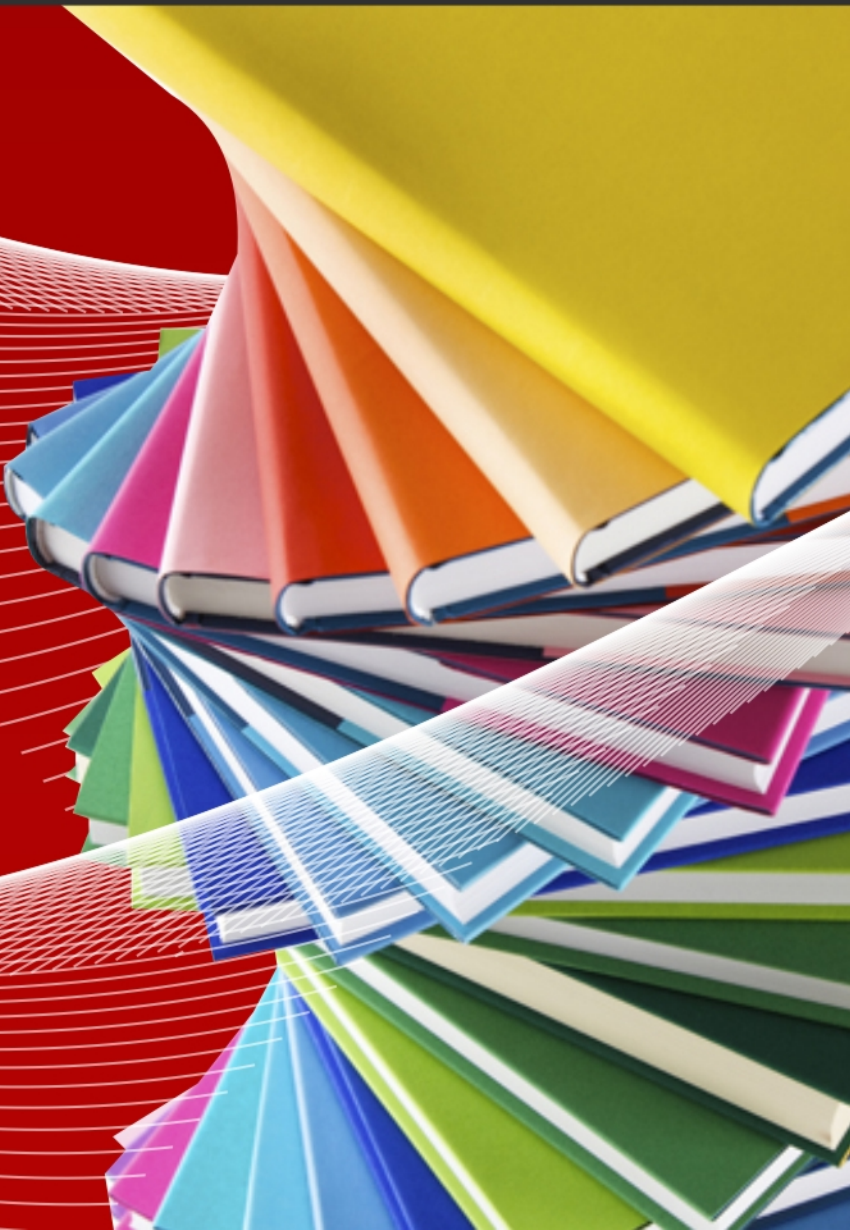
澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Outstanding Academic Papers by Students

學 生 優 秀 作 品



University of Macau

Faculty of Science and Technology

澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

Indoor navigation by image recognition

by

CHOI IO TENG , Student No: DB325174

Final Project Report submitted in partial fulfillment
of the requirements of the Degree of
Bachelor of Science in Computer Science

Project Supervisor

Prof. Pun Chi-Man

26 May 2017

DECLARATION

I sincerely declare that:

1. I and my teammates are the sole authors of this report,
2. All the information contained in this report is certain and correct to the best of my knowledge,
3. I declare that the thesis here submitted is original except for the source materials explicitly acknowledged and that this thesis or parts of this thesis have not been previously submitted for the same degree or for a different degree, and
4. I also acknowledge that I am aware of the Rules on Handling Student Academic Dishonesty and the Regulations of the Student Discipline of the University of Macau.

Signature : ____________________________

Name : Choi Io Teng

Student No. : DB325174

Date : 26 May 2017

ACKNOWLEDGEMENTS

I would like to thank my college, University of Macau, for providing an opportunity to carry out a project as a partial fulfillment of the requirement for the degree of Bachelor of Science.

I would also like to express my gratitude to my supervisor, Prof. Pun Chi-Man, who has aided and provided comments for this project.

ABSTRACT

An indoor navigation system using image recognition method is proposed in this paper. The structure of the proposed method can be classified into two main parts: indoor positioning and navigation. We propose to use image recognition to achieve indoor positioning. Images of indoor environment are first collected and features are extracted to form a feature collection. Then, getting real-time images from the user, and perform feature extraction. Through the comparison of the result and the collection, the current location of the user can be recognized. To achieve real-time computing effect, DARTs features which have a faster performance than SURF and SIFT are used for recognition process. REVV algorithm is then used to generate global signature for the local features to have a faster performance in storing and searching. Once the position is confirmed, navigation will start. On the navigation process, Dijkstra 's algorithm is used to perform the shortest path search on the indoor map graph and return a route to a certain destination. Based on the resulted route, user's position will be kept tracking to confirm that user is within the desired path. Otherwise, path calculation is performed again. Our proposed system is implemented on an Android platform for testing. The experimental results indicate that the error rate of our proposed method is low and the performance of the system is satisfied. Also, indoor navigation by image recognition can overcome the disadvantages and the constrain of other indoor navigation method.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. Introduction

Navigation system is a system which can identify position of the users and aid users to their destination. It is a crucial and favourable assistant for users when they are experiencing in an unfamiliar environment. Due to its advantages, navigation system is widely employed in diverse areas. There are several different types of navigation system and the most common seen is the outdoor navigation. With the help of Global Positioning System (GPS), outdoor navigation is very effective and precise. Moreover, GPS is considered as the most pertinent method for outdoor navigation. However, it is not the case for indoor navigation. GPS is a radio-navigation system which transmit data to receiver by a satellite in space. Satellite sends GPS signal through radio waves and provides the relative geolocation to the receiver. Nonetheless, radio signals have attenuation characteristics. Radio waves are attenuated and scattered by obstacles. For indoor environment, there are coverings and numbers of obstacles which greatly reduce the performance of the strength of the GPS signals. As a result, GPS systems do not function well in indoors.

To overcome this challenge and achieve indoor navigation, researchers proposed various methods as an alternative to be used in indoor environment. These methods are Wi-Fi fingerprinting, dead reckoning, infrared technology and marker image recognition, [1,2] etc. Yet, these methods require pre-install equipment, networking and maintenance. Moreover, network signals also have attenuation characteristics like GPS signal. Similarly, obstacles which are common in indoor environment can easily affected the strength of the signal and degrade the effectiveness of these methods. Hence, an offline-based system which is independent of microwaves signal is a better solution. Based on these reasons, we proposed an indoor navigation system by using image recognition.

We implemented and tested our application on Android platform. The objective of our proposed indoor navigation system is to provide an adaptable navigation system in indoor environment without any user's constraint and network requirement. The only requirement for the user is simply a device that can allow the application to run. The tested application is implemented based on the environment of a shopping mall in Macau. Data collection and testing are all hosted on that shopping mall environment. Our project and proposed method also received affirmation and will be published in the 9th International Conference on Digital Image Processing (ICDIP 2017).

1.1 Overview

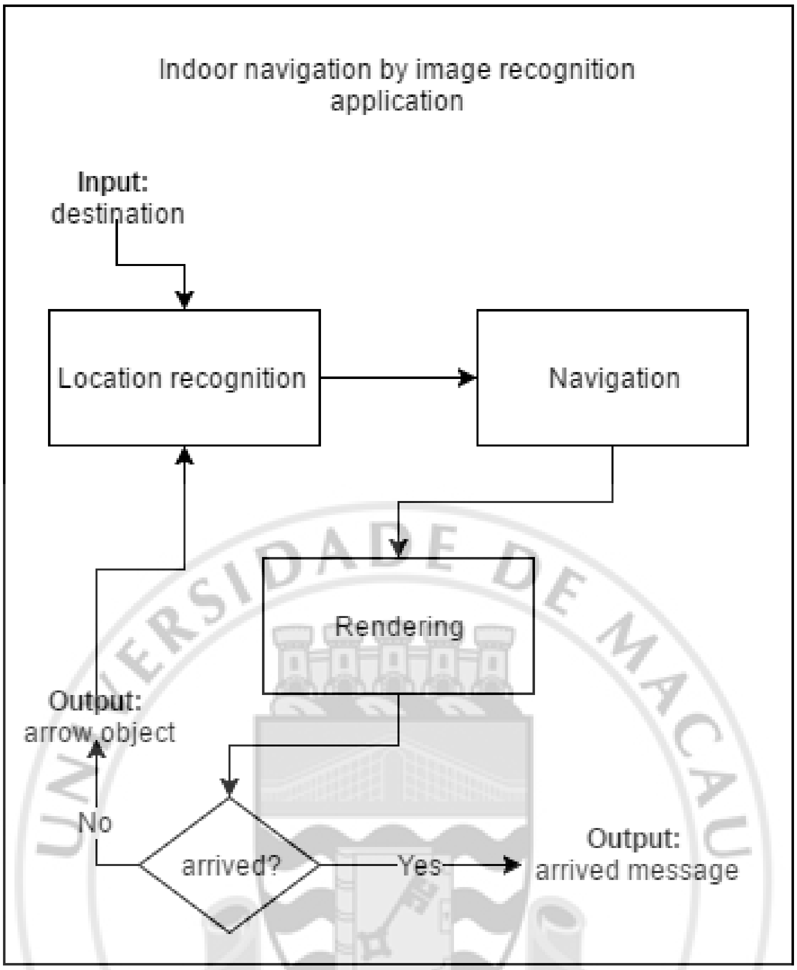


Figure 1: Overview of the proposed system

The overview of our proposed system structure is shown in figure 1. The main workflow of the system can be categorized into three main modules. Firstly, the input, destination is input by the user and recognized as a flag to start the navigation process. The "location recognition" module will firstly in process. The application will retrieve real-time images from user's device. The real-time image is then analysis to perform recognition. Secondly, the user's location which resulted from the first module is then pass to the second module, "navigation" to find the shortest path to reach the location that user has input. Then, the system will check if the location has arrived or not. If not, access the path to learn the orientation connected to the next position. The learned orientation is then passed to the "rendering" module to generate the arrow object to guide user. And process is then returned to the first module if the destination has not arrived yet. However, once a route exists in the memory, the second module will process differently. The second module will track if the current location is within the calculated path. If it is, the system will enter third module. However, if it is not, a route calculation will process again and get a new path. Lastly, if the "navigation" module tracked that the current location is identical with the destination, an arrived message will be altered by the "rendering module". It is a brief overview of our proposed system. A more detail version of the framework of the proposed method will be discussed in the software design specification and implementation section.

1.2 Project schedule

The first step of our project schedule is to have researches on other indoor navigation method to discover pros and cons which might be useful in designing the application. Once we have the basic knowledge of similar fields, the application design step can be processed. We design the application based on certain standards, they are acceptability, reliability, effectivity and efficiency. I am contributed into constructing the basic structure and workflow of the entire application. I am also responsible for all the graphical design of the application.

