

|  |  |
| --- | --- |
| Project Title | A Mobile Information Kiosk with Navigation for Buildings using AR |
| Study Program | Internet Technology |
| Supervisor | Li Tak Sing |
| Student ID | 11658374 |
| Student Name | Chan Kwok Chi |
| Date | May 27, 2019 |
| Grade |  |

Leave for Blank

Abstract

In 21st century, there have been more and more large size buildings being built in cities. Despite Benefits from services provided by those buildings, people also find themselves being confused by the complex indoor environments. This project tries to solve this problem by building a system with AR and AI techniques that could provide information enquiry and indoor navigation services to visitors for exploring an unfamiliar building. The state of the art AR techniques like ARCore or ARKit provides a set of development tools for developers to build an AR mobile app efficiently. However, all of them currently can only recognize artificial markers like QR code or special pattern image like poster. These QR codes or posters require a cumbersome pre-installation and are difficult for scalability. To remedy this weakness, the proposed system combines AI techniques for recognizing static objects like doors. After object recognition process, navigation or other related information can be displayed on AR interface accordingly. It also takes advantage of AR techniques and data from built in sensors to keep track of motions as well as orientation of users.

Acknowledgement

The first people I thank is my supervisor, Dr. Li, who paid much attention and patience to my depiction on my own situation and the project even although the project made progress very slowly. I should also thank my friend, Wu Wenjun, who is a graduate student also studying in AI field. He gave a clear explanation about the common works of AI algorithms to me which helped me pull myself back from unfruitful research.

I should also appreciate the workers of Google company and others who also work on AI, AR and computer graphic fields. It is impossible to finish this project in such short time without their effective products. In addition, I appreciate the bloggers who write delicate technical blogs sharing their knowledge and experience generously.

Thank you.

Declaration

 I, Chan Kwok Chi (OUHK ID 11658374), certify that the work is original and I have utilized guidance of my supervisor in completing this project, and that the content which is not our own has been attributed and referenced properly.  There should be no copyrighted content without permission to use.  There should be no confidential data.

Contents

[Section 1. Problem Definition 1](#_Toc9764188)

[1.1 Introduction 1](#_Toc9764189)

[1.2 Project Aim 1](#_Toc9764190)

[1.3 Project Objectives 1](#_Toc9764191)

[1.4 Value Proposition 2](#_Toc9764192)

[Section 2. Literature Reviews 2](#_Toc9764193)

[2.1 Problem Analysis 2](#_Toc9764194)

[2.1.1 Data presentation 3](#_Toc9764195)

[2.1.2 Indoor navigation 3](#_Toc9764196)

[2.1.3 Problem analysis conclusion 4](#_Toc9764197)

[2.2 Related Technologies 4](#_Toc9764198)

[2.2.1 Brief description of the proposed system 4](#_Toc9764199)

[2.2.2 Related technologies and difficulties 4](#_Toc9764200)

[2.3 Existing Solutions 5](#_Toc9764201)

[2.3.1 AR tools 5](#_Toc9764202)

[2.3.2 AI frameworks 5](#_Toc9764203)

[2.3.3 Camera-based positioning with different references 5](#_Toc9764204)

[Section 3. Methodologies 6](#_Toc9764205)

[3.1 Requirements and selected techniques for this project 6](#_Toc9764206)

[3.2 Design of the proposed system 7](#_Toc9764207)

[3.2.1 User input and information output 7](#_Toc9764208)

[3.2.2 System overview 8](#_Toc9764209)

[3.2.3 Detailed design for the client-side navigation 10](#_Toc9764210)

[3.2.4 Algorithm used in distance estimation 12](#_Toc9764211)

[3.2.5 Detailed design of coordinates calculation and tracking 14](#_Toc9764212)

[3.2.6 Detailed design of navigation service 15](#_Toc9764213)

[3.2.7 AI trained model in TensorFlow framework 17](#_Toc9764214)

[3.3 Evaluation 17](#_Toc9764215)

[3.3.1 Results of implemented system 17](#_Toc9764216)

[3.3.2 Results of evaluation 17](#_Toc9764217)

[Section 4. Conclusion and future works 17](#_Toc9764218)

[References 17](#_Toc9764219)

# Introduction

## Overview

In modern cities, more and more public places (e.g. a big 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, he still has to spend lots of time in remembering 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 indicating a fact that visitors have a delay in information acquisition. They don’t have effective means for gaining information in need. 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 an information system for a common building that can provide building visitors with instant information enquiry 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 used AI and AR technologies to enhance users’ experience. It hopes that this app could help people explore a building efficiently.

