INDOOR MOBILITY

Full Phases of Indoor Navigation Services

Tutor: Prof. Gianmario Motta

Kaixu Liu 刘恺序
kaixu.liu01@universitadipavia.it

Contents

Figures

Tables

Abstract

Over the last decade, navigation systems have been widely used as resources for path planning and way finding. Normally, they are tracked and guided by specific positioning technologies. Outdoor navigation services have been developed over the years by GPS (Global Position System), which is a quite common precise and infrastructure-free solution in large environments. However, the available navigation services in indoor spaces do not complete, and no standard has developed to fulfill all indoor navigation requirements. In this thesis I discuss about indoor navigation requirements, like this question “How can a human being find his/her destination in a building he / she does not know?”, etc.

Here I propose an indoor navigation system that contains the functional modules and related techniques. The functional modules are Indoor Mapping, Indoor Positioning, Path Planning, En-route Assistant and Analysis respectively, which support the full phases of indoor navigation services.

Indoor Mapping: A 2D and 3D indoor mapping approach is proposed, which 1) can build an effective 2D SVG map that updates selectively, 2) converts it into a 3D virtual scene automatically at back-end, 3) includes an information management system at back-end, 4) renders and interacts efficiently and effectively on a smart phone, and 5) offers a vivid virtual indoor visualization for virtual navigation services at front-end.

Indoor Positioning: I propose two magnetic field positioning techniques: Robot Simulation-oriented Mobile Localization (RSML) and Computer Vision-oriented Fingerprint Localization (CVFL). 1) RSML focuses on a probabilistic framework that measures the relative and absolute position to simulate a robot movement. I propose an integrated magnetic field positioning approach based on eXtended Particle Filtering (XPF) algorithm, which fuses magnetic fingerprints, Wi-Fi fingerprints, and Pedestrian Dead Reckoning; 2) CVFL focuses on Magnetic Fingerprint Imagerization (MFI). It converts fingerprints to images and classifies them by CNN training.

Indoor Path Planning: Wayfinding is necessary to reach the destination, which includes route planning and path navigation. An optimized ant colony algorithm is developed to avoid the obstacles (walls).

En-route Assistant: En-route Assistant module provides the services in various devices, handed system, wearable system, etc. can be used for an enhanced navigation. In real-time navigation, it provides instructions based on a convenient trip for citizens and safe trip plan for VIPs (Visually Impaired People) and MIPs (Mobility Impaired People).

Indoor Data Analysis: Here, a POI recommended algorithm is proposed based on social relations. It uses social relations to enhance the accuracy of the recommended algorithm from a large number of user behavior data in social network. Indoor Mobility also provides a social relationship mining model to recommend POI by a classification mapping between social relationship and POIs.

Chapter 1 Introduction

1.1 Background

Over the last decade, navigation systems have been widely used as resources for path planning and way finding. Normally, they are tracked and guided by specific positioning technologies. Outdoor navigation services have been developed over the years by GPS (Global Position System) [1], which is a quite common precise and infrastructure-free solution in large environments. This effort has been only focused on outdoor navigation systems, like Google Maps, etc. However, the available navigation services in indoor spaces does not complete [2], and no standard has developed to fulfill all indoor navigation requirements.

In this thesis I discuss about indoor navigation requirements, like this question “How can a human being find his/her destination in a building he / she does not know?” This is the case of a student searching the office of a new professor, of a passenger crossing a new airport, of a young woman searching a shop in a huge mall; and, also, of an impaired person trying to find his /her way in a multi-floor building. The question is fascinating and socially relevant, because of the peculiarities and challenges of the indoor environment.

1) In real cases, airports passengers continue to grow [3], the total passengers of some international airports increase around 5%. In the future twenty years airports shall face a doubling passenger volume by 2030 [4]. In Beijing, China, 8 major department stores locate in Xidan Commercial Street. Each mall serves daily thousands shoppers [5], especially in holidays and weekends. In London Underground passengers increased 29% from 2010 to 2014. Traffic at 36 stations increased over 25% in the past five years [6], etc.;
2) Since buildings may have multiple floors, indoor is a 3D environment (outdoor map is typically 2D);
3) In complex indoor layout, users do not know where they are and sometimes get lost. System should provide the positioning service to guide users. Additionally, GPS cannot work indoor completely because it relies on satellite signal strength. Hence, alternative positioning systems should be considered;
4) Different stakeholders have different user needs. For instance, a young people pursue an efficient life to get a short path; an impaired people or an old people needs a safe path that avoid static or dynamic obstacles on their way to go to the destination;
5) In many indoor environments, a part of a building hides another part, and users cannot see their final destination nor the way to it [7]. The user should rely on directions, even if he/she is familiar with the environment (people who are familiar with a library, still have no idea

where is the book they are looking for in the library). An en-route assistant is needed to facilitate users' navigation;

6) Indoor maps are typically proprietary and not standard, and their conversion into a standard format easy to process is a key issue.

Facing such challenges and research questions, I make a survey of current status of Indoor Navigation system. GROUPING [157] is a typical indoor navigation system with geomagnetism positioning and support positioning and path planning services. The system has been tested with Google Maps Indoor (GMI) and GROUPING map. In test, GROUPING Map construction proves to be efficient and better than GMI, which relies on WiFi with related localization errors. QoE [158], not quite a complete system, to position the user based on the quality of local wireless networks. Specifically, it is an indoor navigation solution based on wireless local area networks. It evaluates the quality of user experience among three multimedia applications (Skype, YouTube and Hichannel). No matter where users locates, the system will guide users to a space that satisfy them with better service. Although QoE's scope is rather narrow, Wifi assessment could be relevant as a complementary technique for positioning in Wifi intensive areas. Ahmetovic, Dragan, et al. [159] use sensor networks of Bluetooth Low Energy (BLE) beacons to achieve precise localization and accurate guidance for visually impaired. A robot Indoor navigation is proposed by Sprunk, Christoph, et al. [160]. It provides a community way to measure the ground truth and formulates an experimental protocol for benchmarking robot navigation. Then they conduct experiments by using different robots to validate the benchmark.

Even if current indoor navigation systems either integrate the existing technologies or propose new methods to support navigation, however, none of current single indoor navigation systems propose complete navigation services to support different stakeholders. In this thesis, I present full phases of indoor navigation services (Figure 1−1), which address an integrated indoor pedestrian navigation system. Indoor services typically run on a smart phone and rely on a server center and embrace indoor mobility. Such Indoor Navigation Services include various sets that some are contributed by myself and some are by teamwork that guided by me. Specifically:

- Mapping manages maps, a master information that is stable during a reasonable time (actually the maps of a city or building do not change every second). The Indoor Mapping approach is contributed by myself. I use SVG, a standard format, to mapping indoor 2D maps, and convert SVG to 3D automatically as indoor 3D maps.
- Positioning identifies the position of the user on the map. Geomagnetic field positioning [8] is accurate (about 1 meter) and users can conveniently locate indoor rooms, infrastructures and reach any given position. A magnetometer can record, match magnetic field data and provide indoor positioning result. Positioning process is based on recorded data, which can be collected by a mobile sensing device. By fusing various sensor data (magnetometer, compass, accelerator, etc.), the system determines direction, speed, and moving distance of the user. This indoor positioning technology is proposed by myself.
- Path Planning displays to the user the itinerary to follow to reach his/her destination. Positioning tells users where they are. However, to define how to go to the destination, a function of wayfinding is necessary. Such function in turn requires a map on which to

display the course from start to destination and an algorithm to plot the course to such destination (path planning). Our solution, an optimized ant colony algorithm is proposed by me and one of my team members.

- En-route Assistant aids the user where to walk or drive to reach the destination. The proposed En-route Assistant is contributed by myself. My en-route assistant includes different navigation modes that address different stakeholders: 1) common users, 2) VIPs (blind users), 3) Mobility Impaired People (MIPs/wheelchair users). Augmented Reality (AR) navigation enhances the information to common users and MIPs, which includes real-time camera frames, real-time navigation instructions and AR effects. Voice alert navigation informs VIPs obstacles in the unfamiliar environment for safe and independent trips.
- Analysis is a set of services of Business Intelligence, which measures mobility from various perspectives, such as pedestrian amount by area, time, etc. for analysis, reporting, forecasting and simulation. Pedestrian data will be a fundamental aspect to understanding and analyzing the changes indoor. Historical pedestrian flow/counts can suggest users where to go in an unfamiliar environment. An easier way of analyzing pedestrian flow is collecting data around POI. The Indoor Data is extracted from social networks by a team member, and I propose a POI recommendation algorithm to predict/forecast the trends of user flow.

A set of services support the navigation of users (Positioning, Path Planning, En-route Assistance); they run as transactions in classic systems, which respond to events and actions of end users. Other services, Analysis of Mobility Flows and Mapping, gather and update the data that are processed by transactions.

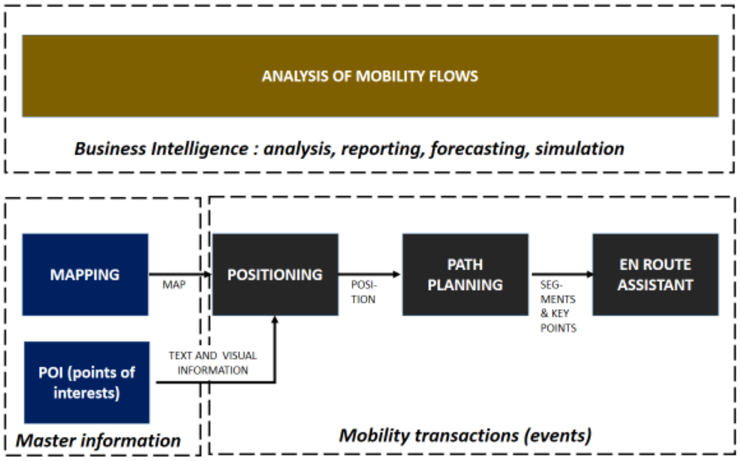
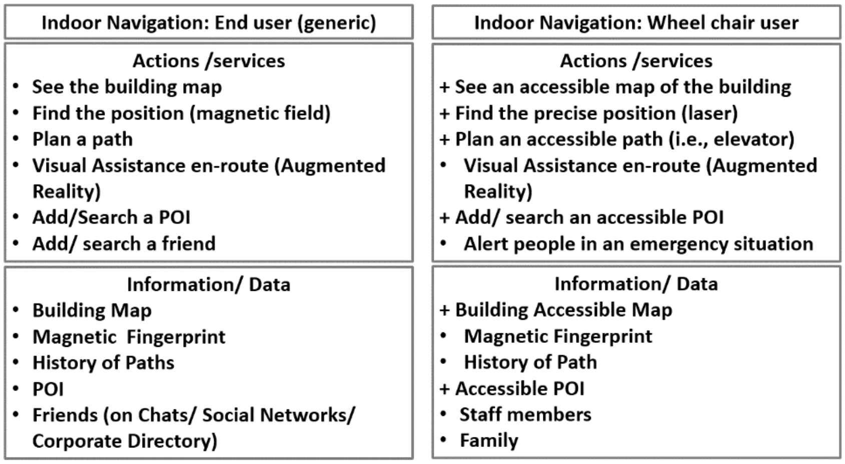

ANALYSIS OF MOBILITY FLOWS
Business Intelligence : analysis, reporting, forecasting, simulation
MAPPING
MAP
POSITIONING
POSI-TION
PATH PLANNING
SEG-MENTS & KEY POINTS
EN ROUTE ASSISTANT
POI (points of interests)
TEXT AND VISUAL INFORMATION
Master information
Mobility transactions (events)

Figure 1-1 Full phases of indoor navigation services [154] [Courtesy of Prof. Gianmario Motta]

The value propositions consider the services from a business viewpoint, i.e., the value that end users perceive from a given set of application services. Figure 1-2 shows an oversimplified value proposition of the indoor navigation for a generic user and for a wheel-chaired user. The top box includes the title of the value proposition, the intermediate box lists the services / action which the user uses, and, finally, the bottom box lists the data the services will use. The box of wheel-chaired user lists the actions and data that differ from the basic version of the generic user.