Once, we have the main structure, we need to decide an appropriate image recognition algorithm used for our application. I am in charged in choosing the desired algorithm for the system. The algorithm is compared based on the computing speed, accuracy and the workload. I did research on different image recognition algorithm and implemented them for comparison. A testing application that perform simple image searching is first implemented by using different image features algorithm. I tested colour histogram, edge histogram, SURF, DAISY etc. There are lots of challenges during the process. For instance, there are some libraries that are only compatible with Java but not in Android platform. Then, tough works are needed to deal with the problem or to find any other alternative algorithm. Finally, after repeatedly met with setbacks, I find DAISY features. Through the searching result, DAISY features also return the highest rate of accuracy and fastest computation time. CraftAR also provides a library to perform DAISY features extraction and recognition. Thus, we decided to use CraftAR SDK to process the image recognition. Besides the image recognition algorithm, we have to decide the route searching algorithm. Our project group chose Dijkstra's algorithm to perform the shortest path searching.

The next step is to take the field study of the testing environment. Then, we design and segment the shopping mall to construct a map graph representing the testing environment. The map graph is used for the route searching using Dijkstra's algorithm. Next process is to collect image data to form data collection. We took photos of the environment and classify them. Then, feature extraction is done on all the data and form items to represent nodes in the graph. At this step, all the preparation for the implementation is done.

The following step is to implement the application. The detail of implementation is discussed in the later section in the report. Implementation is the most challenging part throughout all the steps as errors and bugs emerge and take us lot of efforts to solve. During the process, accuracy of navigation is one of the issues. We then implement extra functions to assist and optimize the performance of the navigation process. Integrating the computer graphic with the screen is another issue. Compatibility errors always emerge. It is a tough work to overcome the challenges and make the entire system functions well.

Lastly, the finished application is then tested in the shopping mall. We pick random location and destination for the test in order to ensure the reliability of the system performance in distinct locations. The error rate of the test result is shown in the later section in the report.

CHAPTER 2. Literature Survey/Related Work

There are two main procedures in navigation system, they are positioning and route navigation. And the biggest challenge of navigation in indoor environment is to identify the user's position and orientation. As GPS signal is no reliable in positioning in indoor environment, past literatures have proposed various methods to achieve positioning in indoor environment.

To identify the position of the user, the location must have unique information that can be obtained during the process. Many literatures have proposed distinct methods to clarify the unique information of the user's location. These proposed methods can be mainly classified into visual or non-visual method [3]. Non-visual positioning method mainly rely on electromagnetic signal to identify the user's position. For instance, using infrared signal, Wi-Fi and radio fingerprint system [4, 5, 6], etc. Infrared and Wi-Fi can identify the user's position by receiving or send signals. For instance, the infrared based method. The infrared sensors are placed at various constant positions and receive signals from emitters within a certain range. Then the signals are analysis to identify the position and orientation through estimating angles and positions of the ceiling infrared beacons. For large environment, massive beacons are required for the system to operate. Also, the error rate of the infrared sensors increase as the distance to the sensors increase. The proposed technique only works well within a small area but not for large scale indoor environment. Most importantly, this method requires a massive installation workload which increase the cost to deploy.

Bahl and Padmanabhan [7] proposed the RFID-based indoor positioning method. It is another non-visual navigation method. This method make use of RF tags and a tag reader to identify the position of a user. The RF tags are glued or appended on equipment in indoor environment. When the reader emits radio wave signal, the circuit in the tag obtain energy and transmit unique information for the reader to identify. This method requires a lower installation cost compare with the infrared technique. However, as the coverage of radio signal is also wide, it is possible for more than one tag responding to the reader. As a result, confusion of position identification occurs. To avoid this situation, tags should be placed wide between each other. If so, a precise navigation cannot be provided. Moreover, the RFID only can identify the position but lack of ability to provide the orientation of the user. It also requires an additional method to provide orientation information in order to achieve navigation. On the other hand, the RFID reader is relatively expensive. It might not be possible to offer every visitor of the indoor environment to have one. Thus, it is essential to have a lower installation cost and a more precise method than a RFID-based method.

Davide et al. [8] proposed Dead Reckoning (DR) technology to estimate the user's position by calculating with user's speed and previous position. The proposed method is achieved with the use of the sensors in the mobile devices. However, sensors in mobile devices are easily interference with magnetic wave or other electromagnetic signals. Recalibration is needed when interference occurs. Through the experiment test in [8], the accuracy and performance of the system will be greatly affected if no recalibration. In DR method, even for a minor difference for one calculation will cause a great deviation for the following positioning as the proposed method requires calculation based on previous position. The accuracy of the system is then greatly affected when interference happen during the process. Moreover, the acceptability is not high since periodically recalibrating might make users feel annoying. As a result, a method that can provide a higher accuracy and acceptability will be admirable.

On the other hand, visual based navigation make use of camera to gain unique information from user's current environment to achieve positioning. For visual methods, the cost is relatively lower than the non-visual methods [3]. The cost for installing and maintaining hardware devices used in non-visual methods is not required in visual methods. A simple mobile device that can take images is enough for the process. It does not require any additional or professional devices to set up the navigation environment. With the popularity of mobile devices, camera is commonly held by everyone nowadays. Moreover, due to the progress of component technology, the hardware performance of the mobile devices is enough to handle the computation requirement for image processing. On the other hand, as unique information is important in identification, a unique marker to identify each location is a common way to achieve the result. This reference marker is usually called Fiducial marker.

Manfred et al. [9] successfully achieve continuous tracking on user's location with a massive number of Fiducial markers. Each marker uniquely identifies each unique location. When user scan the marker with the camera of the device, the system can identify which marker it is and so identify the related position. The advantages of this method are that the accuracy of positioning is high and the cost to implement is low. However, markers are needed to position in each location of the indoor environment. If a continuous and precise navigation is needed, the interval of each marker should not be large. Massive number of markers will be needed for large environment. Moreover, paper-markers are used in the proposed method. These markers are fragile and can be removed or damaged easily. If other materials are used as alternative or integrating the markers into the indoor environment, the cost of implementation will increase. Thus, fixed marker is not an appropriate way to achieve indoor navigation.

Konrad et al. [10] proposed an image searching approach to identify location. The approached method is to make use of a server with several web pages. User takes the picture of the location and sends to the server for searching. Based on the content of the image, the server will return 5 related webpages for the users to decide. Konrad presented a basic concept to use image content for identification. However, it is a web based method and only function well with well-known architecture. Network signal can also be affected by attenuation and some indoor environment do not always provide internet access for visitors. Moreover, the proposed system does not work well for some environment that do not have special and sharp characteristics. For instance, the corridors that are very similar within the same indoor environment might not be correctly distinguished. Thus, a method that can accurately identify difference between similar images and independent of networking signal is needed.

Jongbae Kim and Heesung Jun et al. [11] also proposed an image based indoor navigation method system with pre-installed markers. Images of user's current location is capture by a camera and send to the server computer through LAN network. Then, the server computer will check the image and distinguish the marker from the images. Once the marker is identified, the user's location is known and can perform navigation. The guidelines of route will be an arrow on a 2D map image and shown on a Head-Mounted Display which is worn by the user. The method proposed by Kim and Jun function well when marker can be successfully identified. However, this proposed method share some disadvantages of Fiducial markers method that are mentioned before. Moreover, a HMD is very inconvenient for users to use and networking is required to send images back to the server computer for computation. There are too many constraint on users to use the navigation system. Thus, it might not be a possible method to be used in reality environment. For instance, a crowded area will cause the pre-installed markers being blocked which cause the entire system to be non-functional. Also, it is not possible to require users to search for markers in an unfamiliar environment and wearing a HMD to walk in the indoor environment. Besides, marker-based method can only identify the position of the user but not the orientation of the user. Even if the system can provide an arrow guidelines on a map, it is useless if the user does not know which orientation he or she is facing at. Therefore, a method that can require less constraint on users and convenient for user to use is essential.

On the other hand, route searching and representation is also important in indoor navigation. Dijkstra’s algorithm is every effective in calculating the shortest path to the destination. Route representation is to provide clear information to guide user to the destination and update guidelines based on user’s position. Numbers of literatures had proposed various representations for the navigation process. It is very important that user can understand the guide to the destination otherwise the entire system will be useless.

Christian et al. [12] concluded that 2D or 3D map is not necessary for user using indoor navigation system as sketches of direction is enough for the user to understand. Alessandro et al. [13] and A.J. Brush et al. [14] had experiment with a sequence of testing commands to guide testers to destination. For instance, the commands are "walk 10 steps north", "walk 2 floors down" etc. Through the experiment, these commands provide effective and concrete instructions for a short distance destination. Also, user have a sense to know the time required and the distance till to the destination. However, the situation only works for short distance. If the distance to the destination is a long one, commands will become complicated and difficult for users to remember. Thus, direct and short commands or guidelines are more suitable for user to understand.

Hans et al. [15] approached to use Augmented Reality technology to integrate route direction directly on a paper map. The advantage of this method is that the requirement is not high. User only need to have a paper map and scan with a device to obtain routing information. However, it is lack of information about the landscape of the environment. For those users with a poor sense of direction, even with a map provided, navigation still fail. Despite, using Augmented Reality technology to give information is a good idea. Martin et al, Rehman and Andrew [3, 16, 17] stated that Augmented Reality is a good and straightforward representation to guide user to destination. User can easily understand through Augmented Reality as the view shown in the screen is consistent with human own vision perception. Path directions are like drawing directly on the road clearly showing which way the user should follow. In Andrew Zhong’s paper [17], he uses Augmented Reality to generate an arrow directly on the screen to guide the user. The method Andrew proposed can reduce misunderstanding of the guidelines and have a better acceptability. However, the computation cost of the Augmented Reality is quite high which might not be able to provide a real-time response in using mobile device’s CPU. It is better to have an alternative method that can provide the same effect but with a lower cost. As a result, a method that do not require high computation cost, can guide user in their point of view to provide greater understanding and reduce the chance of navigation failure to occur is needed.

Therefore, our project group summarized all pros and cons of those methods proposed in the mentioned literatures and proposed an imploded approach to achieve the indoor navigation system. We propose to implement a mobile device application which perform identify user's current position based on image features recognition and guide user to destination through integrating computer generated objects on the camera view of the device.

CHAPTER 3. Functional Specification

This section lists all the functional requirement and non-functional requirement of our proposed application. It also provides description to explain the objective and the work that each function does.

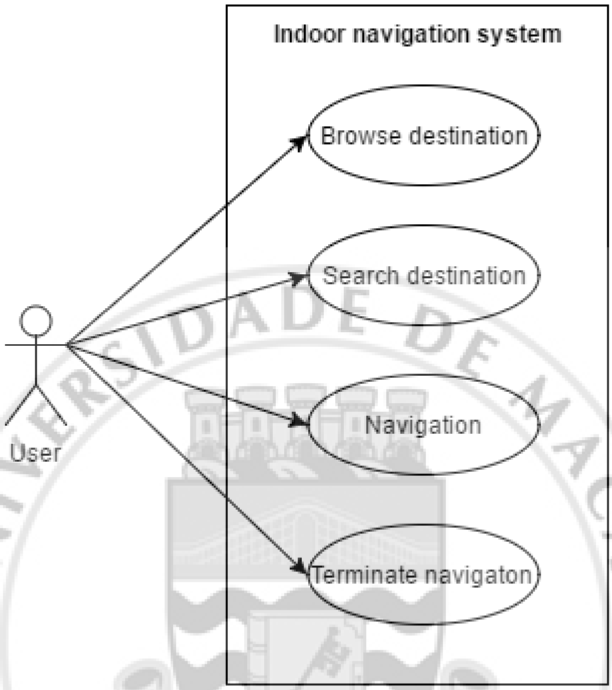


Figure 2: Use-case diagram of the application

The main functions that can be performed by a user in our implemented system is shown in the use-case diagram in figure 1. There are four different functions that are offered for the user to use: browsing destination, searching destination, start navigation and terminate navigation process.