## Project Objectives

1. Design and develop a mobile app with AR techniques.
   1. This app can detect artificial markers and render text-based information properly onto camera live captures.
   2. This app can keep track of users’ accumulated motion and orientation using AR techniques and give navigation guidance accordingly.
   3. This app can estimate users’ coordinates accurately.
2. Design and develop server-side programs.
   1. Implement a path searching program which can search the shortest path from user’s starting point to target.
   2. Implement an information service program which can respond to request from client and other programs resided in server efficiently.
3. Design data structure for indoor reference objects/nodes.
   1. With this data structure, path searching program can work properly.
   2. With this data structure, client and management page can retrieve data needed.
   3. With this data structure, navigation function in client apps can work properly.
4. Implement object recognition program using AI framework.
   1. Implement an AI-based program that can recognize reference object through an AI model.
   2. Select suitable objects for positioning reference inside experimental site and collect data.
   3. Train an AI model so that it is accurate enough for image recognition.
5. Design and implement an information input interface.
   1. a web-based management platform is provided to building manager for information management.
   2. Implement database operation through the web page.
6. Test and evaluation
   1. Test and debug the whole system.
   2. Experiment and evaluation.

## Value Proposition

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

With this system, building manager can save money as they can employ less workers to give information guidance (especially navigation guidance) to visitors.

Although AR, AI and indoor positioning techniques have been developed rapidly in recent years, there is still no general app that takes advantages of those techniques to give general building exploration guidance to visitors in Hong Kong. Some products like the app developed by Cherrypicks [2] can achieve the similar goal, but they are not interested in non-commercial environment. Hence This proposed system can fill the gap.

Technically, this project explores the potential of application, which combines the latest AI image recognition algorithm and AR motion tracking techniques together, in indoor navigation. it adopts the experience from advanced camera based positioning technologies and AR navigation applications, and simplify the process for position estimation and tracking.

## Structure of This Report

The rest parts of this report will first review the researches/works from the angle of what key technologies are needed in this project, especially in camera based indoor navigation field, done previously by pioneers. It will also discuss the existing solutions that can be used in system implementation. The Section 3 will focus on the detailed design of the proposed system. The detailed of navigation function as well as the algorithm/idea used in position calculation and tracking will be highlighted. Section 4 at the end will report the evaluation results of the proposed system with some mobile app screen dumps and data from experiment.

# 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 indoor 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 applications (e.g. google map), text information is often displayed onto a 2D map or along with a picture. This kind of GUI some time is difficult for users to identify the corresponding 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 camera live captures. This way is more natural in sense therefore user experience is improved.

An introduction of AR will be in section 2.2.2.

### Indoor navigation

With the consideration of using AR user interface just mentioned, navigation function of the information kiosk should at least fulfill two requirements: 1) keep track of users’ real-time orientation and position 2) display navigation guidance accordingly. 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 [3]: 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 [4] , 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 and powerful computational hardware. Camera-based positioning technique is intrinsically compatible with AR and now becoming dominant in this filed [5]. 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 for providing 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 [6] 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 corresponding information of indoor objects. 2) A client app. It provides AR user interface, communicates with server and process images inputted from camera. 3) A server-end program. It processes queries from clients, retrieve data from database and searches paths for providing navigation guidance. Obviously, the second component needs a series of AR and AI tools to detect environment and to draw virtual objects; the third component needs an algorithm for path nodes defining and searching.

Details of the system will be elaborated in section 3.

