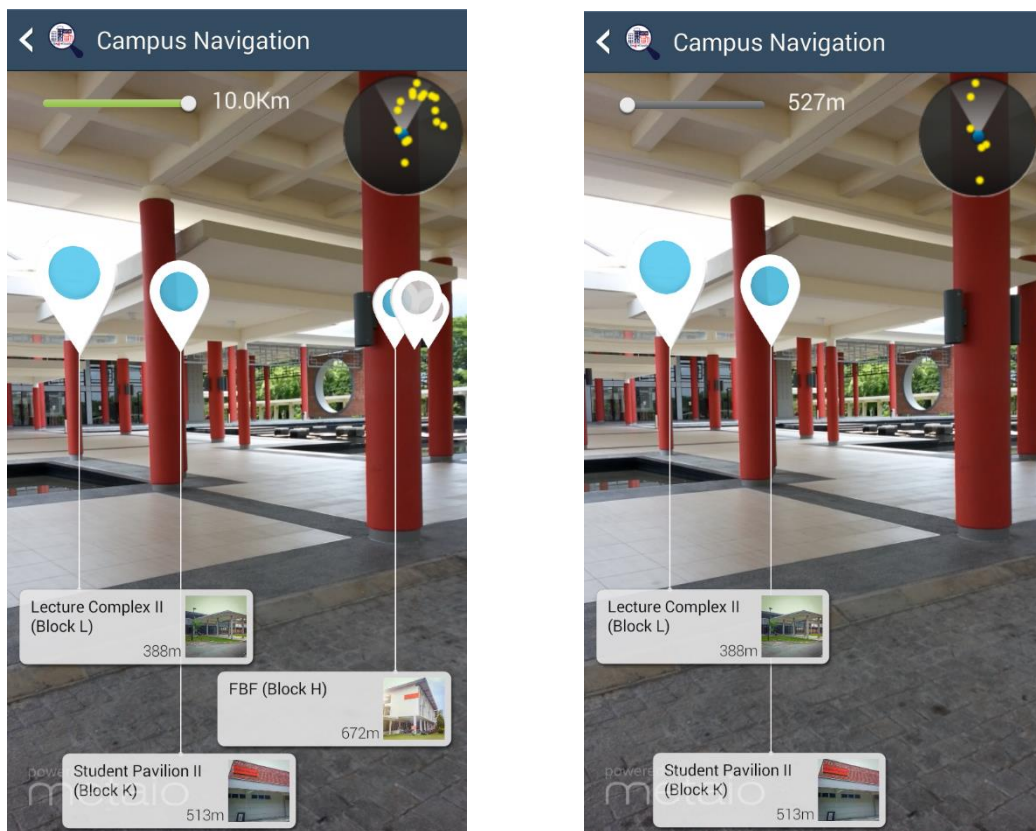


Figure 3-13 Adjust the radar radius: before (Left) and after (Right)

- Retrieve selected destination information and travel mode

All the POIs displayed on camera view is selectable. If user select a particular POI, the POI details dialog will show up as shown in Figure 3-14 (Left). This dialog consists of the POI information, distance from current location, image gallery, route button and close button. If a user clicks on the route button, another travel mode dialog will show up as shown in Figure 3-14 (Right) which contains 3 types of travel mode like driving, cycling and walking. User can select a travel mode and click start navigation button to proceed to navigation process. In the meantime, system will pass the selected destination name, coordinate and travel mode to “Map Route” activity.