3.1 Functional Requirement

Table 1 : Functional requirement of "browse destination"

	Browse destination
Purpose	The application shall let user to browse and select the destination that he or she wants to go before starting the navigation.
Input	A user press activity.
Output	A list of available destinations in the indoor environment.

Operation	Once a press activity is detected from the main page, the system will access to the local database of the application and retrieve the name of all the destination.

Table 2: Functional requirement of "search destination"

	Search destination
Purpose	The application shall let user to search desired destination by typing. The result should be automatically listed for each letter input action.
Input	Text. (Valid for all symbols)
Output	A list of destinations with the longest match name
Operation	For each text-changing activity in the textbox, retrieve the value and search it through the database to find the longest match name of the destination field.

Table 3: Functional requirement of "navigation"

	Navigation
Purpose	The application shall start navigation process once the user selects the destination.
Input	Name of destination, real-time image from device
Output	A computer-generated arrow pointing next-direction
Operation	Use the name of the destination to perform route-search on map graph and real-time image to identify location. Once the location is resulted, check the route to find next direction used to generate the arrow.

Table 4: Functional requirement of "terminate navigation"

	Terminate navigation
Purpose	The application shall allow the user to terminate the process at any time.
Input	Button-press activity
Output	/
Operation	When a button-press activity is detected, clear the route list and jump back to the main page.

3.2 Non-functional requirement

3.2.1 Accuracy

- The software shall provide direction guides correctly.
- The software shall not recognize a wrong location.
- The software shall not lead user to other non-desired location.

3.2.2 Acceptability

- The software shall be easy to use by users.
- The direction instruction provided by the software shall be clear and understandable.
- The software shall not require lots of user constraint.

3.2.3 Efficiency

- The software shall respond in real-time.
- The software shall guide the user to the destination with the shortest path.

3.2.4 Effectivity

- The software shall successfully lead user to the desired location.

3.2.5 Reliability

- The software shall provide continuous navigation effect for the user.
- The software shall not crash down even if recognition is not success.

- The software shall continue navigation even if the user does not follow the path.

3.2.6 Security

- The software shall not capture, transmit or store user’s real-time image.
- The software shall not transmit and store user’s service record.

CHAPTER 4. Software design specification

This section mainly covers the design and structure of our proposed method to achieve the functional and non-functional requirement listed in the previous section. Also, this section has listed each module of the application, the function that the modules perform and the relationship between each of them. Lastly, it also explains the reason why our project group designed the application in this way.

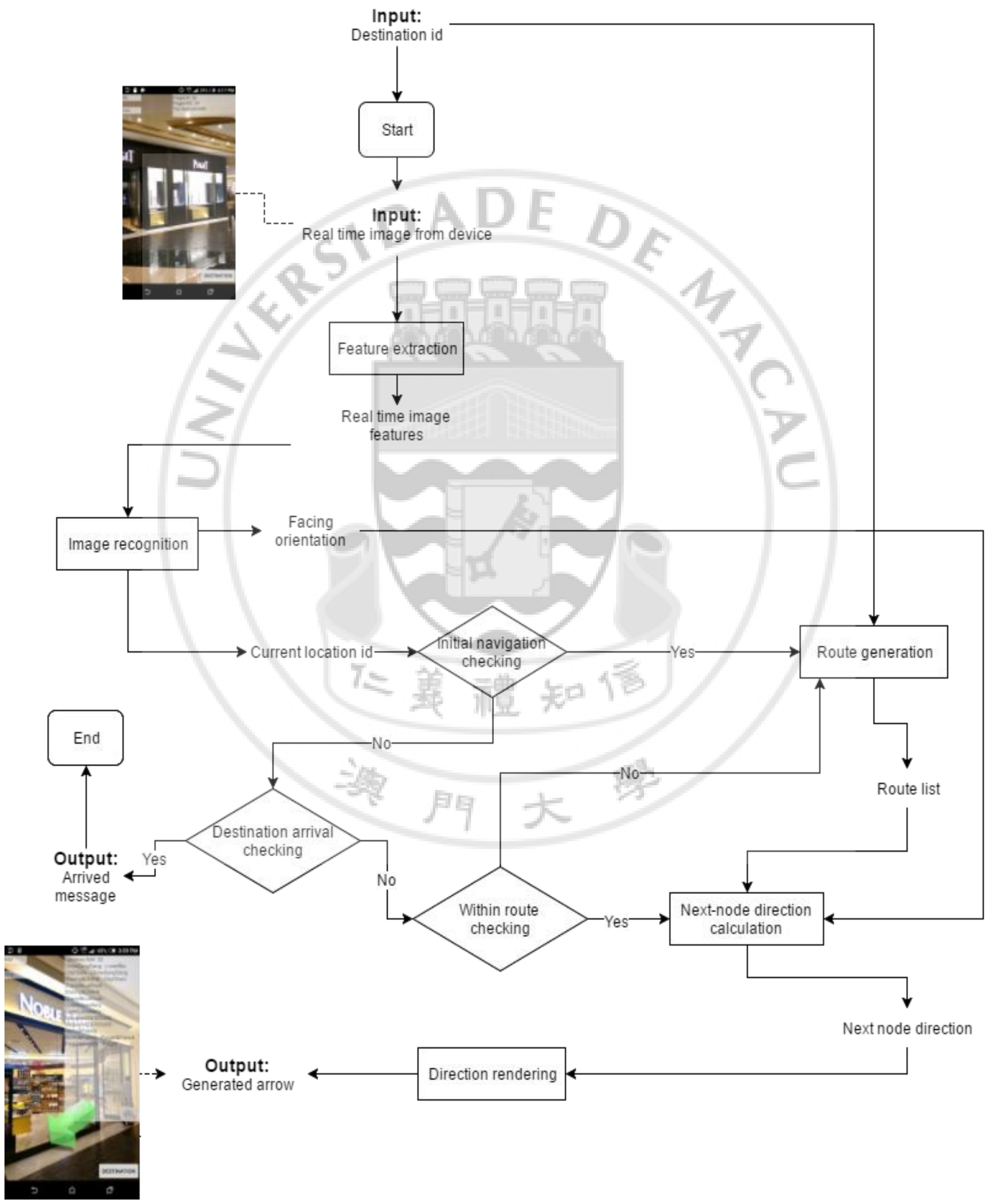


Figure 3: Framework of the indoor navigation by image recognition method

4.1 Overall design

A detail overall framework of our proposed system is shown in figure 3 illustrating the workflow of the indoor navigation process in the application. The entire workflow of our proposed method can be categorized into 3 stages: positioning, navigation, and result rendering. However, this report will not cover all the detail design specification of the entire application. In this section, I will mainly focus to specify the design of the parts which I contributed in building. The remaining part will be discussed in my groupmates' report.

To achieve indoor navigation, our group designed to use real-time image of the user to identify the position. The idea of our proposed method is simple. For instance, if a person wants to reach a destination in an unfamiliar place, he will ask help for people who are local or familiar with the place or even hiring a tour guide to guide him. And the idea of our group is to implement such a tour guide that is familiar with the indoor environment. This tour guide will walk together with the user, knowing where the user is based on own experience and lead the user to the destination. So, the first step of procedures is to implement a tour guide and teach the guide to get familiar with the environment.

To teach the system about the landscape and structure of the indoor environment, images of the indoor environment are collected to form a reference image database. In our proposed method, the target indoor environment is segmented into different items. Each item represents as a specific orientation of a location of the target environment. Each item is named in a format of the location name followed by an orientation. For instance, "ShopA-N", "ShopB-E", "ShopA-W", "ShopB-S", etc. Texts before the hyphen indicates the name of the location and capital letter after the hyphen indicates the orientation. With this design, the system can learn about the location of the user's position through the name id and the orientation that the user is facing from the last capital letter. Besides knowing the location, orientation is included in order to have a more precise navigation effect. The system can direct user to the orientation he or she should go based on the current orientation.

4.2 Design of database

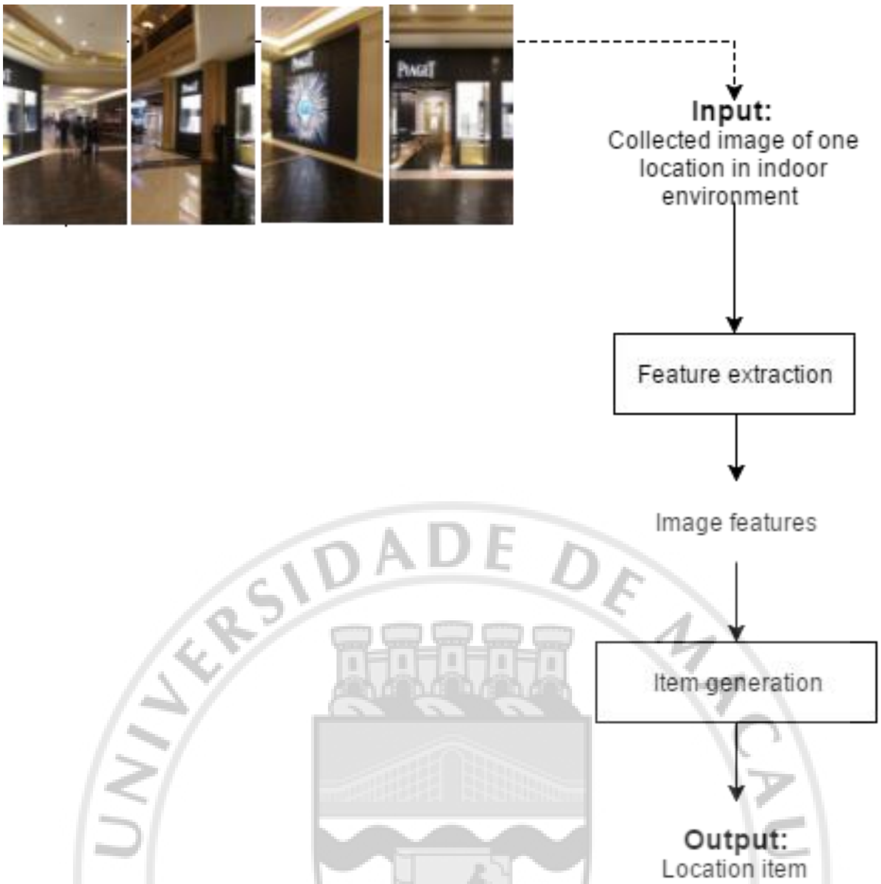


Figure 4: Framework of location item generation

Figure 5: Framework of the generation of image feature database

The framework of generating each location item is shown in figure 4. Firstly, images of a specific orientation of a location are collected. Then, the features of these images are extracted by using DARTs algorithm. The detail and the way that DART's algorithm works will be discussed in the implementation section. In our implementation, we used CraftAR SDK to perform the image extraction. The resulted image features are all combined to form a location item with our designed name tag. Repeatedly works are done for all the segmented location of the indoor environment. The framework of constructing the image feature database is shown in figure 5. All the resulted location items are then combined to form an image collection which will be used in the recognition process. The image feature database is designed to store on local. The database will be embedded into the application. The reason for making the database local is to produce an offline-used application. As mentioned in the previous sections, electromagnetic signal is unstable in indoor environment and not all the indoor environment can provide internet access to the visitors. Embedding the database local will serve as a benefit to provide a stable and reliable application for the user.

On the other hand, it is not enough for the application to know the structure of the indoor environment with the image feature database since the image feature database only teach the system how each location looks like. The system is lack of the knowledge about the connection between each location. As a result, one more database is needed to clarify the structure of the target indoor environment. We designed to use a SQL database to store the information that identify the connection between each location. Each row of the table is each location of the environment with a primary id that indicates the name of the location. And for each row, there are also eight more columns representing each orientation, north, north-east, east, south-east, etc. of the location. And the value of these columns is the id of other location. For instance, if the value of North column of row “ShopA” is “ShopB”, it means that the north of ShopA is ShopB. The SQL table is designed in this way to indicate the connection of each location. With the help of this database, the application knows the structure of the entire target location. Besides, we designed more databases to represent the map graph and weight of each node. The map graph database is used for the Dijkstra’s algorithm for searching the shortest path. However, the map graph is mainly contributed by other members in my group. So, the detail of designing the map graph database is not discussed in this report.

