

|  |  |
| --- | --- |
| Project Title | A Mobile Information Kiosk with Navigation of Buildings using AR |
| Study Program | Internet Technology |
| Supervisor | Li Tak Sing |
| Student ID | 11658374 |
| Student Name | Chan Kwok Chi |
| Date | Jan 21, 2019 |
| Grade |  |

Leave For Blank

Contents

[Section 1. Problem Definition 1](#_Toc535794061)

[1.1 Introduction 1](#_Toc535794062)

[1.2 Project Aim 1](#_Toc535794063)

[1.3 Project Objectives 1](#_Toc535794064)

[1.4 Value Proposition 2](#_Toc535794065)

[Section 2. Literature Reviews 2](#_Toc535794066)

[2.1 Problem Analysis 2](#_Toc535794067)

[2.1.1 Data presentation 2](#_Toc535794068)

[2.1.2 Navigation 3](#_Toc535794069)

[2.1.3 Problem analysis conclusion 3](#_Toc535794070)

[2.2 Related Technologies 4](#_Toc535794071)

[2.2.1 Brief description of the proposed system 4](#_Toc535794072)

[2.2.2 Related technologies and difficulties 4](#_Toc535794073)

[2.3 Existing Solutions 5](#_Toc535794074)

[2.3.1 AR tools 5](#_Toc535794075)

[2.3.2 Camera-based positioning with different references 5](#_Toc535794076)

[Section 3. Methodologies 6](#_Toc535794077)

[3.1 Requirements and technologies for this project 6](#_Toc535794078)

[3.2 Design of the proposed system 6](#_Toc535794079)

[3.2.1 User input and information output 6](#_Toc535794080)

[3.2.2 System overview 7](#_Toc535794081)

[3.2.3 Detailed design of the navigation 9](#_Toc535794082)

[3.2.4 Detailed design for the object recognition 11](#_Toc535794083)

[3.2.5 mapping between live captures and floor plan 11](#_Toc535794084)

[3.3 Evaluation design 11](#_Toc535794085)

[References 13](#_Toc535794086)

[Appendix A. Progress Report 15](#_Toc535794087)

[A. Overview of project progress 15](#_Toc535794088)

[B. Tasks status tracking 15](#_Toc535794089)

[C. Key success and failures 16](#_Toc535794090)

[Success 16](#_Toc535794091)

[Failures 16](#_Toc535794092)

[D. Revised project plan 17](#_Toc535794093)

# Problem Definition

## Introduction

In modern cities, more and more public places (e.g. a big shopping mall, university, library etc.) have been being built up for bringing convenience to people’s life. Although lots of services can be provided with a large size of building, people tend to lose their way when walking within it. This problem includes two aspects:

First, the interior structure of a building is complicated with large size and a lot of facilities or services. Even for a worker of that building, it still has to consume lots of time to remember the position and other detailed information of those facilities and services.

Second, apart from those static objects, there are many dynamic objects/events within a building. Visitors who don't come to the building frequently will not know the changes of those objects/events.

These two aspects are all pointing to a fact that visitors have a delay in information acquisition. Thus the author believes that an information kiosk using AR to provide information to users instantly can solve the problem. The survey conducted by Olsson et al. [1] also shows that users are requiring an AR navigation app to solve the problem.

## Project Aim

The aim of this project is to build a mobile AR application that provide instant information service and navigation guidance. Users can gain information just through scanning target objects by cameras on their devices. Those information includes backgrounds of static objects, time schedule of dynamic objects such as activities or other useful for exploration. Navigation guidance will give suggested directions that could take users to reach the target place. It hopes that this app could help people explore a building efficiently.

## Project Objectives

1. Design and develop a mobile app with AR.
   1. Design GUI and components for the app.
   2. Design and implement AR interaction.
2. Design and develop a server-side program.
   1. Develop a function for object recognition
   2. Design and implement an algorithm for navigation
   3. Design and Implement communication between server and app.
3. Design database schemas.
   1. Design a schema for floor plan of a building.
   2. Design a schema for storing information of indoor objects/events
4. Design information input interface
   1. Design a web page GUI
   2. Implement database operation through the web page
5. Test and evaluation
   1. Test and debug the whole system
   2. Experiment and evaluation

## Value Proposition

Users will be the first group benefits from this system in terms of efficiency when exploring a building.