Indoor Navigation: End user (generic)	Indoor Navigation: Wheel chair user
Actions /services • See the building map • Find the position (magnetic field) • Plan a path • Visual Assistance en-route (Augmented Reality) • Add/Search a POI • Add/ search a friend	Actions /services + See an accessible map of the building + Find the precise position (laser) + Plan an accessible path (i.e., elevator) • Visual Assistance en-route (Augmented Reality) + Add/ search an accessible POI • Alert people in an emergency situation
Information/ Data • Building Map • Magnetic Fingerprint • History of Paths • POI • Friends (on Chats/ Social Networks/ Corporate Directory)	Information/ Data + Building Accessible Map • Magnetic Fingerprint • History of Path + Accessible POI • Staff members • Family

Figure 1-2 Value propositions of Indoor Navigation for a generic user and for a wheel-chaired user

1.2 Overall Architecture

Since a navigation system is an extension of a basic GIS (Geographic Information System), data in a series of information layers are linked through geographic coordinates. In Figure 1-3, I exemplify data layers for indoor navigation. Although data layers look functionally similar, their content differs profoundly: e.g., an indoor map of a building with multiple floors, on which are mapped the Points of Interest (POI) and the paths of pedestrians walking in the building and, the time series of flows. Indoor Mobility relies on both Mobile Terminals and Server that typically runs in a cloud environment. Let us consider the architecture of this indoor magnetic navigation system:

- Indoor Mapping: proprietary information of the Real Estate, which provides static geomagnetic field data for positioning and POIs. It is obtained by converting DWG into SVG files and 3D indoor game environment for navigation simulation;
- Indoor Positioning: positions the user on a floor within a building, Wi-Fi positioning for floor estimation, and informs the front-end services; the position is obtained by using the magnetometer of the smart phone and 2D and 3D indoor map, where each pixel stores the related coordinates;
- Indoor Path Planning: calculates an optimal route for indoor users based on ACO algorithm and forwards directional prompts, such as go straight, turn left or right; guide users to their destinations in an unfamiliar indoor space;
- POI management: gathers and stores information about POIs in a building, and links any kind of attachment, like speech clips, text clips, audio clips, etc. Users search and locate the POIs they are looking for. Those services not only relate to the corresponding indoor POI coordinates, but also concern the POI content. During these indoor activities, when a user is passing by a POI, the system may also pop up a notice to remind service guidelines.

Indoor mobility system can also support disabled users, it shows that the smart phones are replaced by a much more complex hardware and it supports the services that have listed in Figure 1-4. Actually, the system can interact with devices and sensors using a common protocol.

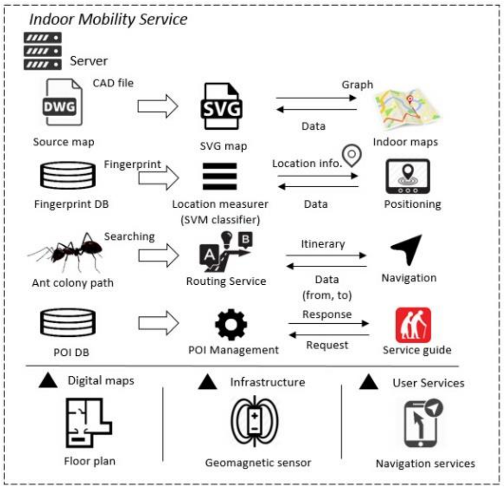


Figure 1-3 System architecture of Indoor Magnetic Navigation [154]

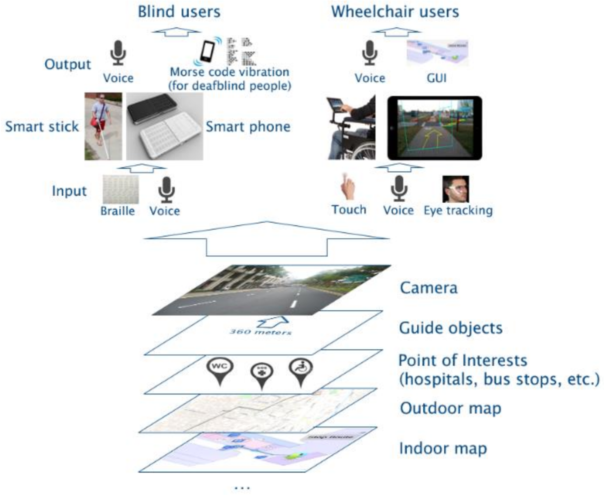


Figure 1-4 Architecture for disabled users [154] [Courtesy of Prof. Gianmario Motta]

1.3 Objectives

The system that I have illustrated meets users’ indoor services demand, improves users’ indoor experiences and reduces the stress of searching positions and paths. In particular, I have discussed a real-time pedestrian navigation in buildings with multiple floors. This indoor navigation can bring various benefits:

- Indoor navigation extends outdoor navigation to multi-floor complex buildings; as Ant Colony Optimization (ACO) path algorithm can detect the boundary of building, it is really capable of a seamless integration of indoor and outdoor in further development.
- Precise positioning within 1 meter offers accurate navigation;
- Indoor navigation can be the foundation for a service system which shall integrate:
    - Hardware and software technologies of smart phones
    - Indoor positioning and path finding models/technologies

- Magnetic fields
- Wifi communication
- Accerator, Dead reckloning etc.

- It is a smart indoor navigation that suggests path and POIs;

Of course, many important future works emerge. Finally, a logic development shall be the design and implementation of an indoor navigation system that will provide a flexible indoor mobility to any user.

Let us point out that Indoor Navigation can have some relevant commercial implications. For instance, shopping malls and libraries are promising application field. Imagine that, when a smart phone user would like to purchase famous brand fragrances or eat Thai food, it is easy to get lost in a large, complex mall. In such case, if the user sets up indoor navigation system simply, he/she will find the destination successfully and easily. In the near future, the commercial discounts and promotions can be available to promote their goods directly promote consumer shopping.

1.4 Thesis Structure

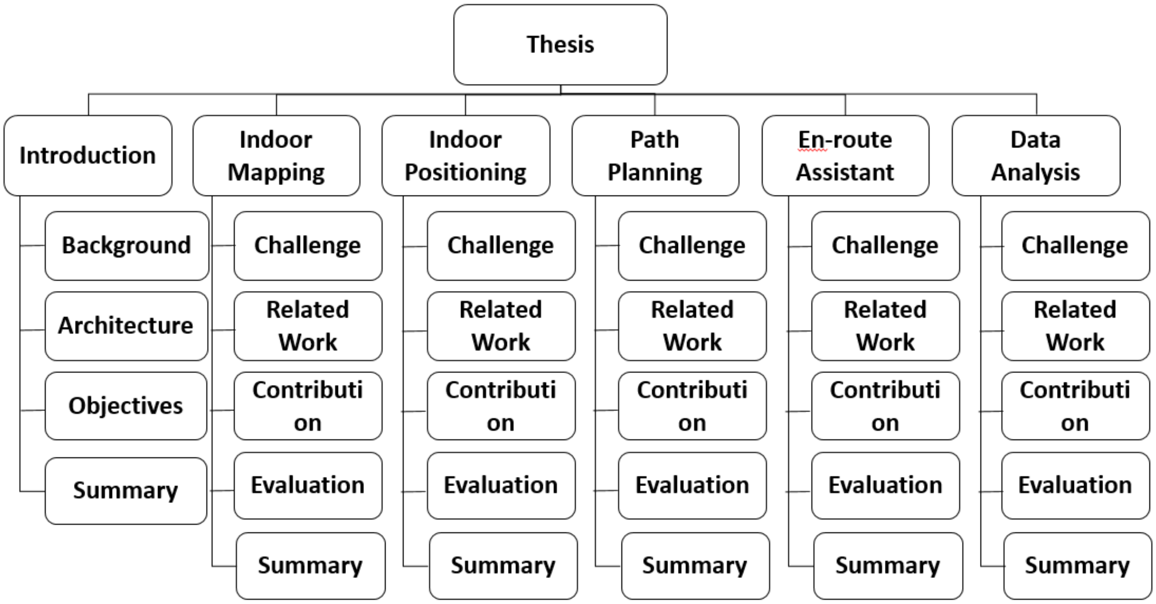


Figure 1-5 Thesis Structure

The thesis structure (Figure 1−5) is organized as follows:

- **Chapter 1 - Introduction**: I introduce the background and challenges of indoor mobility. Then, I present full phases of indoor navigation services, which overcome the challenges of indoor mobility. Then, I describe the overall architecture of Indoor Mobility system. Finally, the objectives of this integrated system and thesis structure are illustrated in the thesis.
- **Chapter 2 – Indoor Mapping**: I detail the challenge of Indoor Mapping. Then, I compare the current states of techniques and approaches in solving the key issues in

Indoor Mapping. According to the surveyed pros and cons of existing researches, I propose my Indoor Mapping approach. Finally, I evaluate my approach and it shows a good performance.