### Related technologies and difficulties

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

The concept of Augmented Reality(AR) technology in the context of this paper is to provide a graphical interface to users with real world perspective superimposed upon by or composited with virtual objects in real-time [7]. It could be used to supplement information, which cannot 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.

Artificial intelligence(AI) is a technology that to take advantages of computing power of computers to predict data and to take proper actions onto data inputs. (HERE SHOULD BE FURTHER MODIFIED)

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 accuracy in positioning estimation.

## Existing Solutions

### AR tools

Recent years, several AR toolkits have been introduced to software developers. Among them ARKit [8] and ARCore [9] 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.

Both of these two AR development toolkits is built up onto a concept called SLAM.

FURTHER DESCRIPTION ABOUT SLAM AND AR TOOLKITS.

### AI frameworks

(FURTHER COMPLETION)

### Camera-based positioning with different references

The main difference of each existing solutions is the type of markers it references to[5]. 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. More importantly, those markers cannot strongly contribute to the navigation function if they are small in size because users have to move closed enough to markers to scan them.

2) Use references from samples of images. Just as described in [10] and [11], 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 [12] and [13], 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 [14]. 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 developing the navigation function.

# Methodologies

In principle, the basic methodology for this project is Build-and-Test. Hence this section will first introduce the requirements and selected techniques for this project, then describe the design for the proposed system and finally the evaluation results of tests onto the system.

## Requirements and selected techniques for this project

To achieve the aims of the proposed system, at least two key technical requirements should be fulfilled. One is the image recognition, and the other is the AR presentation. The image recognition techniques 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 technique in order to enhance user experience.

To fulfill the technical requirements above and construct the whole system, it employs the state of the art techniques as follows:

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

2) TensorFlow, the AI framework that can train and freeze model for object recognition.

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

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

## 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.

Figure 1

system

users

Scanning real scene

Enquiry location

Enquiry information

The proposed system 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 [10] and [12], the system uses flat static objects (i.e. doors, logos) indoors as references for positioning computation and uses artificial markers (i.e. QR code, post) for AR information presentation.

To use AR information enquiry service, users need to scan artificial markers that are near the objects. The corresponding text-based information will be attached to the markers displayed in AR style.

To use navigation function, users should open cameras on their devices and scan environment around them. Objects appear in the live captures will be detected and recognized automatically. Users will be informed of the reference objects with rectangles once identities of those objects are confirmed by the image recognition function. When they want to go to somewhere indoors, for example, they want to know the places of ATMs, they should first input keywords for searching corresponding facilities. Navigation function start working if users choose one facility on the list of searching results and confirm the willing of getting there with system. Users are still required to scan objects nearby after previous step, with their orientation facing against the objects just detected. The system will take the coordinates of reference object as parameter and calculate the users’ current position. The final outputs will be a series of arrows guiding them towards destinations and a point representing their current position on floor plan.

The screen captures of user interfaces are in the following sections.

### System overview

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

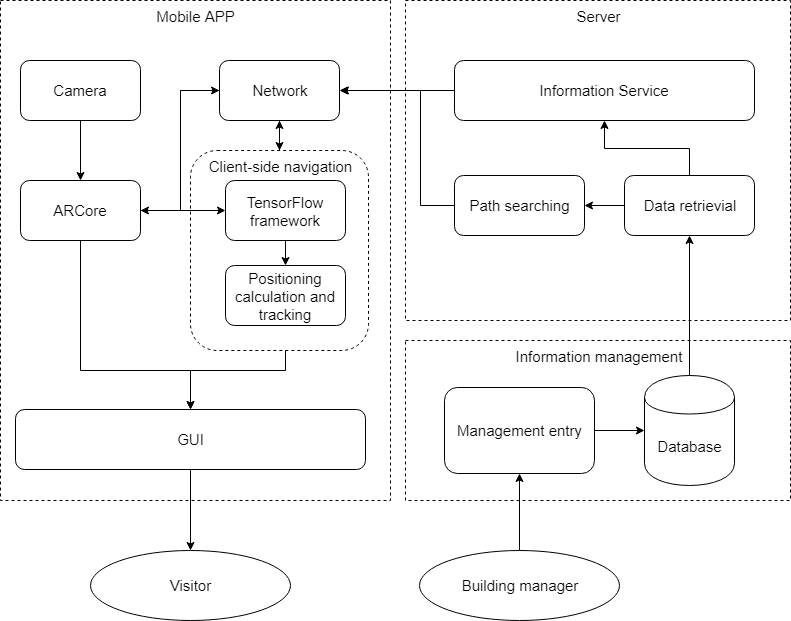
Within *Mobile App* scope, ARCore is responsible for processing image inputs from camera, detecting artificial markers and generating anchors onto which text-based information is rendered. It will also keep track of users’ continuous rotations and motions which are essential for positioning estimation and tracking. The *client-side navigation* uses *TensorFlow framework* to recognize static objects on images passed from ARCore, updates users’ current position coordinates and gives navigation guidance accordingly. Both *ARCore* and *client-side navigation* need to communicate with server for retrieving useful data from server through *network* component. Results from them will be uniformly drawn onto *GUI*.