After the image feature database and the structure database are ready, next is the design of the application. As the database is local, it requires time for the application to install and learn about the database. It is a little drawback for the time needed for the database installation. However, our group think that it can be comprised to provide a stable off-line application. Although, the installation time is just a few seconds, there is a chance for the user to have a misunderstanding that the system crash down. So, to prevent the misunderstanding, we design to implement a splash screen indicating that the system is still working during the database installation time. The splash screen is a logo of our project which is also designed by me.

Figure 6: Project Logo

4.3 Design of the application interface

To be an easy-to-use application, the application shall not be very complicated for the user. So, we designed a very simple interface for the application. The application interface is designed in a very straight forward way. The main page of the application consists of an input text box and a list. The input text box is designed for the user to enter the destination that he or she wants to go. However, it is not possible for a user who is unfamiliar with the environment to know what destination there is in the indoor environment. So, to increase the acceptability of our proposed application, the list is used to list out all the possible destination of the indoor environment. Even for the user who is a new visitor of the environment, he or she can browse through the list and decide their destination based on his or her own interests. On the other hand, it will be a troublesome action for the user who is not a first visitor. So, we also append a function that can look after these types of users. The entries of the list will be adjusted based on the text the user input in the text box. The user can enter part of or the entire destination name, the system will check the text box for every letter's changing activity and filter the result showed in the list with the longest match of the input text. This function can provide efficiency for user when searching the destination.

Once the input interface is well designed, next is the main core of the entire application. As shown in figure 3, the navigation process is aroused by a destination id collected from the user. In this step, the software send a request to the device's camera to capture the real-time image of the user's location.

4.4 Design of the recognition module

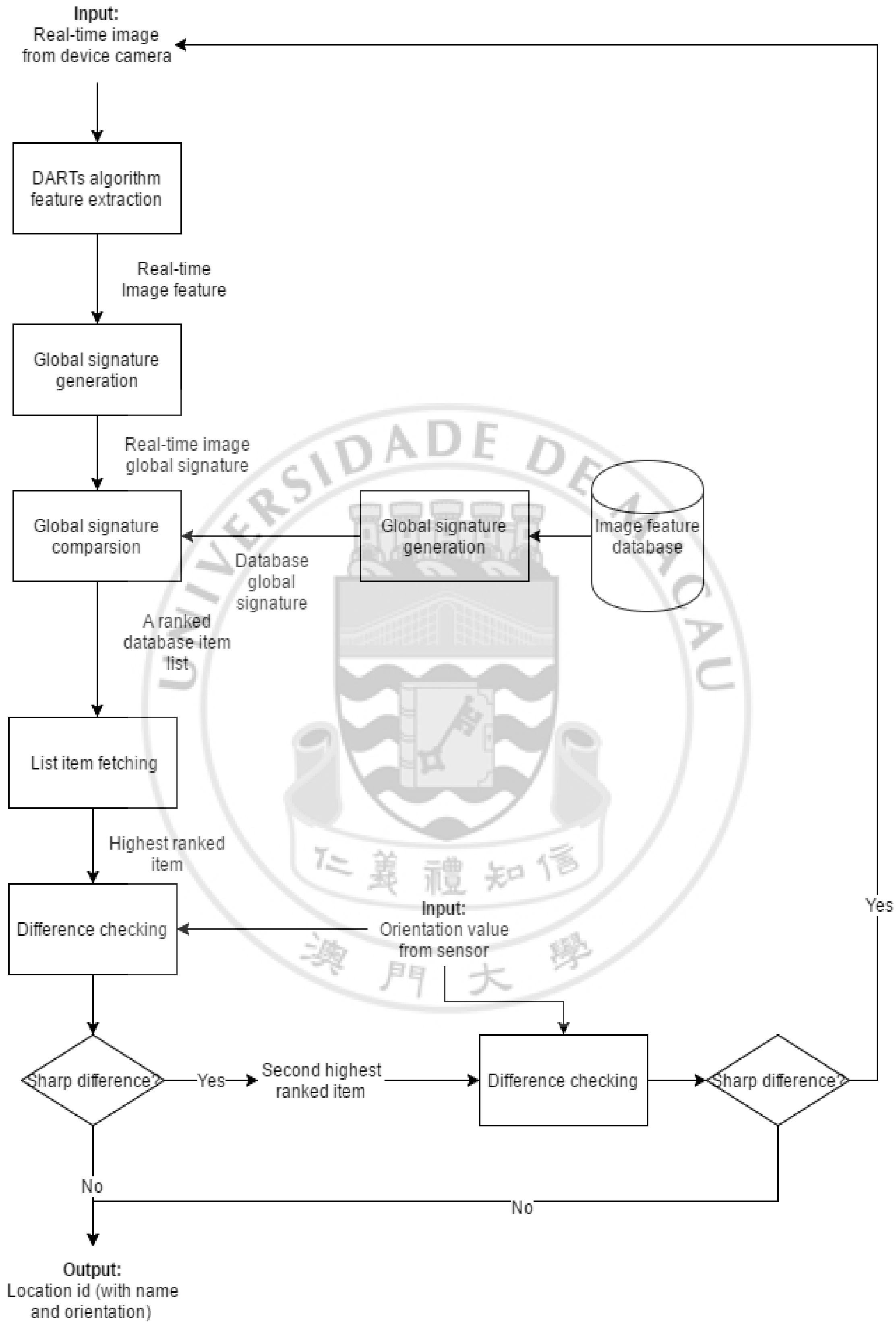


Figure 7: Framework of the feature extraction and recognition

A detail framework of the feature extraction and recognition modules is shown in figure 7. First, the real-time image of the user is retrieved from the user's device camera. Then the local image features are extracted from the real-time image. In our project, we proposed to use DART's algorithm to maximize the real-time performance of the application. It is possible to use other image features, for instance, the SURF, the SIFT or even other context-based image features. However, we have studied and concluded DART's algorithm result the best accuracy and a relatively fast response time. Next is to generate a global signature for the image feature. The reason of generating the global signature is also to maximize the real-time performance of the application. As comparing local features with the database feature is slow and require high computation cost, a global signature is used to reduce the computation cost. A global signature is a signature that can summarize and store the most important characteristics of the features. With global signature, it also has a compressed version of the entire image characteristics. However, the design of this global signature will be very essential to balance the size and the quality. If characteristics of images are too much, then the computation cost is also high. On the other hand, if the characteristics are too little, then the accuracy of the recognition process will decrease. So, a well-designed global signature is very important to be efficient in comparison, having a high-level of distinction for large number of images and also can have a fast access time in memory fetching.

Global signature generation is also needed for the image features in the database in order to result a fast comparison time. After comparison, it is essential to have a ranking list listing out all image id with the highest matching rank at the top. Although, the accuracy of the image recognition can be relied on, there are still chances for errors to happen. To ensure our application can reach the highest accuracy, there is a need to have a verification method to deal with the situation. After a ranking list is resulted, it will pass to the verification module for checking.

4.5 Design of the verification module

In the verification module, we designed to get assist from the orientation sensors that can be commonly found in Android devices. It is a sensor that can provide user the orientation information of the mobile device. However, the accuracy of this sensor is not very high. Through our testing, there is possible to have a maximum difference of 30 degrees. Although the accuracy is not high, the sensor can also be useful in validation. As the maximum error degree does not exist 30 degrees, it is accurate in measuring 4 geographic cardinal directions rather than 8. Once the highest ranked item is resulted, the system will check if the difference between the orientation of resulted item and the sensor's value is sharp. If the difference is not sharp, the ranked item is recognized as correct. However, if the difference is sharp, the system will check through the second highest member in the list. If the difference is still sharp, the system will request another real-time image from the device and perform a recognition process again. Although an extract verification module request extra time for the system to process, it is worth to sacrifice a few computation times to result a higher accuracy of the application. On the other hand, we also designed to use the sensors in the navigation part to provide continuous navigation effect. Once the verification is done, a reliable location id tag identifying the user's current position is resulted. The location name and orientation will be fetched by this id and are used to stimulate the navigation process. However, the navigation module is contributed by other members, so this report will not discuss further about how the design of the navigation module is.

To sum up, the image recognition module is designed to be fast and accurate. We have done a lot of readings and concluded the mentioned design to maximize the performance and accuracy of the proposed application. According to figure 3, it is essential for the image recognition module to response quickly and accurately since this module will be processed frequently during the navigation process. Image recognition module is kept on being processed to identify the user-location in run-time. If the computation cost of this module is high, the load of the system will become massive and may result a chance of application crash. Besides, we aim to provide a real-time application experience for the user, the loading process should be transparent at the user's point of view. An error or slow response time can greatly affect the acceptability and reliability of the entire application. As a result, the image recognition is essential to be designed with using a global signature with DARTs features and a validation module.

CHAPTER 5. Implementation narrative and description

This section covers the discussion of the algorithm explanation, coding, data structure and other justification of implementing the proposed application.

5.1 Implementation of background environment

Our proposed method is mainly implemented by Java language by Android Studio and executed on Android platform. All images used in constructing database are in JPEG type and the images are collected by an Android device's camera in 16:9 ratio. In our project, the data collection device and the testing device is consistent to maximize the effect of our proposed method.

5.2 Implementation of launching interface

As mentioned in the design section, our application consists a splash screen activity for the application to install the required database. We implemented a java splash screen activity to be in charge of the installation process. A delay variable (in ms unit) clarifies how long the screen holds for the database installation.

```
private static final long SPLASH_SCREEN_DELAY = 1000;
```

Then, the collection is loaded into the local site of the application device and update the database is updated and accessed once the collection loading process is done.

```
        mCollectionManager = CraftAROnDeviceCollectionManager.Instance();
        mCraftAROnDeviceIR = CraftAROnDeviceIR.Instance();

    public void collectionReady() {

        DatabaseUpdate databaseAccess = DatabaseUpdate.getInstance(this);
        DatabaseUpdate databaseAccess = DatabaseUpdate.getInstance(this);
    }
```

Once the database is ready, the first activity that the application shown is a list that allow user to search and browse destination. The items (string type) in the list are fetched from the installed database and shown to the user.

```
final DatabaseAccess databaseAccess = DatabaseAccess.getInstance(this);
databaseAccess.open();
List<String> ShopName = databaseAccess.getShopName();
```

As described in the design section, we planned to implement a searching function for the users that are already familiar with the place. The application listen to every typed text in the text box and search through the database based on the user's input. The variable "cs" is the character sequence of the input textbox and is filtered by the adapter. Then the destination list will be updated based on the accessing result. Once the destination is settled by the user (with a button click activity), the application will jump to the Main activity that are in charge of the image recognition module.

```
adapter=   new
    ArrayAdapter<String>(this,android.R.layout.simple_list_item_single_choice,
ShopName);
this.listView.setAdapter(adapter);

searchtext.addTextChangedListener(new TextWatcher() {
            @Override
```

```
            public void onTextChanged(CharSequence cs, int arg1, int arg2, int
arg3) {
                // When user change the text
                adapter.getFilter().filter(cs);
            }
}
```

After all the background setup is ready, we also need certain sensors to perform the validation module, the declaration of the use of these sensors are in the Main Activity of the application. However, the code of the usage of these sensors will not be included in this report due to the same reason mentioned.

```
sm = (SensorManager) getSystemService(SENSOR_SERVICE);
accelerometer = sm.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
magnetometer = sm.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
gravity = sm.getDefaultSensor(Sensor.TYPE_GRAVITY);
```