- **Chapter 3 - Indoor Positioning**: I detail the challenge of Indoor Positioning. Then, I compare the current states of techniques and approaches in solving the key issues in Indoor Positioning. According to the surveyed pros and cons of existing researches, I propose two solutions. Finally, I compare and validate my approaches and they perform well.
- **Chapter 4 - Indoor Path Planning**: I detail the challenge of Indoor Path Planning. Then, I compare the current states of techniques and approaches in solving the key issues in Indoor Path Planning. According to the surveyed pros and cons of existing researches, I propose an Indoor Path Planning algorithm. Finally, I evaluate this algorithm and it shows a good performance.
- **Chapter 5 – En-route Assistant**: I detail the challenge of En-route Assistant. Then, I compare the current states of AR techniques and VIP navigation approaches in solving the key issues in En-route Assistant. According to the surveyed pros and cons of existing researches, I propose my En-route Assistant approach. Finally, I evaluate my approach and it shows a good performance.
- **Chapter 6 – Indoor Data Recommendation**: I detail the challenge of Data Recommendation. Then, I compare the current states of techniques and approaches in solving the key issues in Data Recommendation. According to the surveyed pros and cons of existing researches, I apply my Data Recommendation algorithm in indoor scenarios. Finally, I evaluate this algorithm and it shows a good performance.
- **Chapter 7 – Conclusion and Future Works**: This chapter presents final conclusions and future research.

1.5 Acronyms

Acronyms used in this thesis are listed in Table 1-1.

Table 1-1 Acronyms

Acronyms	Definition
GPS	Global Position System
SVG	Scalable Vector Graphics
RSML	Robot Simulation-oriented Mobile Localization
CVFL	Computer Vision-oriented Fingerprint Localization
XPF	eXtended Particle Filtering
MFI	Magnetic Fingerprint Image-rization
CNN	Convolutional Neural Network
VIP	Visually Impaired People
MIP	Mobility Impaired People
POI	Point Of Interest
AR	Augmented Reality
ACO	Ant Colony Optimization

GIS	Geographic Information System
IoT	Internet of Things
AP	Access Point
RP	Reference Point
SLAM	Simultaneous Localization And Mapping
LBS	Location-Based Services
LBSN	Location-Based Social Network
OSM	OpenStreetMap
JOSM	Java OpenStreetMap Editor
CTM	Current Transformation Matrix
RFID	Radio frequency identification
VR	Virtual Reality
WLAN	Wireless Local Area Network
PDR	Pedestrian Dead Reckoning
SVM	Support Vector Machine
SVC	Support Vector Classifier
IMU	Inertial Measurement Unit
RSSI	Received Signal Strength Indication
CSI	Channel State Information
UWB	Ultra-wideband
SDELM	Semi-supervised Deep Extreme Learning Machine
DNN	Deep Neural Network
ENS	External Navigation System
INS	Inertial Navigation System
ReLU	Rectified Linear Unit
LRF	Local Receptive Fields
FC	Fully Connection
SCM	Subject Clustering Model
LDA	Latent Dirichlet Allocation
VSM	Vector Space Model
CF	Collaboration Filtering

Chapter 2 Indoor Mapping

2.1 Challenge on Indoor Mapping

Indoor mobility should address issues including building assets maintenance, security, wayfinding, etc. For example, users may not find their way to get to their destination when they encounter numerous rooms, corridors, and intersections. For the location to be meaningful for navigation or other purposes, service providers need accurate indoor maps, and rigorous spatial modeling [9] can benefit from path exploration in a complex building layout.

Indoor mapping used to be the privy of engineers, planners, consultants, contractors, and designers, however, it is no longer the only case of commercial enterprises and individuals but also of business process and applications. There are three main reasons for this. Firstly, the last two decades have seen greater use of spatial information by enterprises and the public. Secondly, Indoor Mapping has been advanced in mobile computing and internet communications, making it easier than ever to access and interact with spatial information. Thirdly, indoor modelling has been advanced geometrically and semantically, opening doors for developing user-oriented, context-aware applications. This reshaping of the public's attitude and expectations with regards to spatial information has realized new applications and spurred demand for indoor models and the tools to use them.

Indoor mapping is mainly for Data Structures and Modelling, Visualization, and it needs to be flexible enough to handle indoor data in any of numerous possible formats. However, if a large amount of indoor data is added to the databases, it should end up having undesirable consequences for existing renderers. This can be seen on existing indoor maps. Some objects from the indoor maps are rendered on different renderers, such as objects with a name or walkways.

There are also some other challenges. Indoor maps will show all the data, and the indoor data can be a mess, particularly when simultaneously looking at all the floors from a multi-story building. The most likely way to handle this is to have a well agreed upon form of the data and make an effort to accommodate this format in the renderers. Likewise, additions can be made to facilitate handling the data in indoor mapping project. Indoor maps should be not so easy for users to accidentally mess up the indoor data.

Also, 2D projection are not sufficient to provide a comprehensive and vivid navigating simulation for users, because it lacks stereo parallax and motion parallax [9]. 3D representations are much more suitable to simulate rich indoor content (e.g., furniture, textures, etc.) and building assets (e.g., fire hydrant, Mechanical and Electrical (ME) systems, etc.). 3D visual simulation [10] is widely used in large virtual environments [11] such as buildings, districts, entire cities or terrains.

3D visual representation and virtual urban conceptualization benefits cognitive mapping of architecture immensely [9]. Function and designation of spaces may, and usually do, change over time. It is necessary to realize an information management system for updating indoor maps. Additionally, in many cases, users don’t have visual access to their final destination or the path leading to it [11]. Hence, a 3D realistic-looking design enhances spatial awareness and facilitates user’s pedestrian navigation.

2.2 Related Work on Indoor Mapping

Research in support of indoor mapping and modelling has been active for over thirty years. This research has come in the form of As-Built surveys, Data structuring, Visualization techniques, Navigation models, etc. Much of this research is founded on advancements in photogrammetry, computer vision and image analysis, computer graphics, robotics, laser scanning and many others.

Topological SLAM approaches construct the connectivity (i.e., topology) of the 2D indoor environment rather than creating a geometrically accurate map. Williams and Reid [12] propose a new monocular SLAM system which uses landmarks as points of reference to correct predictions and Parsley and Julier [13] use prior information for possible location of landmarks in GraphSLAM. Milstein [14] uses prior information about the shape of a building to construct more accurate occupancy grid maps in skeletal SLAM. Georgiou et al. [15] extract the architectural drawings and floor plans as a source of information to construct the prior map.

However, navigating in 3D is more effective than a traditional 2D indoor environment. Currently, various indoor navigation solutions are introduced. Some navigate 3D space using various camera perspectives [16], some use Augmented Reality (AR) [10], while others navigate using 3D game environments [17]. The game environment increases the sense of reality and user experience in a virtual world because a game engine can create engaging interaction using dynamic objects [18]. Moreover, a game engine achieves scalable environments [18].

A 2D-to-3D conversion is usually based on video cameras [19]. Harman et al. [20] proposed a 2D to 3D conversion method that is based on the recording of multiple views by synchronized cameras. They provide high quality stereoscopic images based on 2D image sequences. A 3D-TV content generation proposed by Tam et al. [21] converts TV content frames to stereoscopic images real time. However, 3D modeling using cameras has flaws, and such stereoscopic images do not support information management, because these are pixel based.

Thus, an ideal indoor mapping system should a) support information management; b) be transformable from geometry of the building into a 3D model with vertical visual connections; and c) render and interact efficiently and effectively on a smart phone environment. The 3D model must be visualized at high resolution and manipulated in various indoor services.

SVG [22] is a two-dimensional vector graphic format that standardizes network vector graphics, which strictly follows XML (eXtensible Markup Language) syntax. It allows three graphic objects: vector graphic, raster image, and text [23] [24], and allows them to be grouped, designed, transformed, and integrated under a rendering. Additionally, a text index in SVG can be easily created to enable content-based image retrieval. This format provides a new way for large-scale geographic information accessibility and management. The integration of SVG base map and GIS data enables various Location-Based Services (LBS). The XML based SVG format also supports 2D-to-3D map conversion efficiently.

In addition, a 3D indoor model based on the game engine Unity3D [25], can provide users with rich visual effects of an indoor environment, and convenient and accurate indoor positioning and navigation services. This game engine tightly integrates virtual reality technology and an efficient 3D rendering based on materials and shaders [26]. Also, it is compatible with native file formats of various 3D modeling software [18].

2.3 Contribution on Indoor Mapping

I propose a 2D and 3D indoor mapping approach, which 1) builds an effective 2D SVG map that updates selectively, 2) converts it into a 3D virtual scene automatically at back-end, 3) includes an information management system at back-end, 4) renders and interacts efficiently and effectively on a smart phone, and 5) offers a vivid virtual indoor visualization for virtual navigation services at front-end. My contribution includes two main parts: back-end data management, and front-end data visualization. In the back-end, indoor mapping includes: a) 2D SVG indoor map creation; b) 3D SVG indoor map conversion. The front-end includes: a) modeling 3D indoor models in a 3D modeling software (e.g., Rhinoceros); b) exporting and rendering simulated 3D indoor scenes in Unity3D. The interface provides the user with immersion in a simulated indoor environment and direct manipulations for virtual navigation services.

Compared with outdoor mapping, there are some distinct characteristics of indoor mapping. First and foremost, coarse localization with GPS is available only outdoors. Thus, data-association problems of indoor positioning need to be solved. Also, many outdoor maps are available open source, while indoor maps are usually not available publicly. It should be customized for different user needs.

2.3.1 Back-end Indoor mapping management

The overall process consists of the following steps (Figure 3−1): 1) 2D SVG Indoor map creation; 2) 2D and 3D map conversion mechanism; 3) 3D indoor map rendering mechanism. The core research part of indoor map design is 2D and 3D indoor map conversion.

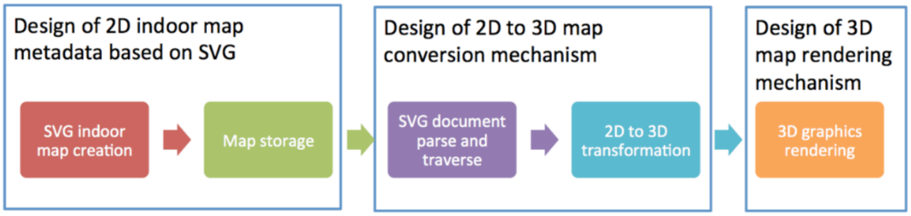

Design of 2D indoor map metadata based on SVG
SVG indoor map creation
Map storage
Design of 2D to 3D map conversion mechanism
SVG document parse and traverse
2D to 3D transformation
Design of 3D map rendering mechanism
3D graphics rendering

Figure 2-1 The process of Back-end Indoor Mapping [155]

2D SVG Indoor Map Creation: SVG [23] is used for the indoor standard map format while Open Street Map (OSM) is used for the base map with GIS information. Map information can be updated selectively in SVG. This is achieved in four steps (Figure 3−2):

Step 1: Export OSM base map of target building and GIS information. This OSM map of the building is exported by JOSM [24];

Step 2: Convert OSM into SVG. The OSM base map is converted into SVG format by Maperitive [27]. This SVG base map is generated.