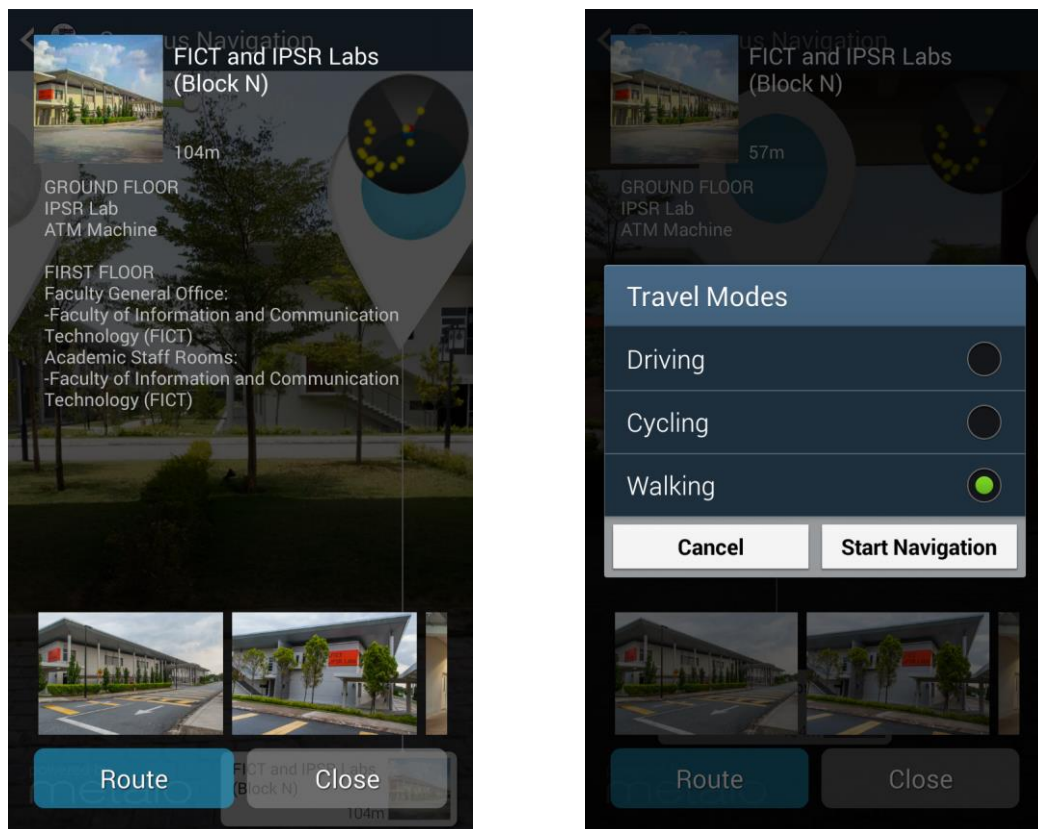


Figure 3-14 Selected destination information dialog (Left) and travel modes dialog (Right)

- Switch to “Map Route” activity

Beginning of the “Map Route” activity, the system will retrieve the data from previous activity (“Campus Navigation” module). Next, Google Map will be initialized and ready to display navigation route. To perform the navigation, there are several background processes that are running. User’s location information will be updated in an interval of time. These background processes

includes get new directions URL according to current position, download routes data from directions URL and read routes data from JSON file. These process will run recursively until the destination is reached. Moreover, if the user is 50 meters approaching to the destination a pop-up message will appear and propose user to switch to “Building Recognition” module (as shown in Figure 3-15 (Left)). If user decline the first propose, system will notify the users in every 15 meters until the user reaches the destination.

The following part will further discuss the details of each process.