Within *Server* scope, *Information Service* is responsible for receiving and processing common information enquiries from *Mobile App*. *Path searching* function starts working once it receives enquiries from clients. It searches the path from the users’ current position towards target and send the path result back to clients.

*Database* will store objects related information.

*Management entry* is provided to building manager for information management such as updating a time schedule of an activity.

Figure 2



database

camera

ARCore

Live capture

GUI

Virtual objects

Client-side navigation

Live capture

Information service

Object identity

Related information

Navigation computation

navigation

Object coordinates

Information response

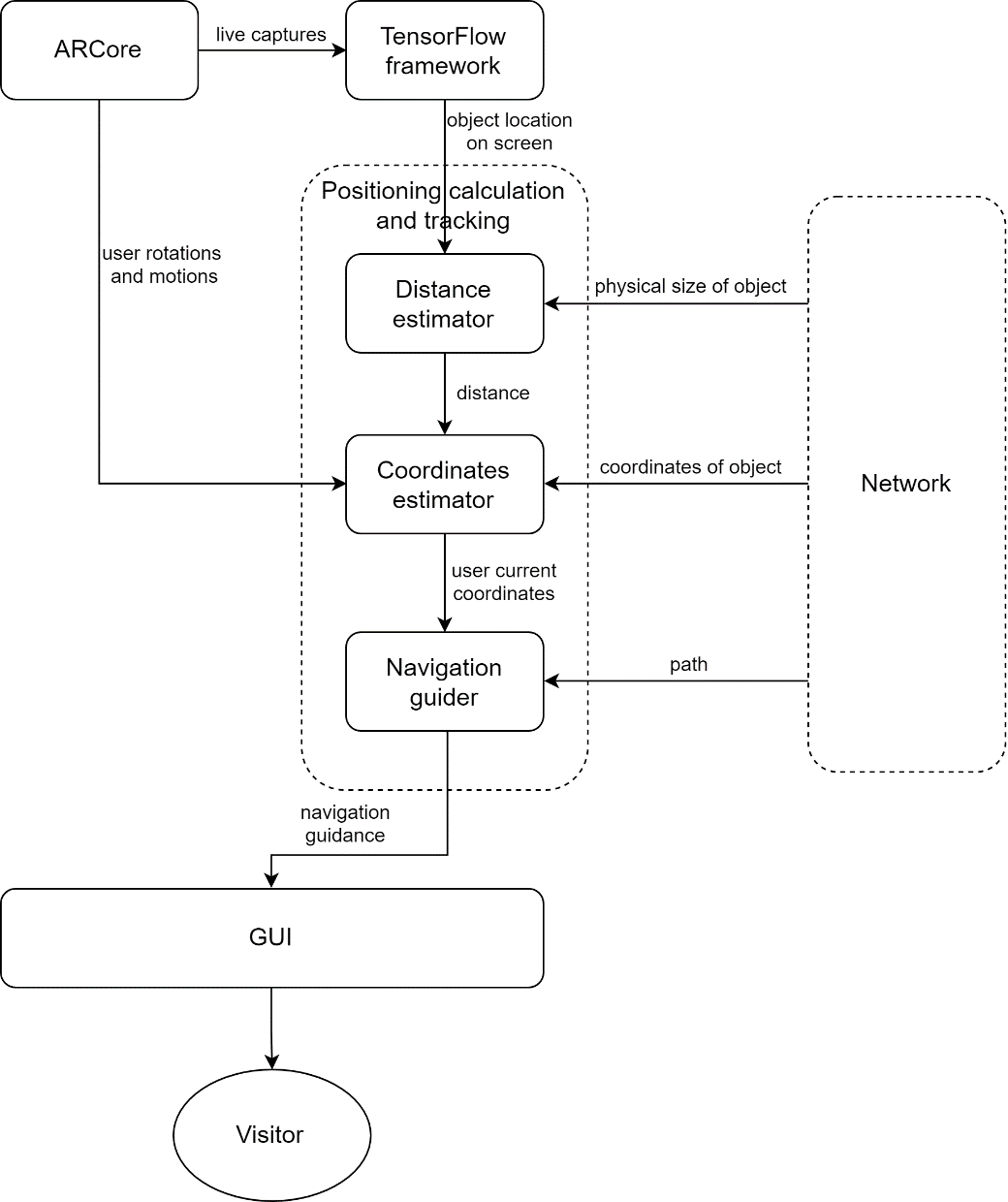
Figure 3

target place

### Detailed design for the client-side navigation

The *positioning calculation and tracking* inside *client-side navigation* consists of three sub-components: *distance estimator*, *positioning calculator*, and *navigation guider*. Fig 4 below illustrates the data flow among these three sub-components.

Figure 4



*Distance estimator* takes identity and real size of recognized object as input to calculate the distance between the object and user. Such distance is essential for accurate estimation of users’ initial coordinates. The result from it