Step 3: Convert CAD floor map into SVG. The SVG floor plan is converted by CAD to SVG converter [28]. This enables the creation of detailed floor plans including inner partitions.

Step 4: Combine SVG base map and SVG floor map. I combine SVG base map and floor map into a complete SVG indoor map in Inkscape [29]. Additionally, other content such as POIs are added as metadata in this SVG map.

Finally, the integrated map is uploaded and stored for data management. The metadata definition groups SVG elements in several layers.

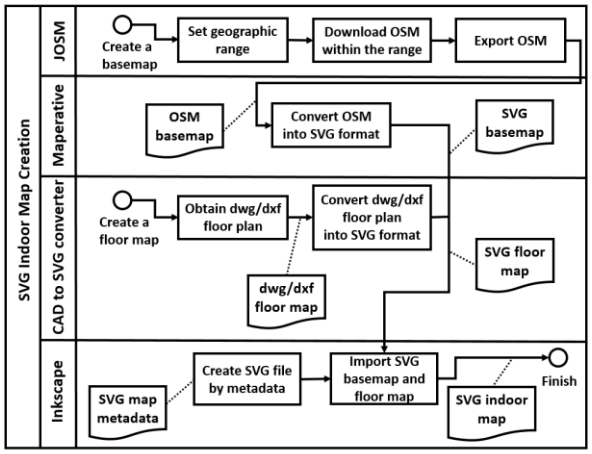


Figure 2-2 The process of SVG indoor map creation (The SVG indoor map builds up from a base map and its SVG metadata, which are obtained from OSM, .dwg/.dxf floor plan and some manual work.) [152]

2D-to-3D SVG Indoor Map Conversion: XML DOM is used to parse the layered SVG indoor map into a node-tree, so that the indoor geographic data can be accessed and further converted to 3D. A typical node-tree structure of an SVG document based on predefined metadata is presented in Figure 3-3. Both as nodes on the node-tree, SVG elements are marked in lighter rectangles, while attributes of the elements are marked in darker ones. Note that the ‘id’ attributes of all elements are omitted in the figure. Based on this node-tree structure, several definitions are made for map data transformation.

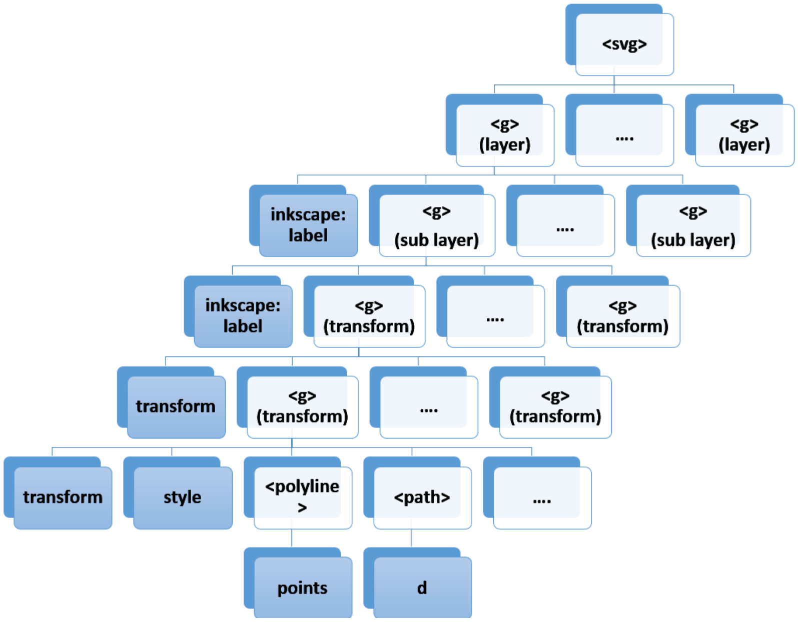


Figure 2-3 A typical SVG indoor map parse node-tree structure [152]

In the node-tree of Figure 3−3, <g> element nodes define SVG indoor map metadata. Each <g> node with a ‘inkscape: label’ attribute identifies the layer information. The roots of the other sub trees are <g> elements having a ‘transform’ attribute, whose value is a <transform-list>, which is defined as a list of transform definitions applied in the order provided. The existence of transform <g> node is due to the inevitable transformations during the conversion of OSM and DWG/DXF maps and their integration into one SVG map. Among all the available types of transform definitions, the 2D SVG indoor map includes only one transform type, namely, matrix.

Definition 1: matrix (a, b, c, d, e, f)

The transform type matrix (a, b, c, d, e, f) specifies a transformation in the form of a transformation matrix of six values and is equivalent to applying the transformation matrix [a b c d e f] (vector expression). Mathematically, a transformation matrix (a, b, c, d, e, f, 0, 0, 1) is represented as 3x3 transformation matrix.

The coordinates and lengths from a previous coordinate system are transformed into a new coordinate system by using transformation matrices, where $x_{newCoordSys}$ and $y_{newCoordSys}$ are coordinates of the new coordinate system, $x_{prevCoordSys}$ and $y_{prevCoordSys}$ are coordinates of the previous coordinate system, which in <g> sub tree of original SVG map. Among them, $x_{prevCoordSys}$ and $y_{prevCoordSys}$ are coordinates that are used to define a shape in the final SVG indoor map.

$$\begin{pmatrix} x_{prevCoorSys} \\ y_{prevCoorSys} \\ 1 \end{pmatrix} = \begin{pmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{newCoorSys} \\ y_{newCoorSys} \\ 1 \end{pmatrix}$$

Definition 2: Nested transformation

However, a transform <g> sub tree may also have one or more transform <g> nodes, which is called nested transformation. The effect of nested transformations is to post-multiply (i.e., concatenate) the subsequent transformation matrices onto previously defined transformations. In the following formula, x_{curr} and y_{curr} are untransformed coordinates that directly come from a 2D

SVG map, x_{prev} and y_{prev} are coordinates used to define a shape. Moreover, (a_1 b_1 c_1 d_1 e_1 f_1 0 0 1) is the transformation matrix located in the higher level on the node-tree.

$$\begin{pmatrix} x_{prev} \\ y_{prev} \\ 1 \end{pmatrix} = \begin{pmatrix} a_1 & c_1 & e_1 \\ b_1 & d_1 & f_1 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{curr} \\ y_{curr} \\ 1 \end{pmatrix}$$

Moreover, Current Transformation Matrix (CTM) is the accumulated transformations of the given element and its ancestors. It can be calculated by multiplied matrix. In the formula, (a_1 b_1 c_1 d_1 e_1 f_1 0 0 1) is the outermost transformation matrix on the transform of <g> sub tree.

$$CTM = \begin{pmatrix} a_1 & c_1 & e_1 \\ b_1 & d_1 & f_1 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_2 & c_2 & e_2 \\ b_2 & d_2 & f_2 \\ 0 & 0 & 1 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} a_n & c_n & e_n \\ b_n & d_n & f_n \\ 0 & 0 & 1 \end{pmatrix}$$

CTM represents the mapping of current user coordinates to viewport coordinates, where $x_{userspace}$ and $y_{userspace}$ are untransformed coordinates, and $x_{viewport}$ and $y_{viewport}$ are coordinates that define a shape in a target map.

$$\begin{pmatrix} x_{viewport} \\ y_{viewport} \\ 1 \end{pmatrix} = CTM \cdot \begin{pmatrix} x_{userspace} \\ y_{userspace} \\ 1 \end{pmatrix}$$

For a transform <g> sub tree whose height is n, the innermost transform <g> node is on the n-2^{th} level. The innermost transform <g> sub tree mainly consists of <polyline>, <path> and 'style' attribute of graphics elements.

Definition 3: Coordinate array

The coordinate array contains x and y coordinate values. In SVG, there are two kinds of coordinate array, Poly coordinate array and Path coordinate array. Polyline contains a list of points with absolute coordinates, while, path always contain Bézier curves, which are parametric polynomial curves.

A third degree polynomial is essential for matching position, slope and curvature in cubic Bézier curves. There are four control points (see Figure 3–4), P_1 and P_2 are endpoints of the curve, and meanwhile, Q_1 and Q_2 are intermediate points. The Bernshtein polynomial and its first derivatives are:

$$\begin{cases} B_3(t) = (1-t)^3 P_1 + 3t(1-t)^2 Q_1 + 3t^2(1-t) Q_2 + t^3 P_2 \\ B_3' = -3[(1-t)^2 P_1 - (1-t)(1-3t) Q_1 - t(2-3t) Q_2 - t^2 P_2] \\ B_3''(t) = 6[(1-t) P_1 - (2-3t) Q_1 + (1-3t) Q_2 + t P_2] \end{cases}$$

For example, a cubic Bézier curve is drawn from the current point (x_0, y_0) to the end point (x, y). It is not possible to transform every single point into the coordinate array. Thus, any coordinates can be calculated by changing value t from 0 to 1 in the following formula, where (x_0,y_0) is the start point, (x_1,y_1) and (x_2,y_2) are the control points, (x_3,y_3) is the end point.

$$\begin{cases} x(t) = x_0(1-t)^3 + 3x_1 t(1-t)^2 + 3x_2 t^2(1-t) + x_3 t^3 \\ y(t) = y_0(1-t)^3 + 3y_1 t(1-t)^2 + 3y_2 t^2(1-t) + y_3 t^3 \end{cases}$$

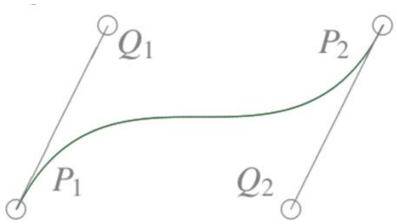


Figure 2-4 Cubic Bézier curve [152]

Definition 4: InnermostG

After parsing an SVG map into a node-tree discussed above, the system is able to access to and transform the map data. An innermost transform <g> sub tree is abstract as a class-like structure called InnermostG, which is the basic data structure of the intermediate results to be used for 3D rendering process. The logic of 2D to 3D conversion is described by the activity diagram presented in Figure 3−5.