5.3 Implementation of the image recognition module

5.3.1 Collection of real-time image

The positioning required the real-time image of the user's location. We used the SDK provided by the CatchRoom to process the image recognition. In the SDK, there is a camera object that can control the device's camera. In the main activity, the camera is needed to be set ready before used.

```
mCraftARSDK = CraftARSDK.Instance();
mCraftARSDK.startCapture(this);
mOnDeviceIR = CraftAROnDeviceIR.Instance();
```

We also included an exception function that deal with the connection error of the camera to notice user and prevent the application from non-functioning.

```
    public void onCameraOpenFailed() {
        Toast.makeText(getApplicationContext(),         "Camera         error",
Toast.LENGTH_SHORT).show();
    }
```

Then the collection process will run frequently with certain seconds' interval. The "startFinder" is a function call that fetch the real-time image from the camera object and do the comparison with the image data collection. The variable "startFinderTimeMillis" indicates how long the interval of the fetching and searching process.

```
    private void startFinding() {
        mCraftARSDK.startFinder();
        startFinderTimeMillis = System.currentTimeMillis();
    }
```

Besides, as listed in the functional requirement section, we need to allow the user to terminate the navigation at any time he or she wants. As a result, we have a function that take care of this situation. It is implemented as a button activity. Once the user presses the button to terminate, an alert message box will pop up to confirm that the user's decision to prevent a mistake pressing action. If the termination is confirmed, the process is terminate with the function "stopFinding()" and kill the process to return to the destination list page. Otherwise, keeps on collecting real-time image with the "startFinder" function.

```
    public void onBackPressed() {
        AlertDialog.Builder isExit = new AlertDialog.Builder(this);
        isExit.setMessage("Confirm to exit?");
```

```
        isExit.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                mIsActivityRunning = false;
                stopFinding();
                android.os.Process.killProcess(android.os.Process.myPid());
            }
        });
        isExit.setNegativeButton("No", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                mIsActivityRunning = true;
                startFinding();
            }
        });
        isExit.show();
    }
```

5.3.2 DARTs feature algorithm

We proposed to use DART's image features for the local feature extraction. It is an algorithm proposed by Marimon, Bonnin, Adamek, Gimeno [18] and used in the CraftAR SDK to provide the image recognition function. DARTs feature is actually an improved version of the SURF and SIFT algorithm which resulted 3 times speed up performance compared with SURF and 6 times speed up performance when compared with SIFT [18]. The concept of the DARTs algorithm is similar with SURF or SIFT but with a different descriptor to speed up the performance. As DARTs algorithm is derived from SURF and SIFT, DARTs algorithm inherits the scale and rotation invariant characteristics from those two algorithms. It is also the reason that we chose DARTs algorithm for positioning. With the scale and rotation invariant characteristics, features of an object can still be recognized at different view point and distance from the object. It is the key idea in the positioning. In practical cases, it is impossible that the user is at a position that is totally identical with the position where the reference image is taken. Thus, the image recognition algorithm should have the ability to identify object in different distance and view point.

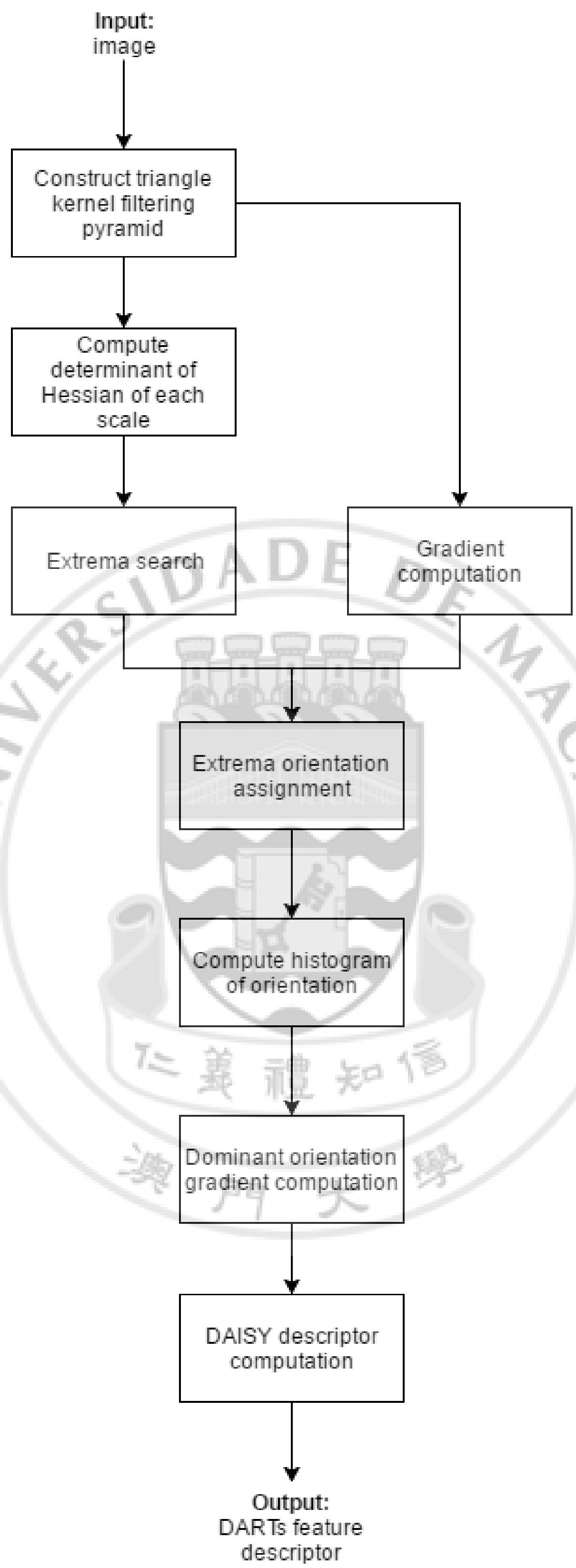


Figure 8: Framework of DARTs algorithm

The overall framework of the DARTs algorithm is shown in figure 8. Firstly, the input image is filtered by a triangle kernel to construct a pyramid. In SIFT algorithm, the process is to filter the input image by Gaussian function and construct the Gaussian pyramid.

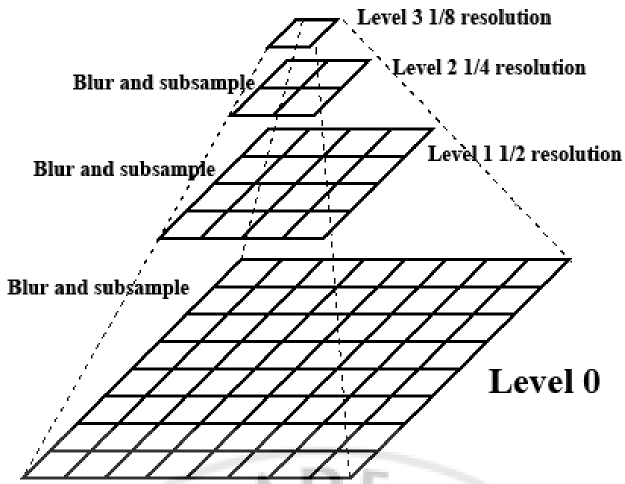


Figure 9: Gaussian pyramid

A Gaussian pyramid is a visual representation used for scale-space representation or multiresolution analysis. In the pyramid, the original input image is filtered by Gaussian filter, subsample with the original image and scale down by each level. An image representation of the Gaussian pyramid is shown in figure 9. Moreover, the number of level of the pyramid is decided by the input image. Different images have different number of layers. The number of layers, n is calculated from the following equation where M and N is the dimension of the input image.

$$\mathrm{n} = log_2\{\min(M,N)\} - t \qquad t\epsilon[0, log_2\{\min(M,N)\})$$

In SIFT algorithm, it makes use of the different scales of the Gaussian pyramid to search of the extrema points. The main idea is to search through the Difference of Gaussian of each level for extrema points. However, Difference of Gaussian has a characteristic of unstable edging response. These edge responses will affect the accuracy of the extrema searching process. As a result, Hessian matrix is used to eliminate the effect. After eliminating the responses, every pixel in each level is compared with all its neighbours and the neighbours of the two adjacent image scale levels to identify whether it is the maximum or the minimum among all compared neighbours. If the situation is satisfied, the pixel is marked as extrema.

DARTs algorithm shares a similar concept with the SIFT algorithm. DARTs also uses the image vision pyramid to search for extrema. However, to speed up the performance of the feature extraction process, DARTs algorithm proposed a 2D triangle shape filter to approximate the Gaussian function to reduce the computation workload. Moreover, DARTs algorithm constructs the pyramid without subsampling to enhance the accuracy of the features. Similar with SIFT, Hessian matrix is used to eliminate the unstable edging responses. The approximation of the second derivate of Gaussian and the computation of the determinant of Hessian matrix in DARTs is as follow:

$$\mathrm{H}\,\partial(\mathrm{i,j}) = \begin{bmatrix} \partial_{xx} & \partial_{xy} \\ \partial_{yx} & \partial_{yy} \end{bmatrix} \qquad \det(\mathrm{H}\,\partial(\mathrm{i,j})) = |\partial_{\mathrm{xx}}\partial_{yy} - \partial_{xy}^2|$$

$$\partial_{xx}^{k} = L(k, i - d1, j) - 2 \cdot L(k, i, j) + L(k, i + d1, j)$$

$$\partial_{yy}^{k} = L(k, i, j - d1) - 2 \cdot L(k, i, j) + L(k, i, j + d1)$$

$$\partial_{xy}^{k} = L(k, i - d2, j - d2) - \cdot L(k, i + d2, j - d2) - L(k, i - d2, j + d2) + L(k, i + d2, j + d2$$

where L is the 2D triangle filtered image, $d1 = \frac{2*\lceil 3\sigma \rceil + 1}{3}$ and $d2 = \frac{d1}{2}$. The value of d1 and d2 is decided through experimental testing from [18] to obtain the best approximation result of the second derivative of the Gaussian. $\partial_{xx}$and ∂_{yy} indicate the second derivative in horizontal and vertical respectively while ∂_{xy} is the cross derivative. ∂_{xy} and ∂_{yx} are identical for different order of the cross derivative. Besides, the extrema comparison in DARTs is identical with the SIFT algorithm.

Once the extrema pixels are all found, next is to assign an orientation for each pixel. To achieve the rotation invariant characteristics, an assigned orientation for the extrema can help to recognize an object even after rotation process. And the orientation is decided by computing the gradient of the pixel. In SIFT, the gradient $(\mathrm{m}(\mathrm{x}, \mathrm{y}))$and orientation $(\theta(\mathrm{x}, \mathrm{y}))$ is computed by the difference of the pixel value where L is the image.