will be filtered and then passed to the *coordinates estimator*. The *coordinates estimator* calculates the initial coordinates of user based on the distance estimation result and the coordinates of reference object, and keeps track of users following position and orientation through cooperation with ARCore. The *navigation guider* retrieves path result from server and give guidance according to user’s current position.

### Algorithm used in distance estimation

The proposed system uses a self-designed algorithm to estimate distance between user and reference object, based on the camera projection theory shown in Fig 5.

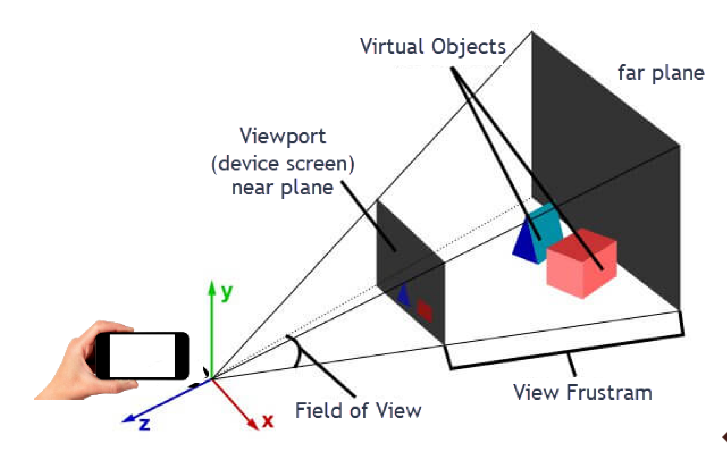


Figure 5

In general camera image capturing process, objects in real world will be first projected onto the far plane and then be translated and transformed onto the near plane which is also the actual image that user can see on their screen. This projection frustum implies that the distance between user and real object can be figured out through similar triangle nature. The relation among essential arguments for calculation is revealed in equation (1).

(1)

Where f is representing the focal length of camera, l is representing the size of object on screen measured by pixels, L is representing the actual size of object measured by meters, and D is representing the distance that is to be calculated.