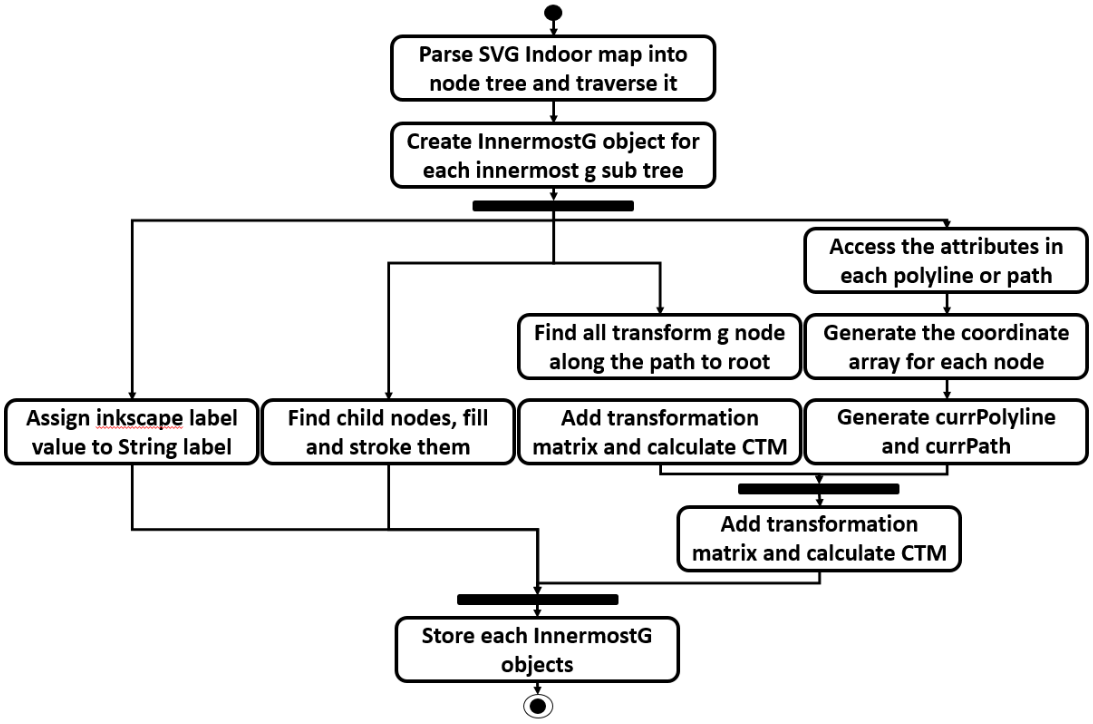

Parse SVG Indoor map into node tree and traverse it
Create InnermostG object for each innermost g sub tree
Access the attributes in each polyline or path
Find all transform g node along the path to root
Generate the coordinate array for each node
Assign inkscape label value to String label
Find child nodes, fill and stroke them
Add transformation matrix and calculate CTM
Generate currPolyline and currPath
Add transformation matrix and calculate CTM
Store each InnermostG objects

Figure 2-5 Activity diagram of 2D to 3D conversion mechanism [152]

3D Indoor Map Rendering Mechanism: Seen.js is used for two ways of rendering, namely, horizontal rendering and vertical rendering. Horizontal rendering is for floors, and vertical rendering is for walls.

2.3.2 Front-end Indoor Mapping Visualization

3D Indoor Map Creation: In the proposed front-end Indoor Mapping, the 2D floor plan with geographic information is used as a base map in 3D virtual scenes in order to support path planning. Then the modeled 3D buildings are positioned with animated features (human objects) in the appropriate places on the base map with static features (corridors, walls, etc,). A realistic 3D indoor scene is rendered with materials, lights, etc. This includes terrain layer, building layer, path layer and other attributes. The architecture is presented in Figure 3−6.

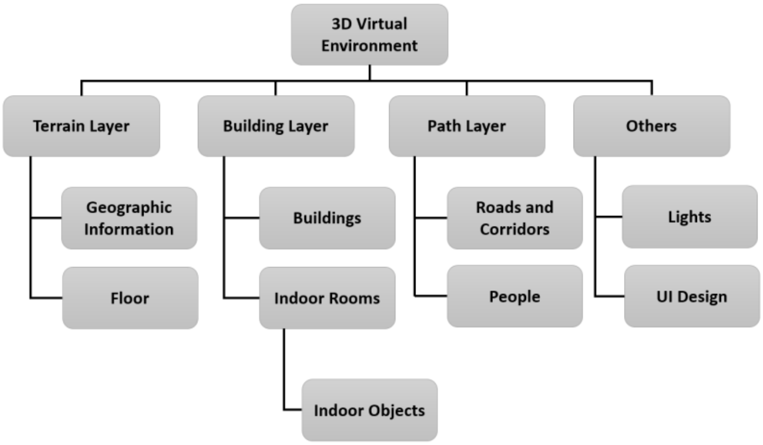


Figure 2-6 The architecture of 3D virtual environment [152]

The accessible accurate 3D models provide rich context and indoor virtual scenarios. 3D building models are created in any 3D modeling environment, such as Rhinoceros [30] or 3dsMax [31].

Moreover, the 3D virtual scene that is generated and optimized in Unity3D, achieves a good user experience with rich visual effects [19]. This 3D indoor map creation includes two parts: 3D map modeling and 3D indoor map rendering and construction. The overall process of 3D indoor environment creation is presented in Figure 3-7.

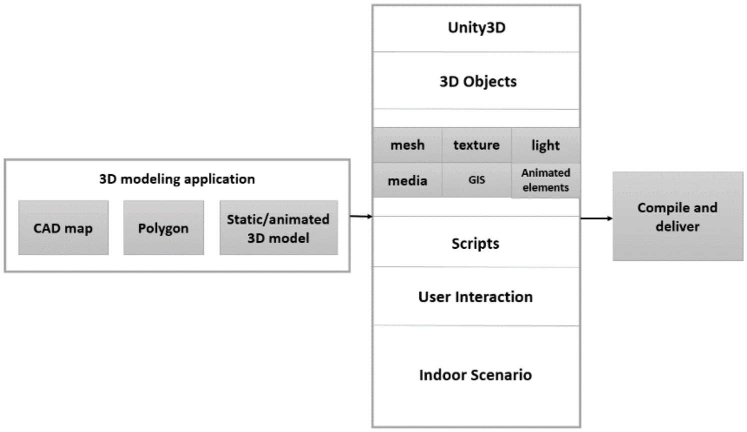


Figure 2-7 Overall process of 3D indoor environment creation [152]

3D indoor map modeling: A 3D model can be achieved in a variety of ways. The most common way of 3D model design is to capture and squeeze the outline of a drawing through an extrusion modifier [32]. I import CAD drawings from Rhinoceros application. Based on such outlines, I thicken the two-dimensional shape with an extrude modifier. This forms a three-dimensional object by closed polygon vector data extrusions. As Figure 3−8 presents, Rhinoceros application flexibly transforms the outline of a plan with the corresponding position to a 3D model.

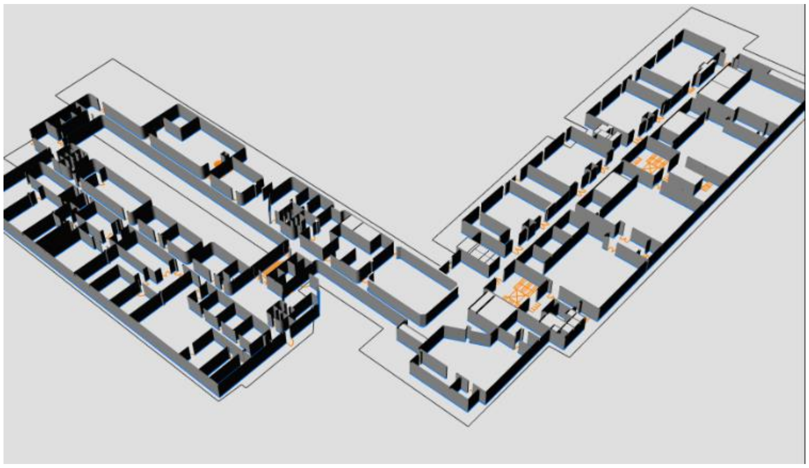


Figure 2-8 3D indoor model [152]

For some special hollow or other irregular features, such as an opening, a niche, etc., the modeling is achieved by a Boolean operation.

3D indoor scenario construction: In order to facilitate the design of a dynamic interactive large-scale virtual reality scene, I export this 3D model into Unity3D game engine. 3D indoor models with interactive functions combine the static features, such as walls, windows, paths etc., and animated features, such as human objects. Developers of the 3D model need to pay attention that the length unit used in the 3D modeling environment matches the one in Unity3D, i.e., millimeters, centimeters, feet, meters, etc., because positioning with an animated feature should be located in the correct position p among static features, as the following formula presents.

$$\mathrm{p} = [\left(\frac{model\ length}{terrain\ length}\right) * \mathrm{x}, \left(\frac{model\ width}{terrain\ width}\right) * \mathrm{y}]$$

In the formula, (x, y) is the positioning point.

At the scale of an indoor environment, the complexity of polygons and rendering becomes the main consideration for achieving a sufficient performance. For rendering, a shader of Unity3D combines the input meshes with texture and color. It packages the input content, calculates and transforms using specific parameters, obtains rendering materials, and produces the render. In order to optimize the graphics rendering and pipeline rendering [33], I choose Surface Shader with lights and shared materials. I decrease the number of vertices to improve GPU and CPU computation performance. Moreover, Unity3D batch processes the moving viewpoint (user location).

Let us recap. The rendered 3D indoor scenarios can be navigated by users, using orbit, pan, zoom, etc. The 3D virtual environment creation can be divided into the following stages:

1. Modeling of building elements, such as doors, walls;
2. 3D data acquisition from 3D modeling application;
3. Interactive operations in Unity3D with rendered 3D models;
4. 3D indoor scenario simulation on smart phones, by real-time interaction, positioning and navigating in this virtual environment.

2.4 Evaluation on Indoor Mapping

The screenshot of the 2D and 3D SVG indoor map is presented in Figure 3−9. Figure 3−10 presents the front-end of 3D indoor map visualization. After testing, users can be positioned and the positioning results can be visualized on the maps clearly, precisely and flexibly.