With this system, building manager can save money as they don't need to employ workers to give information guidance to visitors.

Although AR and indoor positioning technologies have been developed rapidly in recent years, there is still no general app that taking advantages of those technologies for a building exploration guidance in Hong Kong. This proposed system can fill the gap.

# Literature Reviews

## Problem Analysis

To achieve the goal of quick exploration for an unfamiliar building, current position of users as well as the information of indoor objects should be provided to users. Thus this part will be elaborated in two sub-parts: data presentation and navigation.

### Data presentation

An effective GUI requires a proper form of information display which could help users strongly feeling the sense of tie between a real object and that virtual one on GUI. In conventional information application (e.g. google map), text information often displayed onto a 2D map or along with a picture. This kind of GUI some time is difficult for users to identify the related object in real world. This problem can be solved if the software includes real vision into the its GUI. By employing AR technology for data presentation, people can gain what they want to know directly from real vision on camera live capture. This way is more natural in sense therefore user experience is improved.

An introduction of AR will be in section 2.2.2.

### Navigation

With the consideration of using AR user interface just mentioned, navigation function of the information kiosk should at least fulfill two requirements: keep track of users’ real-time perspective and display navigation information according to users’ current position. Due to the poor performance of GPS in indoor environment, professionals employ other technologies to achieve the positioning goal. They can be roughly classified into 3 groups [2]: 1) Signal-based positioning, such as WIFI, Bluetooth and cellular wireless connection signal. 2) Sensor-based positioning, such as IMU and camera (often related to AR). 3) Combination of both, such as AR+IMU, WIFI+AR. Although the first kind of the technology can even reach high accuracy of 1 - 2 cm according to the evaluation done by Khoury and Kamat [3] , it is usually expensive in deployment of devices, neither keeps track of users’ orientation nor provides well user experience. The rest two kinds of positioning technologies are the result that researchers have been trying to take advantages of advances of recent smartphones, which have multiple built-in sensors. Camera-based positioning technique is intrinsically compatible with AR and now becoming dominant in this filed [4]. It has its own advantages such as low-cost in deployment and ability of keeping track of users’ orientation. Note that in applications of the third kind of techniques, sensors or signals are often used to calibrate the deviation from the actual position of devices and it is not much different from the first two.

### Problem analysis conclusion

Conclusion for discussion above is that camera-based navigation can seamlessly cooperate with AR to provide a better user experience in an information app for a building. It can be combined with other sensors or signals for calibration.

## Related Technologies

This part will begin with a brief description of the proposed system, then followed with introductions of related technologies as well as difficulties of implementation and existing solutions for those difficulties.

### Brief description of the proposed system

The survey in [5] summarized the basic tasks of an AR navigation system. Thus this paper designed three main components for the proposed system: 1) a Database. It stores feature patterns (also called markers) that extracted from real vision by data collectors. 2) A client app. It provides AR user interface, communicates with server. 3) A server-end program. It communicates with clients and retrieve data from database. Obviously, the second component needs a series of AR tools to detect environment and to draw virtual objects; the third component needs an algorithm for estimating users’ position from inputted photos or other optional data.

Details of the system will be elaborated in section 3.

### Related technologies and difficulties

Just as mentioned in last section, related technologies are AR and camera based positioning.

The concept of Augmented Reality(AR) technology is to provide a graphical interface to users with real world perspective superimposed upon by or composited with virtual objects in real-time [6]. It could be used to supplement information, which can not be obtained from general perceptions, to objects in real world. The difficulties are that it requires robust graphic recognition and proper placement of virtual objects which result in an illusion that those objects are naturally parts of the real environment.

Camera-based positioning means that users’ positions are estimated from photos captured by their cameras. Note that this technology is built up upon the graphic recognition which is also the base of AR application. The difficulties are that it requires well references planning and high performance in positioning computation.