The system takes on the height of an object on screen, which can be obtained from detection box proposed by AI algorithm, as l in the formula for calculation. It does not take on width for estimation because the width of object on screen varies greatly while the height almost remains unchanged as users rotate their devices in normal motion (i.e. the pitch of device changes little). Therefore, picking the height for estimation can be more reliable.

One more thing should be considered is that the orientation of devices will contribute errors to distance estimation. The android system follows the convention of Euler Angles System in describing the rotation of a mobile device. The angles are Yaw (or Azimuth, which represents the angle between the Y axis of device and magnetic north pole), Pitch and Roll [15], which all can be easily obtained from built-in sensors. The errors are mainly contributed by Azimuth and Pitch just mentioned. We can theoretically reduce the error through triangulation. However, in practice of device orientation estimation, the Azimuth value is not accurate and even variant under the same conditions (i.e. stand at the same place and facing the same direction). Because of the fact just mentioned, the proposed system makes a key assumption that the users are facing against the reference object (i.e. the angle between users’ orientation and the object is 90 degrees) when they are scanning it, just as shown in vertical view in Fig 6. Although user’s actual orientation is different from the assumption, the program can filter out the estimation results that seem to be with large error by comparing the ratio of height to width of detection box on screen with that of real height to real width of object. If difference between the two ratios is smaller than a threshold value, it implicates that the user is facing against the object.

Figure 6

90°

Vertical view

user

object

With this assumption, there will be no need to calibrate the error brought by Yaw of device. However, it is still required to reduce the error, which may be as large as 20cm, brought by the Pitch of device. For clear, this paper summarizes the geometrical model for calculation in Fig 7.

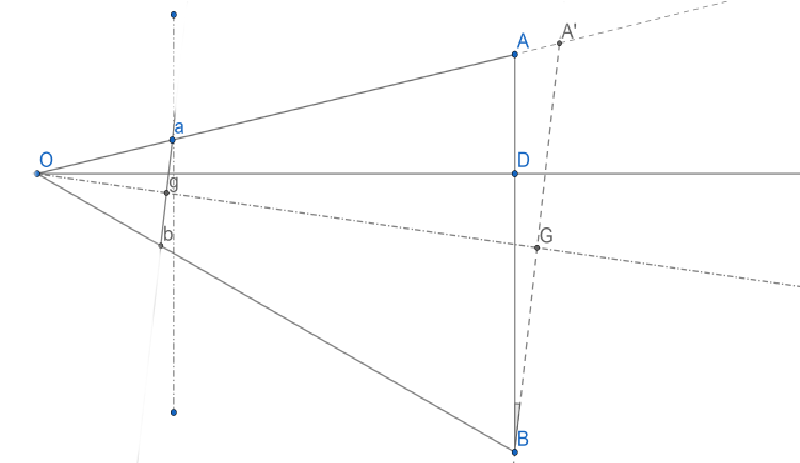


Figure 7

The geometrical model above represents the vertical section of camera projection frustum with triangle OA’B. Point O represents the camera; Og represents the focal length; OG represents the D in equation (1); ab represents the height of object on screen; AB represents the real object; OD represents the actual distance; ∠a = ∠B = θ = Pitch of device.

The estimation result (i.e. D) from equation (1) is not the actual result we want. The actual distance is represented with line segment OD, which is perpendicular to the reference object. We can calculate the OD in the following way:

OD = (2)  
Through equation (2), the distance can be estimated precisely.

### Detailed design of coordinates calculation and tracking

There are many methods to achieve the same goal. Like the method described in [12], it uses the pre-defined feature points in sample images to calculate the user’s current position accurately. However, the algorithm it employs is relatively complicated as it includes feature points matching with a suitable filter and precise data collection for 3D coordinates of those feature points. It cannot keep track of user’s orientation or motion. Thus the proposed system does not employ such method and calculate user’s position in another way. Before going deeper, the data structure of properties of reference object should be well defined. The JSON-like style data structure below lists data (not all) that are essential for coordinates calculation. Building managers should collect these data correctly.