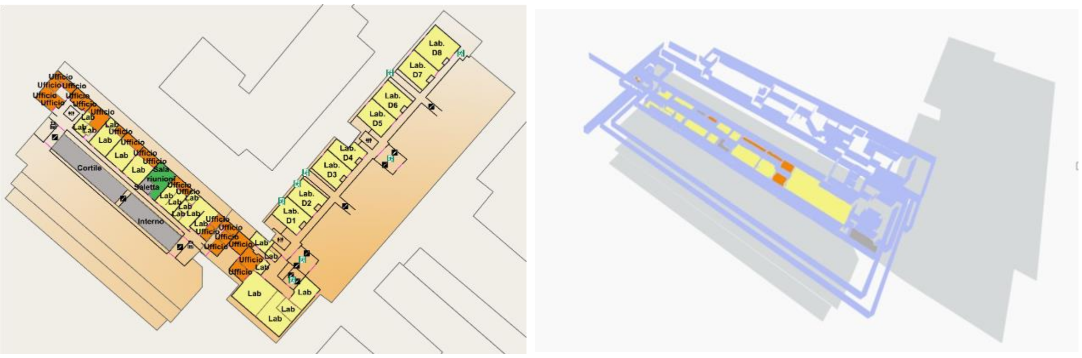


Figure 2-9 An example of 2D and 3D SVG indoor map [152]

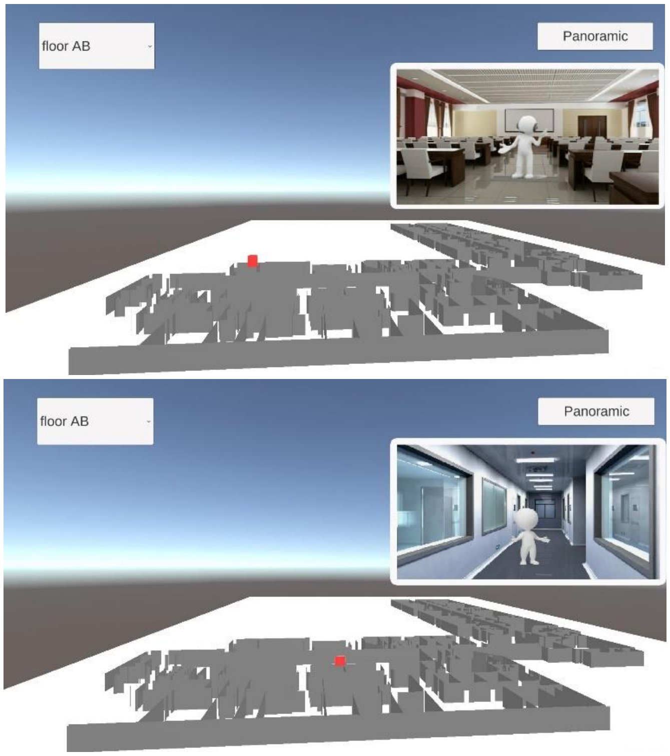


Figure 2-10 Smart phone screenshot of 3D indoor environment [152]

2.5 Summary

Indoor pedestrian navigation services need information support. I propose a specific approach, namely 2D and 3D Indoor Mapping, which provides a realistic virtual indoor environment for navigation.

Indoor mapping is a key module among indoor navigation services. Actually, 1) it builds an effective 2D SVG map and updates the map selectively, 2) it converts it into a 3D virtual scene automatically in back-end, 3) it can manage information in back-end, 4) it renders and interacts efficiently and effectively on a smart phone, and 5) it offers a vivid virtual indoor visualization on front-end.

The contributions of this chapter are front-end indoor map visualization and back-end indoor map management, both based on 2D and 3D mapping approach. The main novelties of this solution are 1) the proposed phases of indoor services, 2) SVG data management applies to indoor mapping, 3) 3D indoor visualization for indoor positioning and navigation services. Those have been validated by an indoor case and perform well.

This indoor mapping approach provides a solid foundation for indoor management and navigation services. Also, indoor map design and rapid 2D-to-3D conversion can be further extended. In my opinion, a richer indoor visualization will be further addressed, in the aspects of, e.g., enrich rendering materials, more indoor content description, etc.

Chapter 3 Indoor Positioning

3.1 Challenge on Indoor Positioning

Indoor positioning is a critical element of indoor navigation services. It can be supported by various technologies with variable effectiveness. Different researches have been done on (indoor) positioning techniques.

In most indoor contexts, the satellite coverage is not sufficient to provide an accurate result: in high buildings, multi-level highways, and tunnels, it is not possible to receive good satellite signals [34]. Therefore, it is necessary to find alternative indoor positioning. Radio frequency identification (RFID) [35] technology exchanges data in non-contact and two-way communication by radio frequency mode. Even if it is accurate, it works only at a short distance (up to tens of meters) and requires hardware. A Robot-Assisted Indoor Navigation system [36] based on Radio Frequency Identification (RFID) is developed for visually impaired people. Another positioning system [37] uses a vision-based approach, which is an infrastructure-free, scalable and cheap method combining Virtual Reality (VR) and Augmented Reality (AR). However, sometimes, users should change the position of the device to ensure the desired level of accuracy. A pedestrian navigation system [38] with WLAN based positioning is proposed in Finland. The quality of position data is rather poor. The accuracy is 1-20 meters but sometimes is interfered by other signals. The Topaz system [39] uses Bluetooth technology to locate tags indoor. It transfers data and locates positions based on signal strength. Bluetooth devices can be reused. It is a low-cost and low-power technology with an accuracy 2 m to 3 m in a delay of about 20 s.

Inertial sensors that are independent with extra devices have become a primary element in navigation. Geomagnetic fields [40] create fingerprints unique to a building. [41] They are influenced by disturbances, which are caused by steel shells, pipes, wires and electric equipment, etc. [42] that can be used as fingerprints to describe the environment. The uniqueness of fingerprint increases with the variability of local anomalies [43]. A three-axis magnetometer can sense not just navigation heading, but also variations in the magnetic field. Such variation and localized perturbations in magnetic field can be regarded as a positioning function. It has a 1m-6m accuracy [44]. If the construction of a building remains static over a period, so will the magnetic flux measured in a particular location of a building remain the same [45]. Once the magnetic anomalies have been measured and mapped to a static floor plan, no additional hardware or time investment is necessary to maintain the system. The flow of magnetic fields is variable in different locations indoor but their intensity is similar. Magnetometer provides a 3D vector that corresponds to the strength and direction of the magnetic field measured at a particular location. For each location, the

3D vector of the measured magnetic field $[mag_x, mag_y, mag_z]$can be used as a fingerprint for localization [146]. M. et al. [147] find that a rich three-dimensional magnetic field information can support location estimates. Furthermore, some researchers compare these three vectors against intensity, and prove that the three vectors are more reliable [148].

Herein, I propose two solutions to indoor magnetic field positioning: 1) Robot Simulation-oriented Mobile Localization (RSML); 2) Computer Vision-oriented Fingerprint Localization (CVFL).

- RSML focuses on a probabilistic framework that measures the relative and absolute position to simulate a robot movement. Indoor positioning requires continuous positioning results while users are walking. The Pedestrian Dead Reckoning (PDR) models a user's pose by updating an ongoing pose estimation through internal measures of velocity, acceleration, time, pedestrian odometer, etc. [46]. It predicts the position by speed measurements with gravitation tracking and gyroscope-based orientation estimation, which is a probabilistic model for pedestrian movement. The reasons of fusing magnetic measurement and PDR are: 1) PDR can reduce the computation time of pure magnetic data matching; 2) PDR itself gets the relative positioning results with accumulated errors. Also, indoor environments are always multi-floor buildings. In that case, a single tri-axis magnetometer is not enough, and it is necessary to combine it with other techniques to identify not only the horizontal but also the vertical position (in which floor you are). To achieve that result, a floor information estimation method needs to be developed.
- CVFL focuses on Magnetic Fingerprint Image-rization (MFI). Here, I name the operation that converts magnetic signals to images of waveform representation magnetic fingerprint image-rization, which is used for CNN training. A computer vision localization approach that converts fingerprints to images and analyze these fingerprinting images in order to localize users in indoor environment. K-nearest-neighbor, Deep Learning [47], and Support Vector Machine (SVM) [48], which are popular machine learning methods, currently have been applied for Wi-Fi fingerprinting based indoor localization [36]. Compared with other neural networks with similarly sized layers, CNNs [49] have much fewer connections and parameters for easier training with their theoretically best performance [50]. Since CNN utilizes layers with convolving filters that are applied to image features and it performs effectively for computer vision, the magnetic field fingerprints are able to be caught and deep learned by images of signal waveforms.

3.2 Related Work on Indoor Positioning

So far, some faculties and researchers have already done some research about magnetic field positioning technologies. Here, I survey some recent related works on journals and conferences about two research directions: Robot Simulation-oriented Mobile Localization (RSML) approach and Computer Vision-oriented Fingerprint Localization (CVFL) approach.

3.2.1 Related Work on RSML

Valter Pasku, et al. [51] propose an approximated distance measurement model, which is characterized by a reduced complexity and the cost that is suitable for 3D localization system. The testing distance results depend on the height difference between coils. Another proposed approach [52] of magnetic field positioning is based on a tri-axial magnetometer and inertial measurement unit (IMU) that measure encoded magnetic field. Their extended Kalman Filtering algorithm integrates the encoded magnetic field with Dead Reckoning. Additionally, someone presents a multisource and multivariate dataset [53], which can collect and synchronize various data from different devices and environments. Such approach can easily fuse various sensor data, e.g., WiFi signal, magnetic field data. R. Montoliu et al. [54] present a magnetic field positioning using Bag of Words paradigm [54], which allows user speed invariance. They classify the fingerprints and improve the localization accuracy from promising results. Currently, a camera-aided region-based magnetic field positioning technology [55] is proposed. It can distinguish positions with nearly same magnetic data, which performs a better positioning result. Also, a “MagSLAM” approach is proposed [56] for measurements of magnetic field strength and human odometer. Such extension of SLAM (Simultaneous Localization and Mapping) [57] algorithm obtains accurate localization results without an a priori map.

From the above surveyed magnetic field positioning technologies, we can see that they all would like to obtain more accurate results with low cost from different aspects. I propose a magnetic field positioning approach with extended particle filtering algorithm, which combines the real-time data matching and Dead Reckoning. Moreover, floor detection in multi-floor buildings by Wi-Fi signals is introduced in this section. Users can be navigated with no infrastructure and floor restrictions. I develop a Wi-Fi-aided positioning that estimates the received RP (Reference Point) signal. Actually, the Wi-Fi localization relies on signal strength RSSI (Received Signal Strength Indication), and it is a relatively simple technique without any infrastructure. Experimental results present a good floor estimation [58]. Table 4−1 lists the details of surveyed technologies along with ours.

Table 3-1 Comparison of surveyed magnetic field positioning technologies [153]

No.	Technologies	Baseline	Optimized	Advantage
1	3D localization system	Outdoor Magnetic field positioning	Distance measurement model	Reduce complexity and cost
2	Magnetic field positioning	Positioning with magnetic data and dead reckoning	Encoded magnetic data and extended Kalman Filtering	Improve accuracy
3	A special dataset	Magnetic data storage	a multisource and multivariate dataset for other data as well	Data fusing more easily
4	Magnetic field positioning	Magnetic field positioning	Bag of Words paradigm	Classify fingerprints and improve accuracy.
5	Magnetic field positioning	Magnetic field positioning	Positioning with camera sensor and detect the region	Perform better positioning results
6	“MagSLAM” approach	magnetic field strength measurement and human odometry	An extension of SLAM: “MagSLAM”	Accurate result without an a priori map.
7	My approach	Magnetic field positioning	Extended particle filtering with heading measurement, WiFi signal for floor detection.	Accurate positioning in multi-floor buildings

3.2.2 Related Work on CVFL

CNN (Convolutional Neural Network) has been successfully applied in face recognition, image classification, machine translation, unmanned aerial vehicle driving, man-machine game, etc. [49]. However, only few researches have been carried out on deep learning of (indoor) localization techniques.