## Existing Solutions

### AR tools

Recent years, several AR toolkits have been introduced to software developers. Among them ARKit [7] and ARCore [8] are useful for mobile apps development. ARKit is introduced by Apple company and packages device motion tracking, real world understanding and other AR related functions. However, it can only be applied to apple’s apps. ARCore is introduced by google company and serves the similar functions as ARKit does. It is a multi-platform framework.

### Camera-based positioning with different references

The main difference of each existing solutions is the type of markers it references to[4]. For convenience, this paper classifies these solutions into 4 groups by types of references:

1) Use references from artificial markers. Since the patterns of artificial markers such as barcodes are really distinguished from nearby environment, they are much easier to be recognized, making the system stable and robust. However, artificial markers are aesthetically defective and the installment of them may be cumbersome.

2) Use references from samples of images. Just as described in [9] and [10], the samples can be a set of pictures taken for designated markers or videos taken for the scene by data collectors walking through the whole environment. The system of this kind of solutions is simpler than the rest others but not robust.

3) Use references from extracted features. Just as described in [11] and [12], those features can be physical points on surface of indoor static objects (e.g. floor-to-wall transitions). It often includes feature points matching calculation that increasing complexity of the system, but in the meantime bring robust and efficiency.

4) Use references from 3D model. Just as described in [13]. Since the constructing process of a 3D model is really time consuming, this paper will not take this solution into consideration.

For better user experience, this paper is going to take advantages of the second and the third solutions for constructing the proposed system.

# Methodologies

In principle, the basic methodology for this project is Build-and-Test. Hence this section will first introduce the requirements and technologies for this project, then describe the design for the proposed system and finally the evaluation plan for the system.

## Requirements and technologies for this project

To achieve the aims of the proposed system, two key technology requirements should be fulfilled. One is the image recognition technology, and the other is the AR technology. The image recognition technology will be used through the whole system as it is essential for camera-based navigation and AR. Once the real object is recognized, related information will be bound to the objects in live captures using AR technology in order to enhance user experience.

It is also going to use some essential tools for system prototype construction. It contains:

1) ARCore, the toolkit that provides API for developing AR mobile app.

2) MongoDB, the database that can store data of objects flexibly.

3) Web page technologies, which can form an entry for building managers or users to input information of indoor objects.

4) Navigation algorithm, which can estimate users current position according to inputted camera captures and give direction suggestions. It is based on image features extraction and matching.

## Design of the proposed system

### User input and information output

The Fig 1 below shows the user diagram illustrating the input from users and corresponding output.

This project assumes that users are unfamiliar with indoor environment of a building and don't know what the useful input is for the proposed mobile app. In order to provide better user experience, with inspiration from [9] and [11], the system uses natural markers (i.e. room signs) and static objects (i.e. doors) indoors as reference for positioning computation. Users just simply open cameras on their device and scan the scene in front of them. Objects appear in the live captures will be detected and recognized automatically. Once identities of those objects are confirmed by the image recognition techniques, users’ current position as well as contextual massages about dynamic objects/events nearby will be displayed. Users can read more detailed information by touching those text on the GUI. They can also ask the system for places of objects/events/facilities by inputting some keywords. The results will be a series of arrows guiding them towards destinations.

Figure 1

system

users

Scanning real scene

Enquiry location

Enquiry information

### System overview

As shown in Fig 2, the proposed system contains 4 big components. Data flow through the whole system as shown in Fig 3.

Within Mobile App scope, ARCore is responsible for processing cameras’ input, communicate with server and drawing virtual objects onto GUI based on its own understanding of indoor environment.

Within Server-side Program scope, Object Recognition program in server receives images from clients and figures out objects identities on images after matching with data stored in database. Information Service will be asked by Object Recognition to search data in database and save the correct results sent from Object Recognition. After computation based on those objects identities results, navigation or other information will be sent to Mobile App.

Database will store features extracted from sample pictures, objects related information and floor plan of a building.

Web Entry is only for information operation such as update.