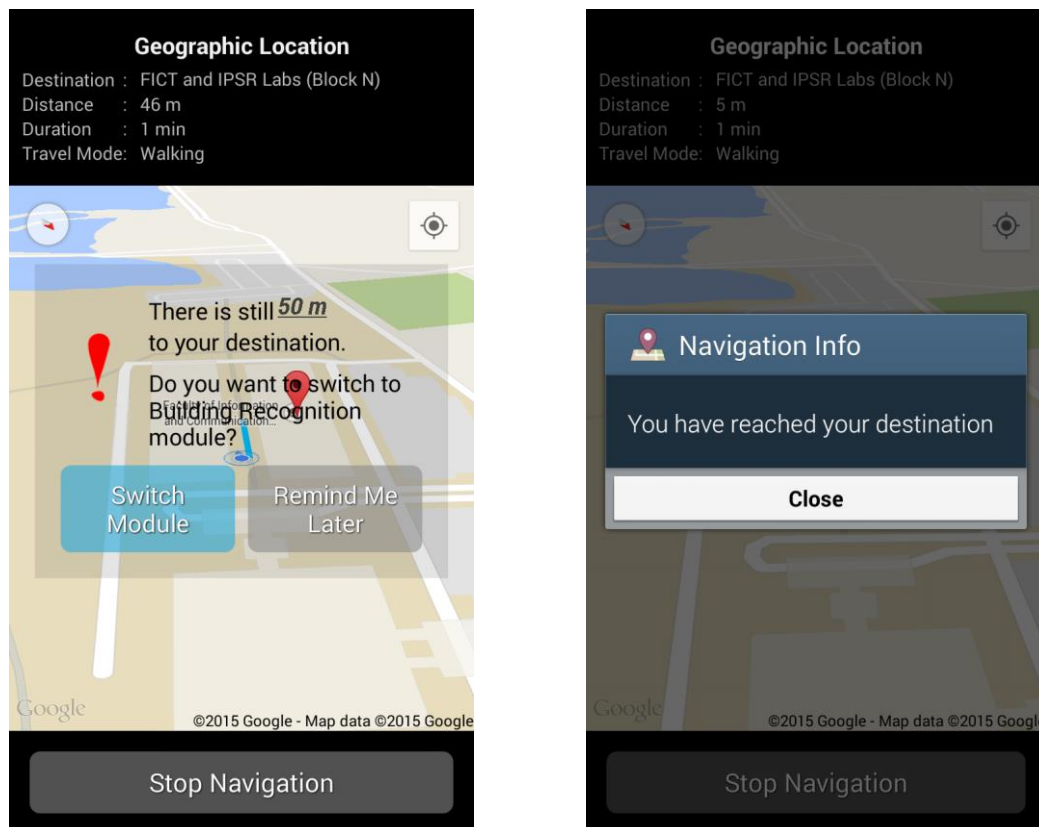


Figure 3-15 Near to destination pop-up message (Left) and destination reached dialog (Right)

- Get new directions URL according to current position

A method called `getDirectionsURL()` will receive the user’s current coordinate and destination coordinate as parameters. The purpose of this method is to build the URL to the web service (Google Directions API) and access the interface to get the routes details. It provides in both XML and JSON file format. JSON file format was selected in this part of the module. The routes details in JSON file format as shown in Figure 3-16.



Figure 3-16 Routes details in JSON file format

- Download routes data from directions URL

Due to the fact that JSON is a light weight data exchange format and it is very easy to generate, parse and read. Once the route details in JSON data format is downloaded from the directions URL, the system will pass it on to JSON parser in order to get the routing details like distance, duration and the path to the destination.

- Read routes data from JSON file

This process will constructs a new instance of “DirectionsJSONParser” and use the method provided in this class to read the routes data from JSON file. The data returned from the method is a list of “HashMap” list.

- Draw polyline on google map

In this process, the system will draw route on the map as shown in Figure 3-17 according to the route data retrieved from the directions API. The “HashMap” is a collection class, it is a data structure based on hashing, which is able to store object as (<key (K)>, <value (V)>) pair. The advantage of using “HashMap” is that the system can retrieve the object at a constant time i.e.O(1), but the precondition is that the key must be known in advance.

This process is run on UI thread.

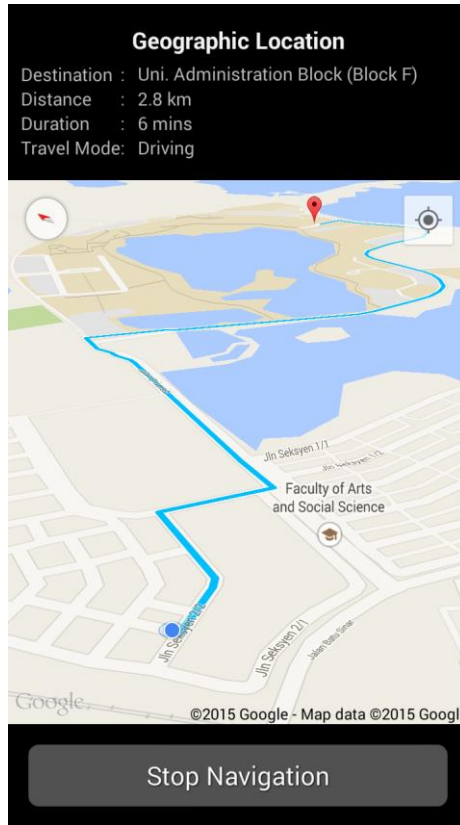


Figure 3-17 Route drawn on google map

- Start “Building Recognition” module  
 The system will switch from “Map Route” activity to the “Building Recognition” module.