Wang, Xuyu, et al. present DeepFi [59], a deep learning based indoor fingerprinting scheme using CSI information. They also propose an indoor localization method that includes a deep network of off-line training and online phase. In off-line training, it introduces a greedy learning algorithm to reduce complexity. In online phase, a probabilistic data fusion method is used for location estimation. One year later, they propose PhaseFi [60] for calibrating channel state information (CSI) with a sub-network between two consecutive layers, in order to form a Restricted Boltzmann Machine (RBM) in off-line stage.

Luo, Junhai, et al propose a fingerprinting based localization technique using deep belief network (DBN) and Ultra-wideband (UWB) signals in indoor environment [144]. They estimate the positioning by mapping the position of a Blind Node (BLN) to the corresponding signatures and creating a pattern-matching model or regression model rather than the traditional geometric localization techniques. They get an appropriate fingerprinting database and the desired accuracy.

A Wi-Fi localization based on Semi-supervised Deep Extreme Learning Machine (SDELM) is proposed by Gu, Yang, et al. [145]. SDELM trains the crowdsourcing data (unlabeled data) to improve localization performance. They deploy SDELM in different real indoor environments. After comparing with other learning methods, the proposed method not only outperforms but also reduces the calibration effort with the help of unlabeled data.

PoseNet [61] is a real-time monocular six Degree of Freedom (DoF) visual re-localization system from a camera pose based on a Bayesian CNN. It regresses the 6-DOF camera pose from a single RGB image in an end-to-end manner with no graph optimization, and achieves a localization and angle accuracy in both indoor and outdoor scenes. One year later, a camera relocation system [62] by modelling uncertainty in deep learning is further proposed to estimate metric re-localization error and detect the presence or absence of the scene in the input image.

Zhu, Yuke, et al. propose a deep reinforcement learning (DRL) framework [63] for target-driven visual navigation. It addresses two issues: 1) generalization capability to new target goals. They propose an actor-critic model for better generalization; and 2) data efficiency. System provides a simulation framework, The House Of inteRactions (AI2-THOR), which enables inexpensive and efficient collection of action and interaction on high-quality 3D scenes. Then their DRL can converge fast and generalize a real robot scenario with a small amount of fine-tuning.

A Deep Neural Network (DNN) on Wi-Fi signals for indoor and outdoor localization [64] is proposed, in which offline stage consists of a four-layer DNN structure to avoid signal fluctuation. It trains the extracted features from Wi-Fi signals and build fingerprints. In online stage, the proposed DNN-based localizer estimates a coarse position and a Hidden Markov Model (HMM) fine localizer further refines it.

It emerges that researches focus on deep learning but from very diverse perspectives, namely Wi-Fi signal or CSI, UWB signals, image sequences from camera, etc. From above, I can learn that deep learning can be applied to process various signals or other data resources. Table 4-2 lists the main features of surveyed technologies along with ours. Here, for fingerprinting localization technique, I propose a deep CNN method of magnetic field data classification, since magnetic fields, an infrastructure-free indoor positioning technique, supports more accurate positioning results. [37]

Table 3-2 Comparison of surveyed CNN-based positioning technologies.

No.	Technologies	Range	Resources	Features
1	DeepFi	Indoor	CSI information	A deep learning based indoor fingerprinting scheme
2	PhaseFi	Indoor	CSI information	Also, forms a RBM (Restricted Boltzmann Machine) in off-line stage
3	PoseNet	Indoor and outdoor	Images	A real-time monocular visual re-localization system based on a Bayesian CNN
4	An UWB navigation	Indoor	UWB signals	A fingerprinting based localization technique using deep belief network (DBN) and UWB signals
5	SDELM	Indoor and outdoor	Wi-Fi signals	A Wi-Fi localization based on Semi-supervised Deep Extreme Learning Machine (SDELM) with unlabeled data
6	PoseNet++	Indoor and outdoor	Images	A camera relocation system based on uncertainty modelling
7	A visual navigation	Indoor and outdoor	Images	A DRL (deep reinforcement learning) framework for better target generalization and data efficiency
8	DNN-based Wi-Fi	Indoor and outdoor	Wi-Fi signals	A four-layer DNN (Deep Neural Network) structure avoids signal fluctuation; HMM is for fine localizer
9	My approach	Indoor	Magnetic fields	Deep learning of fingerprints classification and accurate positioning

3.3 Contribution on Indoor Positioning

From the related work of two directions. I propose two indoor positioning solutions in the Indoor Mobility. The first RSML solution is driven by an integrated magnetic field positioning approach based on eXtended Particle Filtering (XPF) algorithm, which fuses magnetic fingerprints, Wi-Fi fingerprints, and PDR (Pedestrian Dead Reckoning). The second CVFL solution addresses fingerprint classification in Convolutional Neural Network (CNN) for magnetic field localization.

3.3.1 The RSML Solution

I propose an integrated indoor positioning approach with two essential elements, namely feature-based positioning methods on reference map, and Dead Reckoning. Such approach includes External Navigation System (ENS) and Inertial Navigation System (INS). The overall integrated positioning technology is illustrated below (Figure 4−1):

- The collected magnetic fingerprints are matched against the observed terrain. Also, the Wi-Fi signal strengths need to be collected for floor estimation. The map is modeled by natural spline interpolation, which filters the abnormal information of a building’s magnetic field.
- A person’s movement being tracked is referred to as a Pedestrian Dead Reckoning. The accurate positioning result is calculated by data matching algorithm and is overseen by

an extended particle filtering algorithm, which is a sequential Monte Carlo approach for proposing potential states and finding the observed one with matched ones.

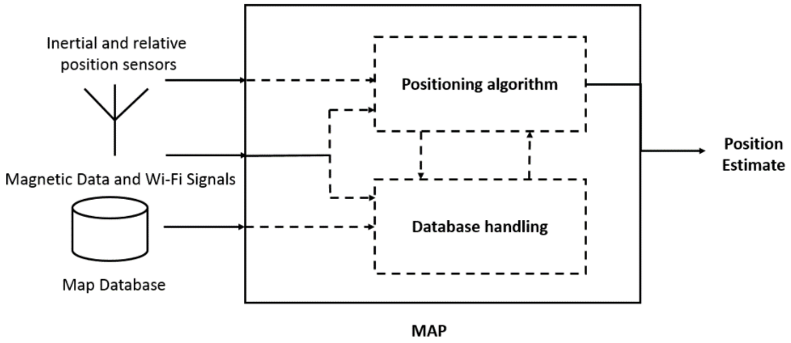


Figure 3-1 The architecture of proposed integrated indoor positioning technology [153]

Within such architecture, the positioning technology is based on an IoT approach. It is an information fusion strategy on multiple inertial sensors in this indoor navigation system. In ENS, the system estimates the floor information by Wi-Fi signals and positioning information by a single tri-axis magnetometer. In INS, pedestrian motion will be measured based on heading, user steps and walking distance. Thus, Dead Reckoning predicts the motion by accelerometer. The IoT framework of positioning technology is presented in Figure 4−2.

My integrated magnetic field positioning approach based on eXtended Particle Filtering (XPF) algorithm, which fuses magnetic fingerprints, Wi-Fi fingerprints, and PDR (Pedestrian Dead Reckoning).

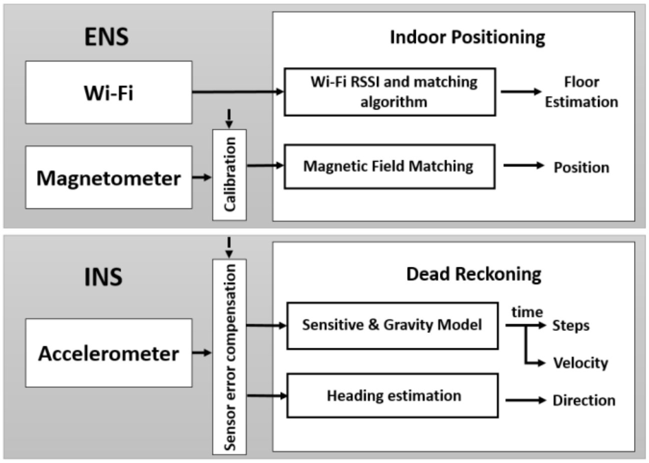


Figure 3-2 The IoT framework of Positioning Technology [153]

3.3.1.1 Magnetic Fingerprint Measurement with PDR

This approach integrates absolute and relative positioning during the whole process. First, absolute positioning technology estimates the starting location by magnetic field data. Secondly, relative positioning results of Dead Reckoning are obtained when the users are moving. Absolute positioning result corrects the relative one, while, relative positioning assists the absolute one. I obtain the initial position of PDR location by magnetic fingerprints measurement. Then, the system observes the absolute positioning result continuously. The smaller the difference between the relative one and the matched one, the more reliable.

Since particle filtering is a non-linear method approximating the posterior probability of state by weighing the random sampled particles with the principal of minimum mean square errors. It resamples the particles combining Sequential Importance Sampling (SIS) [65] and additional resampling step [66]. Such XPF is evaluated with PDR displacement data as input, and measured by positioning algorithm. It is recursive in two phases: prediction and update. In prediction, I consider both PDR displacement and magnetic data matching measurement. Then, the state is updated by my proposed non-linear motion model.

$$\begin{cases} x_{t+1} = f(x_t, u_t) + n_t \\ y_t = g(x_t) + e_t \end{cases}$$

In the formula, x is the random initial state, y is the update state, and f and g are non-linear functions. n_t and e_t are sense noises. In general, for each geomagnetic fingerprint, the more signals, the more precise the positioning result obtained. In g function model, I investigate and correct three kinds of signals, positioning, walking distance, and heading, as listed in the following table.

$$g_{t+1}^{(n)} = f(p_t^{(n)}, d_t^{(n)}, h_t^{(n)}) + \theta \cdot p_{bias}$$

Table 3-3 Investigated signals in positioning technology [153]

Object	Position	Walking Distance	Heading
User	$p^{(n)}, p_{bias}$	$d^{(n)}, d_{bias}$	$h^{(n)}, h_{bias}$

The walking distance and the heading always change in real pedestrian scenarios. The data accessed from sensors are always accompanied with deviations. This approach measures all the numerical correlated biases, as Figure 4–3 presents.