{

Object name: the identifier of object

Coordinates: 2D coordinates of center point of object

Floor: the floor where the object is on

Facing: the surface of object is facing at certain direction

}

We need the facing of object to determine user’s initial orientation:

User’s orientation = facing – π (3)

Note that we should first define 2D-coordinates system for floor plan with Y axis direction pointing at 0 degree (no need to be consistent with direction of magnetic direction in nature). All directions are relative to that axis. User’s initial coordinates can be calculated by minus distance offset as belows:

User’s coordsX = object’s coordsX + distance \* sin (facing)

User’s coordsY = object’s coordsY + distance \* cos (facing)

It is still needed to keep track of user’s motion and orientation. As ARCore framework provides tools for motion tracking and rotation estimation which are relative to device’s initial state described by ARCore world coordinates system, we can convert user’s position as well as orientation from ARCore coordinates system into floor plan coordinates system. The tactic used in the system is that: we first record down user’s position coordinates and orientation in floor plan coordinates system as a starting state when the user is scanning reference object to start navigation function. Then we use ARCore to obtain user’s motion and rotation accumulated since starting state and take them as offsets. The updated position and orientation will be result of starting state minus such offsets.

### Detailed design of navigation service

*Navigation guider* is responsible for finding ways from user’s current position towards destination. Before giving navigation guidance to user, it should first enquire server for shortest way from user’s starting point to target. The path design (i.e. nodes setup) should be illustrated in this paper, because such path is not always straightforward. The difficulty is that the system should give proper guidance to indicate whether user should make a turn at corner/cross district. To achieve this goal, hidden nodes should be introduced, which are opposite to observable nodes (i.e. reference objects). A number of hidden nodes should be set inside the building, especially at corner places, for indirect reference for navigation. The JSON-like style data structure below summarizes data of nodes that are essential for navigation service.

{

isHidden: marker for hidden nodes

Connected nodes: neighbor nodes that are reachable from this node

}

All data of nodes (including data of reference objects) are stored in the same database for structural simplicity. The ‘isHidden’ property is used to identify whether the node is hidden or not. If the node is hidden, it will not be countered into the searching result list of targets to which users want to go. ‘Connected nodes’ is a list of neighbor nodes that are connected with current node. *Path searching* algorithm in server will select nodes in such list as successors of current node on which it takes steps in searching iteration. With all nodes properly set up (i.e. no node is isolated), this data structure makes sure that all the nodes along with the path result are reachable.

After receiving path result from *path searching,* which is a queue of nodes with coordinates property, *navigation guider* starts its work by checking the currently updated coordinates of user and selecting the nearest node in queue as intermediate/current target iteratively. For each intermediate target, *navigation guider* interprets it into an arrow with direction pointing at it relative to user’s current orientation. It is assumed that users will follow the guidance and move from starting points to intermediate target and then the next intermediate target until they reach the end target. Therefore, those nodes in front of the selected target in queue will be removed. To make the system works smoothly, we should set a threshold value for distance between user and nodes. If such distance is smaller than the threshold value, we can infer that the user has reached the target and just remove it from the path queue. The next node will be selected as the current target. Navigation function will stop if the TensorFlow framework has detected the target object on screen or user is closed enough to target (distance smaller than threshold value).

### AI trained model in TensorFlow framework

This part will introduce the detailed standard of reference object and the AI algorithm selected in the proposed system.

The system uses natural flat static object (e.g. doors) as reference markers. This kind of object is compatible with ARCore and can make the system less complex but more robust and efficient, compared with reference to non-flat object. In addition, the object should be in regular shape if possible (e.g. rectangle) and at least with size of 30cm \* 30cm.

# Evaluation

The Open University of Hong Kong is set as experimental site. Given that moving between different floors is not much different from moving within one floor, the evaluation only selects ground floor 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.

## Results of implemented system

## Results of evaluation

# Conclusion and future works

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] Cherrypicks Company, *products.* [Online] Available: https://www.cherrypicks.com/products/. Accessed on: 26 May, 2019.

[3] 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.

[4] 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.

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

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

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

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

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Google Company, *SensorManager.* [Online] Available: https://developer.android.com/reference/android/hardware/SensorManager.html. Accessed on: 24 May,2019.