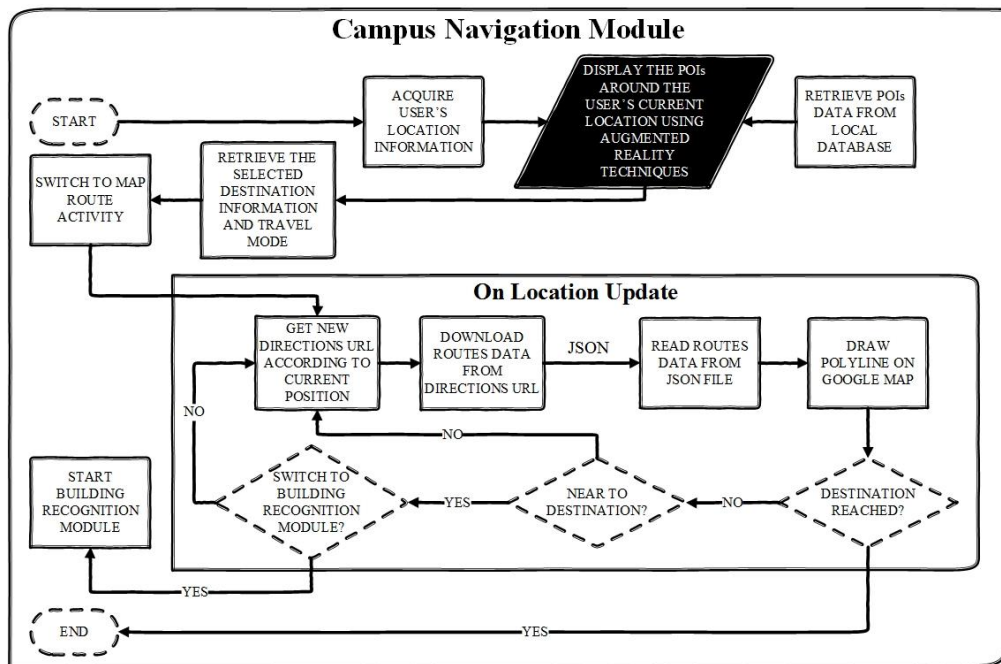


Figure 3-18 Campus Navigation Module flow diagram

## CHAPTER 4 METHODOLOGY AND TOOLS

### 4-1 Methodology

To develop a successful system, choosing an appropriate methodology is important. Methodology provides the basic guidelines that will guide the developer to accomplish the project task. In this section, what is the best methodology need to be applied on this project will discussed.

This project involves heavy user interaction thus is more suitable to use prototyping approach. A prototype will be developed based on the known requirements of the system, so that with the prototype the client have a basic idea of how the completed product will look like. The interactions with the client can have a better understanding the requirements of the desired system. Normally, the prototype is not a complete system and there are some of the features that are not built in the prototype. The objective of the prototype is to deliver a system that can provide basic functionality. The prototyping methodology shows in Figure 4-1.

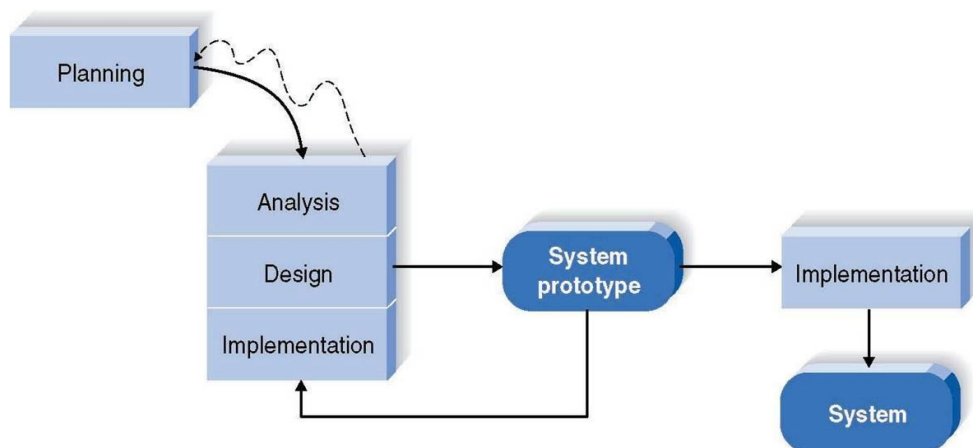


Figure 4-1 Prototyping Methodology

The prototyping methodology has the ability to go back the previous stages such as planning, analysis, design and implementation. Hence, the error and missing functions can be easily adapted into the methodology. The process continues in a cycle until the project has reached satisfaction and achieve the expectation. Prototyping methodology is commonly applied where there are high user interactions required so that feedback from the client can be used to constantly work on the system.