Sever-side Program

Mobile App

Web Entry

MongoDB

Figure 2

Camera

ARCore

GUI

Information Service

Object Recognition

Navigation

Data Retrieval

database

camera

ARCore

Live capture

GUI

Virtual objects

Object recognition

Live capture

Information service

Object identity

Related information

Navigation computation

navigation

position information

All information

Figure 3

Live capture

### Detailed design of the navigation

The navigation component contains three sub-components: *positioning calculation*, *path finding*, and *path tracking*. Fig 4 shows the data flow among these three sub-components.

*Positioning calculation* is responsible for calculating users’ current position through triangulation according to those recognized objects and their location records. For more accurate calculation, it needs parameters from cameras and other built-in sensors (i.e. magnetic sensor) for input. Although ARCore provides APIs that are helpful for calculating position (i.e. estimate the distance between device and target object), there is still another way to estimate position by comparing size of an object in live captures and that in sample database. The proposed system uses both two ways for more accurate result. The results will be then passed to *path finding*.

*Path finding* is responsible for finding ways from users’ current positions towards destinations. It takes results from *positioning calculation* and other data from users’ mobile devices (i.e. users’ current orientation) as input and selects the shortest way towards destination. Results will be a series of vectors pointing at the nearest referencing point (i.e. an exit sign) along the path. These results will then be passed to *path tracking*.

*Path tracking* cooperates with mobile devices and is responsible for path tracking and calibration. Vectors sent by path finding will be arranged as a queue and stored as buffer. The head of the queue will be sent to users (these vectors will be converted in arrows on GUI). Users will not receive the next vector until they reach a new reference point and scan the new reference point into *position calculation*. It is not guaranteed that the path selected by path finding is absolutely correct or users will follow the navigation arrow strictly. Therefore, when a new result calculated by *position calculation* comes up, it should be compared with the buffer stored previously for calibration. If no problem is found, the head of the rest queue will be sent to users’ devices. Otherwise, the system will calculate and select a new path again.

Figure 4

Positioning calculation

Path finding

Path tracking

User devices

Live captures

Data from built-in sensors

Positioning data

database

Path

buffer

store

results

Check error

Re-find path

navigation

### Detailed design for the object recognition

This part is going to describe what kind of the objects should be taken as reference markers for the proposed system and then describe the process of object recognition.

The system uses natural flatten object (i.e. doors, windows or room signs) as reference markers. This kind of object is compatible with ARCore and can make the system less complex but more robust and efficient. In addition, the object should be in regular shape if possible (i.e. rectangle) and at least with size of 30cm \* 30cm.

In the designed object recognition algorithm, the system first lets ARCore in users’ devices classify the type of detected object and then marks those object with rectangle boundaries. The boundaries will not only appear on GUI, but also be drawn on live captures that are to be sent to the server. With recognized type and dawn boundaries, object recognition program running on sever can speed up its feature extraction and matching process.

The rest works of object recognition are done by a framework, of which the algorithm and concept are not developed by this paper. Hence the details of them will not be elaborated.

### mapping between live captures and floor plan

Another key process is to establish mapping between real objects and corresponding coordinates on floor plan. Building managers should provide a floor plan of a building indoor environment to make the proposed system works. The floor plan will be first split into a number of grids. Each corner of the grids has its unique coordinates. Therefore, positions of reference markers will be converted into such coordinates on the floor plan. When calculating users’ position, the results will be rounded to the nearest coordinates.

## Evaluation design

OUHK will be set as experimental site. Given that the image sampling is very time-consuming, the evaluation only selects three floors of main campus as experimental site.

As the ARCore currently can only works in small range of phones, this project uses Mi 8 as an experimental device.

The experiment will evaluate the proposed system from different aspects, including *performance*, *robustness*, and *usability*.

In *performance* evaluation, it estimates the time delay between scanning real objects and displaying corresponding information.

In *robustness* evaluation, it estimates correctness ratio of object recognition with variant illuminating conditions in different indoor places.

In *usability* evaluation, it lets students or staffs in school use this app and collect their feedback. It mainly tests whether users can gain information that they really want, and whether users can take advantage of navigation to reach their destination.