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

$$\theta(\mathrm{x}, \mathrm{y}) = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}$$

To decrease the access times when computing the gradient and orientation, DARTs algorithm computes the gradient for each pixel by only two accesses. The derivate of x and y is computed by the following method where $d3 = \frac{2*\lceil 3\sigma \rceil + 1}{6}$

$$\partial_{x} = L(k, i - d3, j) - L(k, i + \mathrm{d3}, \mathrm{j})$$

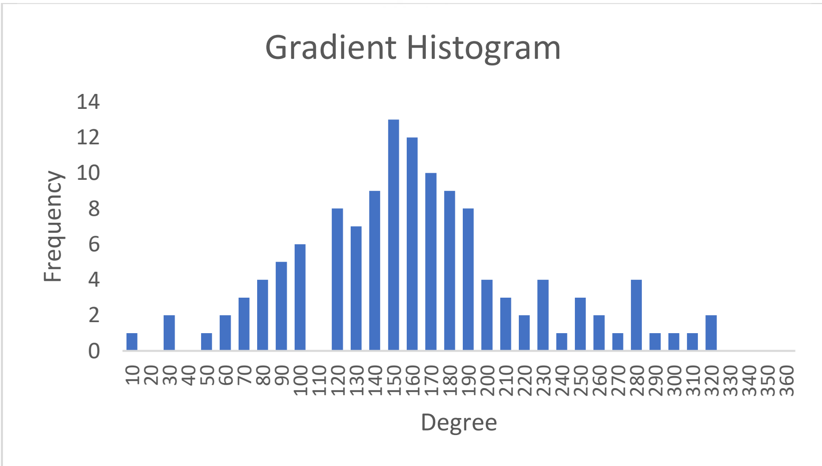


Figure 10: Example of gradient histogram

After the gradient of the extrema is computed, a histogram is constructed with the gradient of the extrema and all its neighbour. The gradient histogram is ranged from 0 to 360 degree, with an interval of 10 degree, totally 36 bins. An example of the gradient histogram is shown in figure 10. Then, the orientation with the vertex value of the histogram will be chosen as the main orientation for the extrema pixel. Besides, the dominant orientation operations are the same in DARTs and SIFT algorithm.

The final step of the DARTs algorithm is to construct a descriptor for the features. After all the computation, each extrema pixel contains three information: coordinate, magnitude and orientation. And a descriptor is a vector that represents this three information so that these features will not be affected by light, scaling or rotation process etc. The descriptor not only include the extrema point but also the relationship between the extrema and its neighbours. This can provide a unique information that can accurately recognize an object. The type of descriptor used in DARTs is very different with SIFT algorithm.

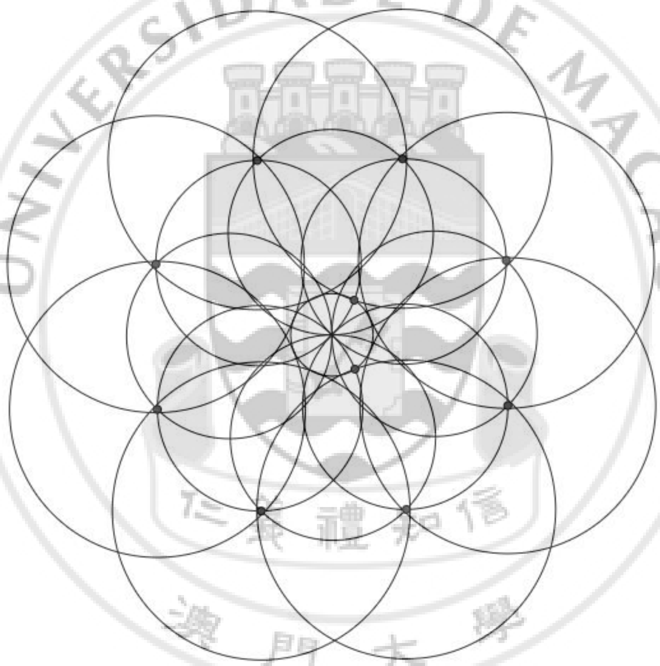


Figure 11: DAISY descriptor

In DARTs algorithm, it proposed to use a DAISY-like descriptor. The first step to construct a descriptor is to segment the neighbour of the extrema to regions. In SIFT, four regions are segmented. However, DARTs segment the neighbour into two rings and eight segments for each ring forming a daisy shape. DARTs proposes this number of rings and segments to produce the best result in the shortest length of the descriptor. The layout of the descriptor is shown in figure 11. The descriptor vector contains 68 values. The benefit of DAISY descriptor compare with the four-region descriptor of SIFT is that it makes use of Gaussian weighting and circular symmetric kernel that can speed up the performance of the computation.

The descriptor of DARTs is computed as follow: Firstly, first derivative computation is computed on neighbours around the key point. The derivative is oriented according to the orientation of key point from the previous step. The cost of computation step is low by taking only two accesses to the filtered image. This can greatly reduce the computation time to have real-time response.

$$\partial_x^\theta = L(k, i - d3cos\theta, j - d3sin\theta) - L(k, i + d3cos\theta, j + d3sin\theta)$$

The derivatives are then accumulated together to form a vector with four values extracted from the samples with a DAISY-like layout. Finally, the descriptor vector is processed by the L2 normalization and resulted the DARTs features.

5.3.3 Comparison and searching algorithm

The image retrieval process in Craft SDK is proposed by David M. Chen and Bernd Girod [19]. They proposed a method to retrieve image from database efficiently. The main workflow is shown in the design section. This section mainly focus on the detail of the process in the structure. To deal with a large amount of image data, they proposed to construct a vocabulary tree for all the image data.

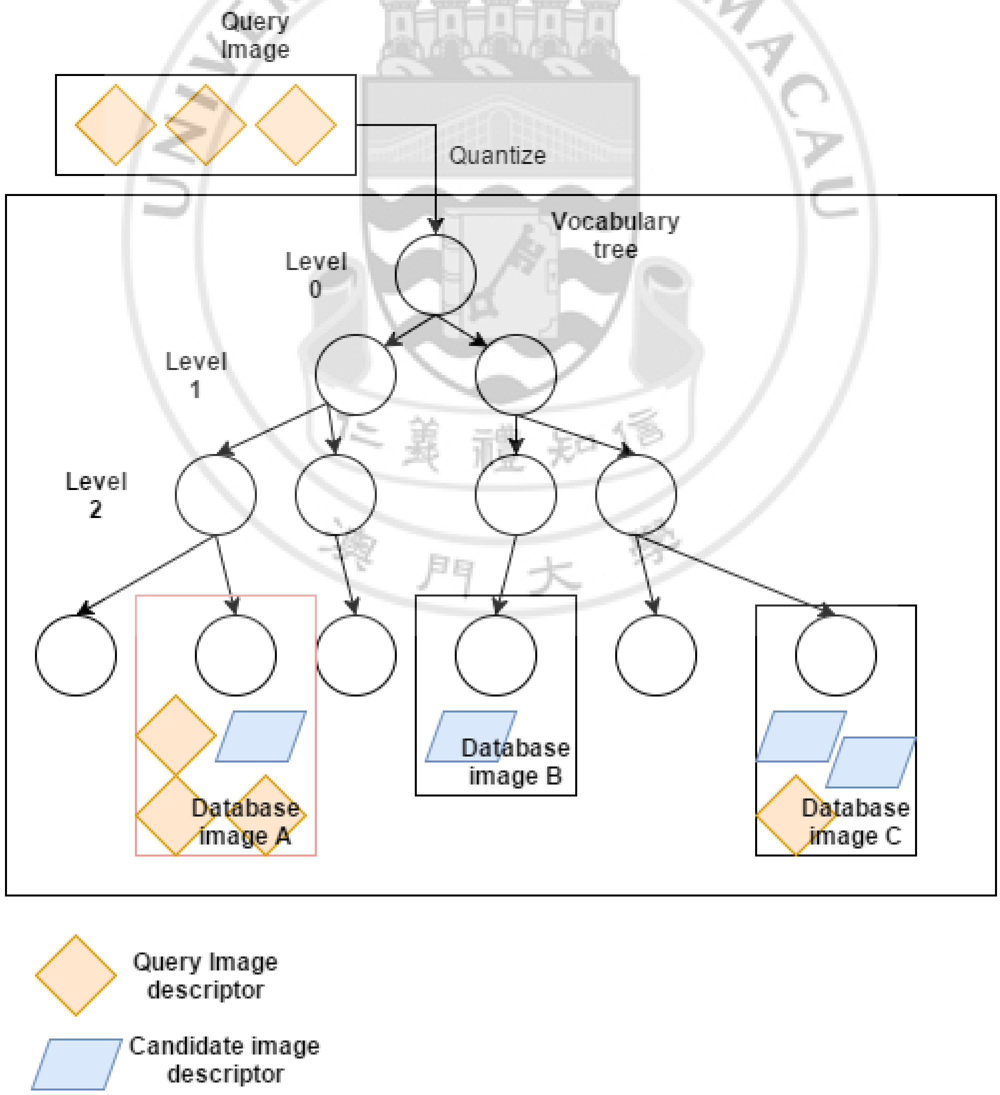


Figure 12 Vocabulary tree of image pairing

Figure 12 show a representation of an example vocabulary tree. However, as direct comparison between image and descriptor is a heavy workload when the image database is large. A global signature is more efficient in comparison. On the other hand, as the RAM of mobile device is limited, it is optimized to use the memory in an efficient way. As a result, an algorithm called REVV signature is used as the generation of the global signature. According to [19], REVV algorithm can have a higher level of image retrieval performance than other methods discussed in [19].

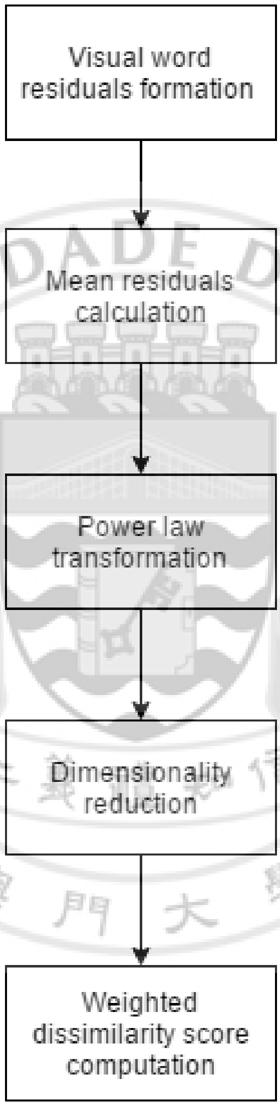


Figure 13: Workflow of REVV algorithm

The workflow of the REVV algorithm is shown in figure 13. Firstly, the image features are quantized to form a set of visual words. Visual word is a set of pixels that contain the main and interesting features of an image. After the visual words are resulted, calculate the mean of the residual of each of the visual words. Then apply a power law transformation on the resulted vector to reduce the unfavourable peak values. The following is a general expression of a power law transformation where s is the output bin value, r is the input bin value, c and γ are any real numbers.

$$s = c * (r^{\gamma})$$

After then, the resulted vector is transformed by a linear discriminant analysis eigenvectors. Linear discriminant analysis is a dimensionality reduction technique and eigenvector is a non-zero vector that remain direction unchanged after linear transformation. The main use of the LDA is to reduce the computational cost of computing the input vector. Finally, the resulted vectors are constructed to a stream of bits which can have a less memory consumption.

All the image in database and the query image will be processed by REVV and resulted a global signature for matching. The comparison will be processed based on the generated global signature rather than a direct feature comparison.

5.3.4 Implementation of positioning result

After the comparison of the query image and the image database, an array list with a sequence of potential location is resulted. However, the result list from the CraftAR sdk is not sorted. We need to have it sorted. The following code is used to find the element with the highest score. The score of every element is fetched and compared with the variable "tmpScore" which stores the highest score among all the previous entries. After the loop is finished, the item name with the highest score is resulted.

```
Integer tmpScore = results.get(0).getScore();
CurrentPlaceName = results.get(0).getItem().getItemName();

for (CraftARResult result : results) {
    String itemName = result.getItem().getItemName();
    Integer score = result.getScore();
    if (score > tmpScore) {
        tmpScore = score;
        CurrentPlaceName = itemName;
    }
    resultsText += itemName + ": " + score.toString() + "\n";
}
```

Once we have the highest ranked item, the item id is passed to perform validation and start navigation. The implementation detail and the algorithm of the remaining module is not discussed in this report as it is the contribution of my project groupmates.

CHAPTER 6. System Quality

This section mainly discusses the quality, validation and testing of our proposed application.