## 4-2 Tools to Use

### 4-2-1 Hardware

The hardware used in this projects are:-

- Smart device with Android Operating System (OS)

This smart device is used for test and install the mobile application.

Table 4-1 Smartphone Hardware Specifications

Operating System	Android 4.2.2 Jelly Bean
CPU	ARM Cortex-A9, 1200MHz, Cores: 2
GPU	ARM Mali-400 MP4, 266MHz, Core: 4
RAM	1GB
Wi-Fi	a, b, g, n, Dual band, Wi-Fi Hotspot, Wi-Fi Direct
Positioning	GPS, A-GPS

- Personal Computer

As an interface to program the mobile application.

Table 4-2 Personal Computer Hardware Specifications

Operating System	Windows 8.1 Pro 64-bit
CPU	Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz
GPU	NVIDIA GeForce GT 650M
RAM	8GB

### 4-2-2 Software

The software used in this projects are:-

- Eclipse IDE

Eclipse Integrated Development Environment (IDE) is used to write the application for the smart device.

- Android SDK

The Android Software Development Kit (SDK) is used to develop the application based on Android platform, which includes a comprehensive set of development tools. These include a mobile phone emulator based on QEMU, libraries, debugger, documentation, tutorials, and sample code.

- **Android ADT**  
The Android Development Tools (ADT) is a plugin for the Eclipse IDE. It is used to extend the functionality of the development environment with the ability to set up Android application.
- **Java Native Interface (JNI)**  
JNI is a programming framework and is a part of Java SDK. It enables the Java code to interact with the native code (written in C/C++).
- **OpenCV for Android**  
The OpenCV is an open source library for computer vision, machine learning and image processing. It has multiple interfaces such as Python, C, C++ and also the Java and supports Linux, Mac OS, Windows, Android and iOS. The “Smart Navigation System” will develop by using the Java interface.
- **Metaio SDK**  
The Metaio SDK is used for programming PC, mobile, web and custom offline augmented reality applications. It is currently supported on iOS, Android and Windows. It released the free version of SDK with a Metaio watermark, if want to remove the watermark have to upgrade to BASIC / PRO license.
- **Google Play services SDK**  
It is required to setup the Google Play services SDK in order to use the Google Play Services API.
- **Google Location Services API**  
The Location API provides the functions required in develop location-aware applications such as determine the user’s geolocation, it simplify the development process.
- **Google Maps Android API**  
The Google Maps Android API provides the function to add maps based on the Google Maps data to the application. It will automatically handles access to Google Maps servers, response to map gestures, map display and data downloading. It also provides the functions to manipulate the user’s view of a specific map area.
- **Google Directions API**  
The Google Directions API is a web services that provide HTTP interfaces, developers can access the routing algorithms through simple HTTP request.

- Windows 8.1

The Windows 8.1 operating system of the computer is used to connect the smart device and program the mobile application.

### 4-3 Requirement

In order to run this applications, the mobile device have to fulfil several requirements. These requirements can categorised into two type which are hardware requirement and software requirement.

- Minimum hardware requirements
  - CPU that supports x86 or ARMv7 (with NEON) architecture(s)
  - OpenGL ES 2.x
  - Camera that can deliver at least QVGA (320x240) preview resolution
  - Display with at least HVGA (480x320) resolution
  - Accelerometer/Gravity sensor
  - Magnetic sensor
- Minimum software requirements
  - Android 2.3.3 (API Level 10) or above
  - Location service
  - Internet connectivity

Before the system installation, the system is required to access the following permissions:-

- Take pictures and videos.
- Approximate location (network-based) and precise location (GPS and network-based).
- Modify or delete the contents of SD card and read the contents of SD card.
- Read google service configuration.
- Full network access and view network connections.
- Change the audio settings.

On the other hand, there are additional requirements for the “Building Recognition” module. One of the requirements, is such that the captured or uploaded image dimension cannot be larger than 640 pixels \* 480 pixels (Width \* Height). The first reason is because the recognition server is only capable to recognize the image within

this dimension currently. The second reason is because the efficiency of image data transmission will be affected if the dimension is too large. Besides, in order to use the OpenCV library the mobile device is required to download the “OpenCV Manager” package from “Google Play” store.