References

[1] T. Olsson, E. Lagerstam, T. Kärkkäinen, and K. Väänänen-Vainio-Mattila, “Expected user experience of mobile augmented reality services: a user study in the context of shopping centres,” *Pers Ubiquit Comput*, vol. 17, no. 2, pp. 287–304, 2013.

[2] C. Ruizhi and C. Liang, “Indoor Positioning with Smartphones: The State-of-art and the Challenges,” *Acta Geodaetica et Cartographica Sinica*, vol. 46, no. 10, pp. 1316–1326, 2017.

[3] H. M. Khoury and V. R. Kamat, “Evaluation of position tracking technologies for user localization in indoor construction environments,” *Automation in Construction*, vol. 18, no. 4, pp. 444–457, 2009.

[4] R. Mautz, “Indoor positioning technologies,”

[5] G. Bhorkar, “A Survey of Augmented Reality Navigation,” *preprint arXiv:1708.05006*, 2017.

[6] Ronald T. Azuma, “A Survey of Augmented Reality,” *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[7] Apple Company, *ARKit.* [Online] Available: https://developer.apple.com/documentation/arkit. Accessed on: 20 Oct, 2018.

[8] Google Company, *ARCore Overview.* [Online] Available: https://developers.google.cn/ar/discover/. Accessed on: 20 Oct, 2018.

[9] M. Neges, C. Koch, M. König, and M. Abramovici, “Combining visual natural markers and IMU for improved AR based indoor navigation,” *Advanced Engineering Informatics*, vol. 31, pp. 18–31, 2017.

[10] J. Kim and H. Jun, “Vision-based location positioning using augmented reality for indoor navigation,” *IEEE Trans. Consumer Electron.*, vol. 54, no. 3, pp. 954–962, 2008.

[11] A. Xiao, R. Chen, D. Li, Y. Chen, and D. Wu, “An Indoor Positioning System Based on Static Objects in Large Indoor Scenes by Using Smartphone Cameras,” (eng), *Sensors (Basel, Switzerland)*, vol. 18, no. 7, 2018.

[12] Harlan Hile and Gaetano Borriello, “Positioning and Orientation in Indoor Environments Using Camera Phones,” *IEEE Computer Graphics and Applications*, vol. 28, no. 4, pp. 32–39, 2008.

[13] H. Bae, M. Golparvar-Fard, and J. White, “High-precision vision-based mobile augmented reality system for context-aware architectural, engineering, construction and facility management (AEC/FM) applications,” *Vis Eng*, vol. 1, no. 1, p. 3, 2013.

# Appendix A. Progress Report

## Overview of project progress

The almost the whole system has been designed expect one component – object recognition. Because it has not determined which framework should be used to develop the task. (OpenCV, tensorflow, and so on). Only about 20% of the project has been completed. It moved slowly because there are a lot of tools to learn.

## Tasks status tracking

|  |  |
| --- | --- |
| **Tasks** | **Complete** |
| Design database schemas. | 100% |
| Design a schema for floor plan of a building  Design a schema for storing information of indoor objects/events | 100%  100% |
| Design information input interface | 50% |
| Design a web page GUI  Implement database operation through the web page | 50%  50% |
| Design and develop a mobile app with AR | 50% |
| Design GUI and components for the app  Design and implement AR interaction. | 100%  50% |
| Design and develop a server-side program. | 0% |
| Develop a function for object recognition  Design and implement an algorithm for navigation  Design and Implement communication between server and app | 0%  0%  0% |
| Test and evaluation | 0% |
| Test and debug the whole system  Experiment and evaluation | 0%  0% |

## Key success and failures

### Success

GUI of mobile app was preliminarily designed and implemented.

Building web entry with mongodb(a non - data schema database) is almost completed.

### Failures

It tried to rendering text information on GUI using APIs provided by the Android or ARCore, however this way is not compatible with AR App interface. Hence it has to use freetype (a C/C++ framework that can process font file) + openGL to render text instead. But the problem is still difficult.

Another problem is that although ARCore has APIs that can access camera, it cannot adjust parameters of camera (i.e. focal length) and it may affect the quality of live captures.

## Revised project plan