6.1 Testing environment and background

We implemented an Android application to test our proposed indoor navigation structure. We chose a very large shopping mall in Macao as our data collection and testing environment. The overall area of the shopping mall is about 90000 m^2 and we selected one third of the overall area for testing with area of around 30000 m^2. There are several reasons for us to select this shopping mall as our first testing environment. As it is a very popular shopping mall in Macao, there are many visitors walking around in the indoor environment. Visitors can be a variable influence the feature of the collected image and affecting the recognition accuracy. If our proposed recognition algorithm can work well even in such a crowd environment. Then our proposed method can absolutely be successful in other indoor environments. Another reason is that we want to ensure the application can work for large indoor space. We want to ensure that the application can still provide real-time searching effect for large image database.

The device we used for testing is a HTC M8 with an operating system of Android 6.0. During testing, the internet function of the device is shut down to ensure the application can function well in offline environment. Besides testing the application with our group members, we also invite some people to use our application to test the acceptability and the effectiveness of the application.

6.2 Positioning testing

Before testing the application, we also have done testing on the accuracy of the positioning function. We tested the recognition function by randomly picking location in the shopping mall and justify whether the application can correctly recognize where the place is by the real-time image.

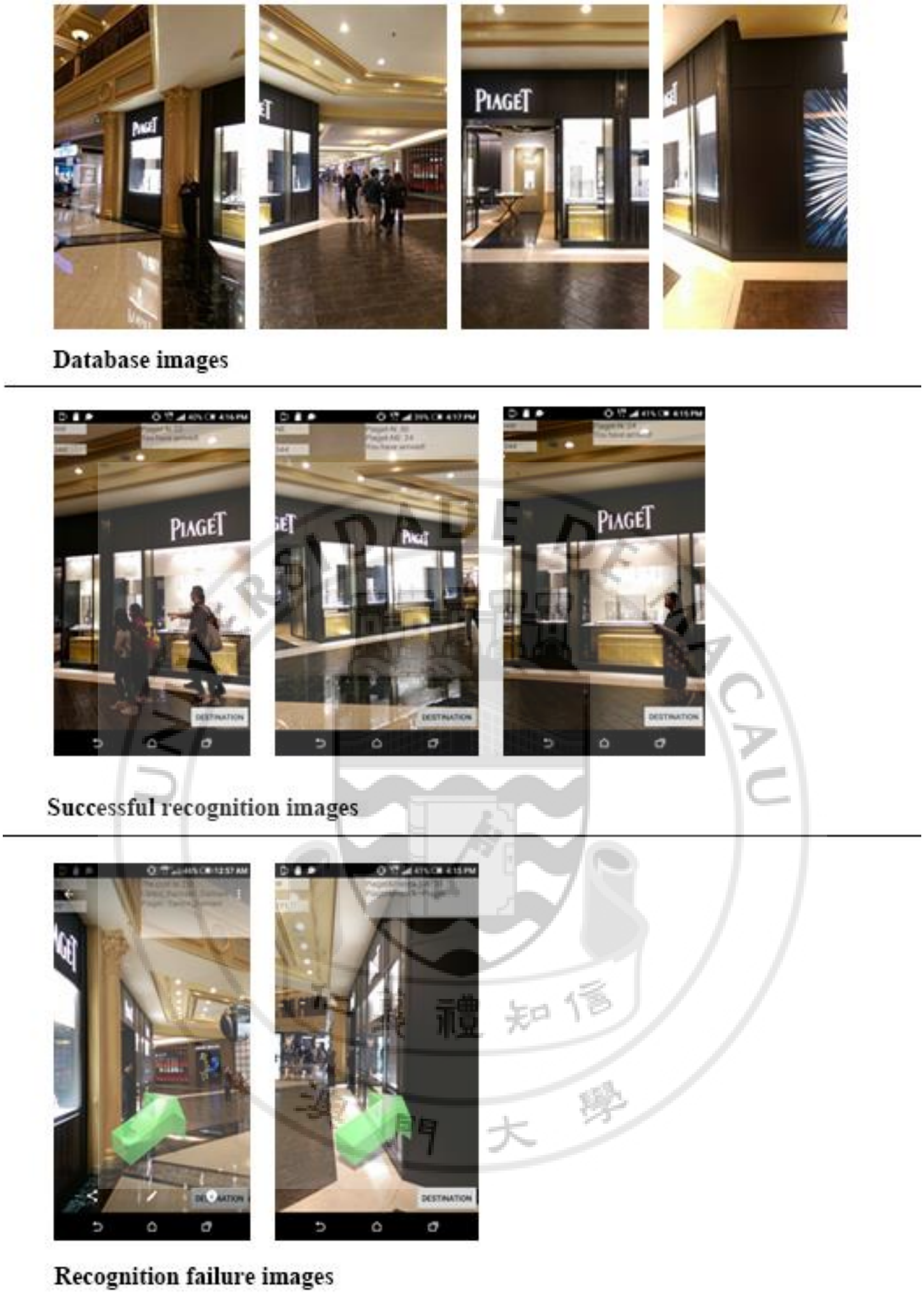


Figure 14: Sample images of the positioning result

Figure 14 shows some screenshot of our positioning testing. The database images of the related location are also shown for a comparison. From the result, we can know that the failure of the recognition process occurs when user is standing near the boundary of the road. The reason for the failure to occurs is that there is lack of information that can identify the position. For instance, the logo of the shop cannot be seen clearly. However, this problem can be solved by collecting more images at such view point to let the system cover such boundary areas. On the other hand, from the successful result, we can know that the system can recognize the location even the real-time images are at a different view point, distance, or angle with the collected database image. The system can also recognize the location successfully even when some people get caught in the images and block part of the environment. It shows that our proposed application can still function well in crowded environment.

Table 5: Positioning result of 20 randomly picked location

Location Name	1	2	3	4	5	6	7	8	Success Rate (%)
Shop A	✓	✗	✓	✗	✓	✓	✓	✓	75
Shop B	✗	✓	✓	✓	✓	✓	✓	✓	87.5
Shop C	✓	✓	✓	✗	✗	✓	✗	✓	62.5
Shop D	✓	✓	✗	✓	✓	✓	✓	✓	87.5
Shop E	✓	✓	✓	✓	✗	✓	✓	✓	87.5
Shop F	✓	✓	✗	✓	✓	✗	✓	✓	75
Shop G	✓	✗	✓	✓	✓	✗	✓	✓	75
Shop H	✓	✓	✓	✓	✓	✗	✗	✓	75
Shop I	✓	✓	✗	✓	✓	✗	✗	✓	62.5
Shop J	✗	✓	✓	✓	✓	✓	✗	✓	75
Shop K	✓	✗	✓	✓	✓	✓	✓	✓	87.5
Shop L	✓	✓	✓	✓	✓	✓	✓	✗	87.5
Shop M	✓	✓	✓	✓	✗	✓	✗	✓	75
Shop N	✓	✓	✓	✓	✓	✗	✓	✓	87.5

Shop O	✓	✓	✗	✓	✗	✓	✓	✓	75
Shop P	✓	✓	✗	✓	✗	✓	✓	✓	75
Shop Q	✓	✓	✗	✓	✗	✓	✓	✓	75
Shop R	✓	✓	✓	✓	✓	✓	✓	✓	100
Shop S	✓	✓	✗	✓	✗	✓	✓	✓	75
Shop T	✓	✓	✓	✓	✓	✗	✓	✓	87.5
Average Success Rate (%)									79.37

using ✓ to indicate successful recognition of the location

using ✗ to indicate failure recognition of the location

1-8 indicate the number of trials

Table 5 shows the testing result of the position recognition in 20 different location. The tested locations are randomly picked in order to increase the reliability of the result. From the table shown, the success rate of the proposed recognition method is high. However, there are still several failures during the test. As mentioned above, these failures occur at location that cannot provide enough image features. To sum up, to improve the rate of success rate, the solution is very simple. We can collect more images to enhance the accuracy and reliability. And to enhance the quality, efficiency, and cost for collecting image data, a robot car with a camera recorder can be used. The robot can walk through the indoor environment once and record video of the surrounding environment. Images can be fetched from the video to form the image database. With this method, the cover area of the indoor environment can be larger, increasing the performance of our proposed application. Moreover, the human power used in collecting image data can also be greatly reduced.

6.3 Navigation testing

Besides testing the positioning function of our proposed application, more importantly is the testing for the overall function. We randomly picked different starting location and destination to do the test. It is important that the places are picked randomly to ensure the reliability of the application in different circumstances. Our testing also includes users that are outside of our implementation group to make sure that the application is easy to use and the direction guide is clear and understandable.

Table 6: Result of navigation testing

From (Shop)	A	Q	R	L	D	M	S	C	F	B	H
To (Shop)	T	B	H	E	Q	A	E	S	O	I	T
Average success rate (%)	80	90	100	90	100	100	100	80	100	100	100
Total average success rate (%)									95		

Table 6 show the result of our navigation testing. Each pair of location is tested 10 times. If any direction mistake occurs in the navigation, that test is marked as failure. Otherwise, if the user can reach the location successfully, that test is marked as success. The average success rate is the average of the success times in 10 tests for each pair of location. Finally, all the average success rate is accumulated and take average to result the final total average success rate. From the testing result, it shows that our proposed application can reach a high successful rate in navigation. The reason that the overall system can have a higher accuracy than the recognition is that we have implemented process to deal with the situation of the recognition failure. The system does not stop navigation even if the position cannot be recognized during the process. With the help of the device sensors, we can still know the approximate orientation of the user. Based on the orientation and the route, the system can still provide temporary navigation until a position is recognized. From the navigation test result, it shows that the temporary recognition failure handler does not affect the accuracy of the entire navigation process and even provide a continuous navigation effect for the user.

On the other hand, there are still several failures occur during the testing. We recorded down the situation and discovered the main cause of the problem. Some failure occurs when the location is extremely crowded. The camera cannot collect any features of the indoor environment but only pictures with a lot of people. If this circumstance only lasts for a short period of time, the recognition failure handler can deal with the problem. If the situation lasts for too long, the navigation will be entirely depended on the device sensors other than the image of the user’s current environment. As the accuracy of geometric sensors on mobile device is not precise, it can only provide approximate orientation. Moreover, if the situation lasts for too long, the location of the user cannot be updated on the route. The system will then assume that the user is staying at the same location and providing the same direction guide. As a result, an unreliable or a wrong direction will be shown to user until the recognition process works again.

Besides, failure also occurs when user points the device to the floor rather than to the surrounding environment. As the floor cannot provide unique feature to identify a location, the recognition process also fails and results a similar mentioned problem. As people used to hold the mobile device in a way that the camera is facing the ground when walking, this may be one little user constraint of our proposed application. Since our application relies on collecting unique features of surrounding environment by camera, it cannot work when the camera is pointing to the ground. Although we do not want to have any constraint on user, it is inevitable that we need some reminding message to tell the user to hold the device in a way that the camera is facing the surrounding environment for the application to work.

6.4 Result comparison

Table 7: Comparison with existing indoor navigation methods

Location Name	Image recognition	Wi-Fi fingerprinting	Image Marker
Continuous Localization	✓	✓	✗
Network Requirement	✗	✓	✗
Positioning Constraint	✗	✓	✓
Require Pre-installed equipment	✗	✓	✓
Error Rate	5%	10%	2%

The comparison of our proposed method with other indoor navigation method is shown in Table 7. We chose one visual and one non-visual based method for comparison. Both of our proposed method and the Wi-Fi fingerprinting can provide continuous localization effect. However, our proposed method does not require network access, pre-installed marker, or other special equipment. Indoor navigation by image recognition is also invariant with the attenuation effect of the electromagnetic signal. On the other hand, although image marker method results the lowest error-rate, it has user constraint that requires user to discover the marker and scan it with the device-camera. Besides, our proposed method only need a little user-constraint that require the user holding the device in a way that the camera is pointing to the surrounding rather than to the ground. It is more convenient for user to use and the only requirement is to install the application in the mobile devices. Moreover, our proposed method has a lower error rate compare with the Wi-Fi fingerprinting method and the accuracy can be easily enhanced by the amount of environment image in the database.