#### 4-4 Verification Plan

This guidance system must be able to provide reliable and accurate information. However, it may be affected by some coding or network errors. Hence, there are some verification steps to ensure the system accuracy and reliability. Few verification steps are shown as follow:-

- The connection between client and server

Table 4-3 Verification P1

Procedure Number	P1
Method	Testing
Applicable Requirements	The application is using client-server architecture
Purpose / Scope	To test the connection between the client and server
Precautions	None
Special Conditions / Limitations	1. Server is not operating 2. Internet connection unstable
Equipment / Facilities	Smartphone
Data Recording	None
Acceptance Criteria	The data able to send through the internet and receive the respond from the server
Procedures	1. Send image data to server 2. Through receiving a response from server to verify the connection between client and server
Troubleshooting	1. Review the source code 2. Check internet connection 3. Repeat the procedure
Post-Test Activities	None

- The best route provided by the “Campus Navigation”

Table 4-4 Verification P2

Procedure Number	P2
Method	Testing
Applicable Requirements	The application able to calculate the route to the selected destination
Purpose / Scope	To test whether the system able to help the user avoid unnecessary route
Precautions	None
Special Conditions / Limitations	1. Destination does not exists
Equipment / Facilities	Smartphone
Data Recording	None
Acceptance Criteria	The application able to calculate the best route and help user to avoid unnecessary route
Procedures	<ol style="list-style-type: none"> <li>1. Select a destination</li> <li>2. Select a travel mode</li> <li>3. Start navigate</li> </ol>
Troubleshooting	<ol style="list-style-type: none"> <li>1. Review the source code</li> <li>2. Repeat the procedure</li> </ol>
Post-Test Activities	None

## CHAPTER 5 IMPLEMENTATION AND TESTING

### 5-1 Overview

In order to test the robustness of the system, there are some testing scenario that had been tested and verified. These test case is stated as below:-

- The connection between client and server.
- The best route provided by the “Campus Navigation”.

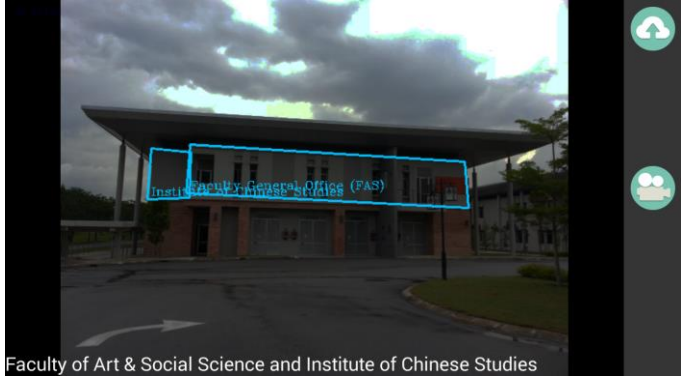


These testing process is to ensure the system has the ability to provide reliable and accurate information. To test the “Building Recognition” module by using both image acquisition method: capture real-time image and upload existing image. To test the “Campus Navigation” module by testing every travel mode once. The travel mode includes: driving, cycling and walking.


### 5-2 The connection between client and server

The test result of connection between client and server is as shown in Table 5-1.

Table 5-1 Connection between client and server testing results

Screenshot	Return result
<p data-bbox="300 1173 619 1207"><u>Capture real-time image</u></p> 	<p data-bbox="1023 1173 1385 1207"><b>Connection successful:</b></p> <p data-bbox="1023 1227 1385 1317">The recognition result is overlaid on camera view.</p>

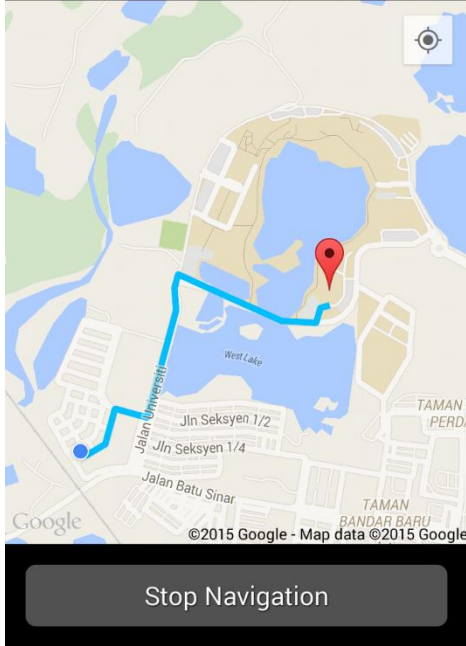
<p><u>Upload from existing image</u></p>  <p>Faculty of Art &amp; Social Science and Institute of Chinese Studies</p>	<p><b>Connection successful:</b></p> <p>The recognition result is overlaid on uploaded image.</p>
<p><u>Capture real-time image</u></p>  <p>The building are not found</p>	<p><b>Connection successful:</b></p> <p>There is no recognition result (The building is not found).</p>
<p><u>Upload from existing image</u></p>  <p>The building are not found</p>	<p><b>Connection successful:</b></p> <p>There is no recognition result (The building is not found).</p>
<p><u>Capture real-time image</u></p>  <p>Connection Timeout Cannot connect to the recognition server. Please try again later. Close</p>	<p><b>The connection failed:</b> The connection timed out.</p> <p><b>Reason:</b> The recognition server is not in operation.</p>