6.5 Overall validation and improvement

To sum up, our proposed application has achieved the objective and successfully perform navigation in indoor environment. The acceptability of our implemented application is also high. Users can easily understand how to use the application. Moreover, our project and result is also recognized by publishing in the 9th International Conference on Digital Image Processing (ICDIP 2017). However, through the testing results, our project concluded that there is still room for improvement of our proposed method. The first thing that need to improve is the accuracy of the recognition result. The simplest solution is to collect more image data of the indoor environment. However, using human forces in collecting image data increase the cost of implementing our proposed method. Thus, we plan to collect image data through video taking rather than taking images one by one. It is also possible to use robot to reduce the cost of collection. The second thing that need to improve is to have a more reliable recognition failure handler. The handler of our proposed method cannot update the location of the user. In the future, it is possible to include one more sensor value which calculate the distance or the speed the user travel in the handler. By including the distance between each node in the map graph and the value obtained from the distance or speed sensors, the handler can approximately update the user location even if the recognition fail for a long time. By minimizing the chance of recognition failure and increasing the reliability of the failure handler, our proposed method will be more accurate and reliable in indoor environment navigation.

CHAPTER 7. Ethics and Professional Conduct

7.1 Copyright issue

In our project, we have used some open source library and sdk that is provided by the CatchRoom to implement the recognition function. It is unethical if we used their efforts and coding without giving credits to them. In our project and report, we give credit and thanks for the effort of these external source provider. Moreover, our project is only used for academic research purpose. The sdk and library used is for providing assistant in testing our proposed ideas and method without earning any self-benefits. It is unethical to use others' work for commercial usage. If our project is recognized by some commercial company and being invited for commercial usage, we will remove any external source and implement all the function by ourselves. It is a promise from all our members in the final year project group to protect the copyright and benefits of all the external source provider that we used in our academic project.

7.2 User privacy information

As we proposed an indoor navigation application, it is inevitable that the system can track where the user is. Moreover, it is also possible to analysis the user's habit and favour based on the location that the user visits through our proposed application. This information is beneficial for different commercial companies to settle strategies for advertisement. Some big social media platform track user's activity and use the information to attract companies to pay for advertisement. Thus, the information of the user's habit and preferences can be valuable and possible to make money. However, it is unethical to monitor and collect data of user's habit with the user's permission. It is also unethical to force user to have agreement being monitored in order to use the application. It is also unethical to record the current location of the user for other usages. Thus, our proposed application does not have record on the user's activity. To protect the user's privacy and right, our application does not require any privacy information agreement to use. Our project group promise that we will not keep track of the user's habit and sell user's information for commercial benefits.

7.3 Portrait right

As we proposed to use real-time images of the user to perform indoor navigation, it is inevitable that our application can have access and control of the user's device camera. During collecting real-time images from users, it is possible that the appearance or faces of the users or different people is captured. Anyone person has the right to protect his or her own portrait. It is unethical to store or user other's portrait without permission or notice. It is also unethical to do so even not for commercial use. Thus, our proposed application will not save or send any of the captured real-time images. Once the positioning process is finished, the real-time image of the user is deleted.

7.4 Security responsibility

As our proposed application is an indoor navigation application, it is inevitable that the system can identify where the user is. We want to develop an application that can assist people rather than harm people. Similar with the situation mentioned in 7.2, the user's habit information is valuable. Also, our application has access and control of the user's device camera, it is possible that the images of the camera can be accessed through our proposed application. Thus, it is unethical that the application will leak user's information to third party when using. It is also dangerous that if the image of the user's camera can be peeped by someone with evil intentions. It is also unethical that the implementation team do not ensure the implemented application is secured and release an unsecured application to the users. As the developer of the application, it is our responsibility to ensure that our proposed application is secured to protect all the user's privacy information otherwise it is our fault.

CHAPTER 8. Conclusion

In this paper, our final year project group have proposed an indoor navigation application by image recognition to overcome the navigation challenges in indoor environment. We propose to use the feature of the user's surrounding environment for location and orientation identification. We aim to propose a high accuracy, efficiency, easy-to-use and reliable method to achieve indoor navigation. Our proposed application can be classified into three main modules: positioning, navigation, and direction rendering.

Positioning is achieved by using image recognition method. The structure of an indoor environment is constructed into an image database. The real-time images of the user's current location are collected and compared with the database to identify the user's location. To achieve the recognition and searching, DARTs descriptor is used to extract location's features and REVV algorithm is used to generate global signature for efficient storing and searching. DARTs descriptor is accurate and has a low computation cost in descripting and providing unique features in images. It can provide high accuracy of recognition result with a real-time response. DARTs feature is mainly a speeded-up version of SIFT algorithm. It shares the main concept of the SIFT algorithm but revised to use 2D triangle filtering function for approximation and a DAISY-like descriptor to describe maxima features and orientation. With the revised improvements, DARTs algorithm is outstanding to be used in fast-response image recognition. Thus, we used DARTs algorithm to extract image features for image recognition in our application. On the other hand, we chose REVV algorithm to perform fast and efficient searching in mobile devices. With REVV algorithm, image features can be stored efficiently in limited RAM of mobile devices when processing. REVV algorithm includes power law transformation and LDA to reduce the size of the image features and the computation workload in image retrieval. Thus, it is suitable retrieving algorithm to be used for low order hardware. With the combination of DARTs and REVV algorithm, the image recognition process can be accurate and response in real-time.

Navigation is achieved by using map graph and Dijkstra 's algorithm. Dijkstra's algorithm is used to perform the shortest path searching on the indoor map graph and return a route to destination. Navigation is performed based on the result from the positioning module and the shortest route. Repeated positioning processes and route updating are performed during the process.

Rendering is achieved by integrating direction guideline with the real environment on screen to provide clear instructions and user-friendly experience to increase the acceptability of the application.

Lastly, we demonstrate the effectiveness and accuracy of our proposed navigation method in the system quality section in this report. Our proposed application can successfully guide user to destination in indoor environment. Moreover, in terms of continuous localization and unconstrained condition, our proposed method have better performance compare with other indoor navigation methods. The advantage of our proposed indoor navigation method is that our method is independent of internet connection and the accuracy will also not be affected by attenuation of electromagnetic waves caused by obstacles. The disadvantage is that images of the indoor environment is needed to be collected before use. Besides, the success of our project is recognized by the 9th International Conference on Digital Image Processing. Our future improvement will focus on collecting more image data to increase the success rate of the recognition process, implementing a more reliable recognition failure handler, speeding up the recognition computation time and providing the user additional information of the indoor environment during the process of navigation.

CHAPTER 9. REFERENCES

[1] R. Harle. "A Survey of Indoor Inertial Positioning Systems for Pedestrians", IEEE Communications Surveys & Tutorials,1281 - 1293 (2013).

[2] Liu, Hui, et al. "Survey of wireless indoor positioning techniques and systems." in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2007. pp. 1067-1080.

[3] Smit, Martin, and Richard J. Barnett. "A comparison of augmented reality indoor navigation systems with traditional techniques." in Proceedings of 2012 Annual research conference of the South African Institute for computer Scientists and information Technologists. 2010.

[4] Robert Harle. " A Survey of Indoor Inertial Positioning Systems for Pedestrians", IEEE Communications Surveys & Tutorials, vol.15, no.3, pp. 1281 - 1293, Jan 2013.

[5] Liu, Hui, et al. "Survey of wireless indoor positioning techniques and systems." in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2007. pp. 1067-1080.

[6] Randell, Cliff, and Henk Muller. "Low cost indoor positioning system." in International Conference on Ubiquitous Computing. Springer Berlin Heidelberg, 2001.

[7] P. Bahl, V. Padmanabhan, "RADAR: an inBuilding RF-based user location and tracking system", 2000. Proc. in IEEE INFOCOM, vol. 2, pp. 26-30.

[8] Davide Merico and Roberto Bisiani. " Indoor Navigation with Minimal Infrastructure", in Positioning, Navigation and Communication: WPNC '07. 4th Workshop on, Mar 22, 2007. IEEE, 2007

[9] Klopschitz, Manfred, and Dieter Schmalstieg. "Automatic reconstruction of wide-area fiducial marker models." in Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. IEEE Computer Society, 2007. pp. 1-4.

[10] Tollmar, Konrad, Tom Yeh, and Trevor Darrell. "IDeixis–searching the web with mobile images for location-based information." in International Conference on Mobile Human-Computer Interaction. Springer Berlin Heidelberg, 2004. pp 288-299

[11] Kim, Jongbae, and Heesung Jun. "Vision-based location positioning using augmented reality for indoor navigation." IEEE Transactions on Consumer Electronic, vol. 54, no.3, pp. 954-962, 2008..

[12] Christian Kray, Christian Elting, Katri Laakso and Volker Coors. " Presenting Route Instructions on Mobile Devices ", in Proceedings of the 2003 International Conference on Intelligent User Interfaces, Jan 12-15, 2003, Miami, FL, USA. ACM, 2003

[13] Mulloni, Alessandro, Hartmut Seichter, and Dieter Schmalstieg. "Handheld augmented reality indoor navigation with activity-based instructions." in Proceedings of the 13th international conference on human computer interaction with mobile devices and services: Mobile HCI 2011, Aug 30 - Sep 2, 2011, Stockholm, Sweden. ACM, 2011. pp. 211-220.

[14] Brush, A. J., et al. "User experiences with activity-based navigation on mobile devices." in Proceedings of the 12th international conference on Human computer interaction with mobile devices and services: Mobile HCI 2010, Sep 7-10, 2010, Lisbon, Portugal. ACM, 2010. pp. 73-82.

[15] Müller, Hans Jörg, Johannes Schöning, and Antonio Krüger. "Mobile Map Interaction-Evaluation in an indoor scenario." in Informatik 2006: Informatik für Menschen, 2006, Dresden. pp. 403-410.

[16] Rehman, Umair." Augmented Reality for Indoor Navigation and Task Guidance ". Diss. University of Waterloo, 2016.

[17] Zhong, Andrew. "Improving User Experiences in Indoor Navigation with Augmented Reality.", 2014.

[18] D. Marimon, A. Bonnin, T. Adamek. and R. Gimeno, "DARTs: Efficient scale-space extraction of DAISY keypoints," In Computer Vision and Pattern Recognition (CVPR), IEEE Conference, 2010, 2416-2423

[19] David M. Chen and Bernd Girod. "MemoryEfficient Image Databases for Mobile Visual Search." IEEE MultiMedia, Jan.-Mar. 2014, volume: 21, Issue: 1. pp. 14-23.

CHAPTER 10. Appendix

This section listed all the software and hardware used during the implementation of the project.

10.1 Project Planning

Table 8: Project planning chart

Task name	Start (Y/M)	Finish (Y/M)
Related Field study	2016/10	2016/12
Topic decision	2016/11	2016/11
Define objectives	2016/12	2016/12
Design application structure	2016/12	2017/01
Algorithm research	2017/01	2017/02
Interface prototype	2017/02	2017/02
Design map graph	2017/02	2017/02
Data collection	2017/03	2017/03
Database implementation	2017/03	2017/03
Implement alpha version	2017/03	2017/04
Testing alpha version	2017/04	2017/04
Implement revised beta version	2017/04	2017/04
Testing beta version	2017/04	2017/04
Implement final version	2017/04	2017/04
Testing final version	2017/05	2017/05

10.2 Software requirement:

- Android studio
- DB Browser for SQLite
- Adobe Photoshop CC

10.3 Hardware requirement:

- HTC M8
- Samsung Galaxy Tab 2