<p><u>Upload from existing image</u></p>  <p>Connection to the server timed out</p>	<p><b>The connection failed:</b> The connection timed out.</p> <p><b>Reason:</b> The recognition server is not in operation.</p>
--	--

### 5-3 The best route provided by the “Campus Navigation”

The test result of best route provided by the “Campus Navigation” is as shown in Table 5-2.

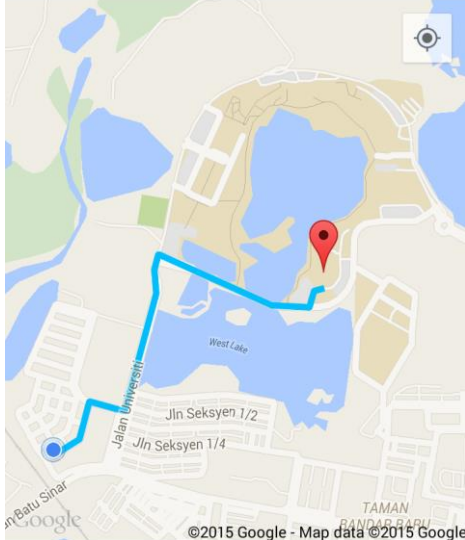
Table 5-2 Best route provided by the “Campus Navigation” testing results

Screenshot	Return result
<p><b><u>Travel mode: Driving</u></b></p> <div data-bbox="300 1032 767 1205"> <p><b>Geographic Location</b>            Destination : Heritage Hall (Block A)            Distance : 1.8 km            Duration : 4 mins            Travel Mode: Driving</p> </div> 	<p><b>Get route successful:</b> The best route of driving mode as shown in maps.</p>



**Travel mode: Cycling**

**Geographic Location**  
Destination : Heritage Hall (Block A)  
Distance : 1.8 km  
Duration : 4 mins  
Travel Mode: Cycling



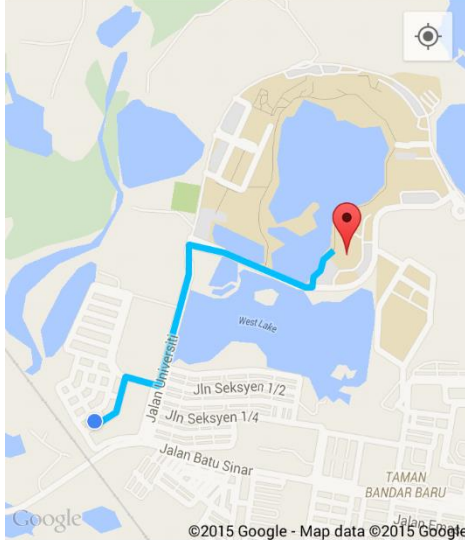
©2015 Google - Map data ©2015 Google

Stop Navigation

**Get route successful:** The best route of cycling mode as shown in maps.

**Travel mode: Walking**

**Geographic Location**  
Destination : Heritage Hall (Block A)  
Distance : 1.8 km  
Duration : 22 mins  
Travel Mode: Walking



©2015 Google - Map data ©2015 Google

Stop Navigation

**Get route successful:** The best route of walking mode as shown in maps.

## CHAPTER 6 CONCLUSION

### 6-1 Conclusion

The problem and motivation of this project is that most of the existing navigation system are unable to provide routes accurately as well as further information of the building within a region such as campus, shopping mall, hospital and etc. Furthermore, the method of by typing words is sometimes inconvenient, an advanced search method by using images can be used to substitute the method of typing words into the search engine.

In a nutshell, the main focus of this project is to develop an informative, reliable and precise navigation system, this navigation system is developed on android platform. The purpose of this system is to solve the problems face by UTAR freshman and visitors during the adaptation of unfamiliar environment as most of the existing navigation system are unable to provide routes precisely and further information of the building within a region.

In the context of this project, a new mobile application for the Universiti Tunku Abdul Rahman is introduced. This “Smart Navigation System” are consists of two modules, which is the “Building Recognition” and the “Campus Navigation”. The “Building Recognition” module is able to recognize a building within an image and return the information regarding the recognized building. The “Campus Navigation” module is able to get the directions to the destination.

### 6-2 Limitation

This final system had performed up to satisfaction. However, this system has the following drawbacks:-

- Internet connection  
This system is fully rely on the internet connection. The system is unable to work as expected if there is failed to connect to internet.
- Speed of recognition  
Because the recognition task in “Building Recognition” module is required to interact with the server. If the network speed is not fast enough, the time cost for the recognition process would increase.

### 6-3 Future Work

In order to produce a more informative yet user-friendly application and have the ability to provide precise guidance, some features and functions mentioned below should be enhanced in future developments.

- Implement AR feature in “Map Route” activity

In order to prevent users from looking at the device screen while driving which could actually cause a fatal accident, a head-up display can be implemented. The head-up display means that user can see through the device screen via camera view, a 3-D arrow will be overlaid on the camera view. This 3-D arrow is able to point towards the direction the user is supposed to go (as illustrated in Figure 6-1). Therefore, users can keep the head up while using the navigation system. The map is minimized into a mini map on the bottom right and the mini map can be maximize when click.



Figure 6-1 Screenshot of 3-D arrow pointing to the route direction with the minimized map

- Make the building recognition details clickable

As the recognition result screenshot shown in Figure 3-8, the name of department on camera view according to the provided coordinate was just overlaid and it is not possible to overlay all information on the device screen because the device screen size is limited. Therefore, the area of rectangle (department) should be made clickable and a dialog pop-up which should be able to display information regarding the selected department.

- Able to calculate route itself without using the google directions API

In this project, the directions between the locations is calculated by the google directions API which is a web services that provide HTTP interfaces, thus the system is required to have internet connectivity in order to access the routing algorithms.

The recommended solution is to develop a routing algorithm that is able to compute the directions between the locations. Therefore, the navigation system can be work even though offline.

## REFERENCE

Bangare, P. et al., 2014. The Campus Navigator: An Android Mobile Application. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(3), pp. 5715-5717.

Developers, A., 2014. *Location Strategies*. [Online] Available at: <http://developer.android.com/guide/topics/location/strategies.html#Updates> [Accessed 15 August 2014].

Griffin, D., 2011. *PocketGPSWorld*. [Online] Available at: <http://www.pocketgpsworld.com/howgpsworks.php> [Accessed 12 August 2014].

Leow, V. L. D. et al., 2011. Staff Outdoor Positioning in Large Area Campus using GPS Enabled Phone, Google Map and Mobile Network. *Symposium on Information & Computer Sciences*.

Lowe, D., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), pp. 91-110.

physics.org, 2014. *physics.org*. [Online] Available at: <http://www.physics.org/article-questions.asp?id=55> [Accessed 13 August 2014].

Rosin, P., 1999. Measuring Corner Properties. *Computer Vision and Image Understanding*, 73(2), pp. 291-307.

Rosten, E. & Drummond, T., 2006. Machine learning for high-speed corner detection. *9th European Conference on Computer Vision*, Volume 1, pp. 430-443.

Rublee, E., Rabaud, V., Konolige, K. & Bradski, G., 2011. ORB: an efficient alternative to SIFT or SURF. *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564-2571.

Shah, D. & Shah, K., 2014. Wi-Fi Based Positioning System.

## REFERENCE

Takacs, G., El Choubassi, M., Wu, Y. & Kozintsev, I., 2011. 3D Mobile Augmented Reality in Urban Scenes. *2011 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1-4.

webopedia, 2014. *webopedia*. [Online]  
Available at: [http://www.webopedia.com/TERM/A/Augmented\\_Reality.html](http://www.webopedia.com/TERM/A/Augmented_Reality.html)  
[Accessed 10 November 2014].

Wikipedia, 2014. *Wikipedia*. [Online]  
Available at: [http://en.wikipedia.org/wiki/Augmented\\_reality](http://en.wikipedia.org/wiki/Augmented_reality)  
[Accessed 10 November 2014].