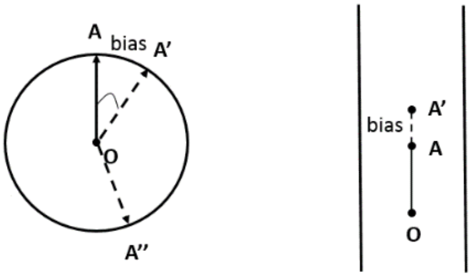


Figure 3-3 Two signals of which offsets occur [153]

In other words, in Figure 4-3, I observe the current point A with surrounding potential points (e.g., A') to obtain the highest probability point and update the pedestrian location state.

A) Magnetic Data Matching Correction

I collect magnetic data along the path by walking. Therefore, the premise of the localization is the path matching. Matching the path based on heading information first can reduce the computational time on quite large dataset.

The matching algorithm is to calculate the distance between current point and candidate point in dataset. As the following formula presented, d is the distance between the current point (x, y, z) and the collected point(x_1, y_1, z_1). Finally, the positioning result is corrected by matching biases.

$$\mathrm{d} = \sqrt{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2}$$

$$p^{(n)} = d^{(n)} + \delta \cdot p_{bias}$$

B) Walking Distance Correction

Accelerometer senses linear acceleration along one or several directions. If the changes of magnitude of the acceleration vector is higher than sensitivity, the number of walking steps increases. At the same time, I assume walking distance based on step changes is calculated by the following formula:

$$d = \begin{cases} \left(\frac{Current_{step}}{2}\right) * 3 * Step_{length} * 0.01, Current_{step}\ is\ even. \\ [\left(\frac{Current_{step}}{2}\right) * 3 + 1] * Step_{length} * 0.01, Current_{step}\ is\ odd. \end{cases}$$

Among that, I define step length as 70.

C) Heading Direction Correction

The heading of a pedestrian in motion can be measured by a software-based sensor, orientation sensor. The value of azimuth φ is estimated from orientation sensor as:

$$h = (value[0] + 360)\ \%\ 360$$

$$h_t^{(n)} = h_t^{(n)\prime} + \delta \cdot h_{bias}$$

In the formula, the value[0] is the first data in the received sensor data array.

3.3.1.2 Wi-Fi Fingerprint Measurement for Floor Estimation

Each wireless AP (Access Point) has a global unique MAC address. A smart device can detect the surrounding AP signals and obtain their MAC addresses, regardless of whether the signal is encrypted or connected. Thus, the distance between the smart device and the AP can be calculated according to Wi-Fi signal strength [67]. The Wi-Fi RSSI can be accessed by Android SDK and is calculated as follows:

$$\mathrm{RSSI} = -(10\mathrm{n}\ \log_{10} d + \mathrm{R})$$

Where, d is the distance between smart device and the reference AP. R is the RSSI value when d is 1 m; n is the signal attenuation indicator (usually 2-4) [68].

Currently, Wi-Fi based positioning technology are mainly RF triangulation based and RF fingerprint based methods. I first test the triangular based algorithm, the accurate result of itself is not high enough. In the gird of 4*4 the average accurate result within 1m achieves 43%. That is because during propagation, Wi-Fi signals interfere and diffract [69] [70]. In order to remedy the propagation drawbacks, I propose a new algorithm based on both two methods, which extends the triangular positioning approach.

A) Triangular Positioning algorithm

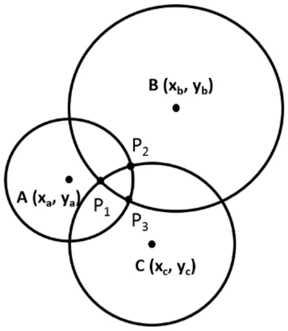


Figure 3-4 The figure of Triangular Positioning Theory [153]

In triangular positioning algorithm (Figure 4−4), the RSSI signal strength corresponding to the anchor point {A, B, C} is {R_A, R_B, R_C}. The three circles are intersected with each other, as Figure 4-4 presented. Then, the distance between smart phone to AP is {d_A, d_B, d_C}. Thus, the formula is as follows:

$$\begin{cases} (x - x_A)^2 + (y - y_A)^2 = d_A^2 \\ (x - x_B)^2 + (y - y_B)^2 = d_B^2 \\ (x - x_C)^2 + (y - y_C)^2 = d_C^2 \end{cases}$$

The intersected point P_1 of circle B and circle C is calculated as follows, the same as intersected points P_2 and P_3:

$$\mathrm{F}(x_1, y_1) = \begin{cases} (x_1 - x_B)^2 + (y_1 - y_B)^2 = d_B^2 \\ (x_1 - x_C)^2 + (y_1 - y_C)^2 = d_C^2 \\ (x_1 - x_A)^2 + (y_1 - y_A)^2 \le d_A^2 \end{cases}$$

Thus, the user location (location of smart device) (X, Y) is:

$$\mathrm{X} = \frac{x_1 + x_2 + x_3}{3}, Y = \frac{y_1 + y_2 + y_3}{3}$$

B) Wi-Fi Fingerprint Matching algorithm

Triangular positioning approach runs on three non-collinear points on the same plane. But in multi-floor buildings, the positioning result, including the vertical data is essential to estimate floor information.

I propose a fingerprint based algorithm, which extends the triangular into 3D indoor space. This algorithm using similarity ratio finds the potential points with minimum matching error between current RSSI and candidate RSSI dataset. Then, calculate the distance between current point (x, y, z) and potential points (x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4) by fingerprint data. Finally, the current point (x, y, z) is obtained, and therefore so is the vertical data z. As the following formulas present:

$$\begin{cases} \dfrac{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2}{(x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2} = (\dfrac{{S_1}^{\lambda}}{{S_2}^{\lambda}})^2 \\ \dfrac{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2}{(x_3 - x)^2 + (y_3 - y)^2 + (z_3 - z)^2} = (\dfrac{{S_1}^{\lambda}}{{S_3}^{\lambda}})^2 \\ \dfrac{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2}{(x_4 - x)^2 + (y_4 - y)^2 + (z_4 - z)^2} = (\dfrac{{S_1}^{\lambda}}{{S_4}^{\lambda}})^2 \end{cases}$$

$$\begin{cases} S_1 = \sqrt{\sum [RSSI\ (x, y, z) - RSSI(x_1, y_1, z_1)]^2} \\ S_2 = \sqrt{\sum [RSSI\ (x, y, z) - RSSI(x_2, y_2, z_2)]^2} \\ S_3 = \sqrt{\sum [RSSI\ (x, y, z) - RSSI(x_3, y_3, z_3)]^2} \end{cases}$$

This Wi-Fi fingerprint matching method extends the triangular positioning algorithm from 2D into 3D space, and it works well on small sparse dataset instead of building dense and complex one, which reduces large computational cost.

3.3.2 The CVFL Solution

I sketch deep neural networks on the shapes of three magnetic field vectors to be used in indoor localization. Thus, I present a new magnetic positioning technology in steady-state indoor environment, which 1) transforms into images the three components x, y and z of the measured magnetic field vectors; 2) trains the signal image dataset for classification in the proposed CNN architecture; and 3) matches graphic fingerprints continuously. I first label the pre-collected magnetic fingerprints by K-means, and then train the labeled ones with this CNN for classification tasks. Such trained network model is used to support further data matching for indoor localization. Figure 4-5 presents the architecture of my indoor magnetic data localization method.

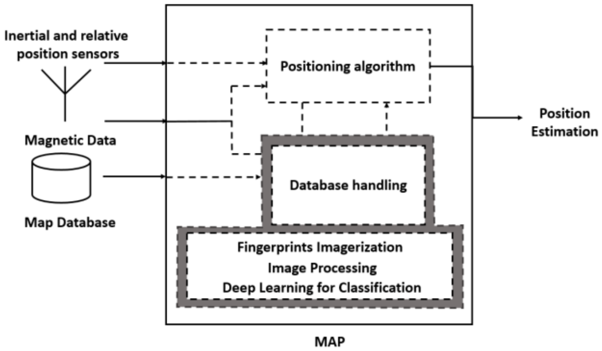


Figure 3-5 The architecture of my indoor magnetic data localization method

Typically, magnetic fingerprinting localization consists of two fundamental phases, off-line and on-line.

In off-line training phase, magnetic field measurements are taken throughout the building, and 3D maps are generated. Reference measurements (fingerprints) of the ambient magnetic field are first collected and processed to generate 3D magnetic field maps of the building; a computation technique should be chosen to deal with accuracy variance. With CNN-based technique, positioning becomes a classification problem on shapes of components' waveforms (and not of values themselves). A CNN with nine convolutional layers trains on such supervised magnetic field waveforms. The method includes MFI, image processing, image clustering, image deep learning, image matching and localization. They are all obtained from CNN, which performs well on tasks that are very different from the original ones, namely dead reckoning and fingerprint measurements by particle filtering. Such deep learning method not only reduces computational complexity, but also better performs in classification of localization fingerprints. After a fine-tuning of parameters, the neural network achieves excellent results on various classification tasks.

During the online phase (matching phase), a mobile device records real-time sensor data and matches it with the classified ones by a probabilistic method. The system estimates the user's position by searching the same category to find the most probable match and continues to match subsequent input data with the next points on this unique path. Only the matched path can ensure the target location. Figure 4-6 illustrates the whole process of this indoor localization method.

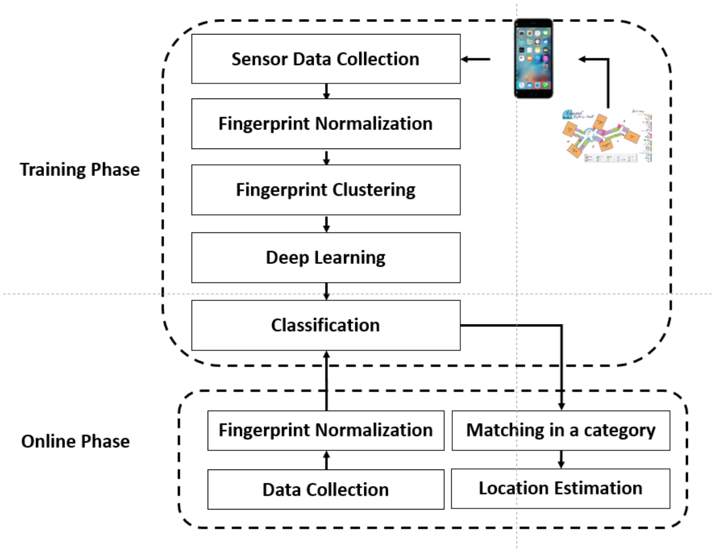


Figure 3-6 The detailed process of offline and online phases of localization method

3.3.2.1 Fingerprint Collection and Transformation

Fingerprint vectors can describe a unique indoor position. Thus, I analyze the three vectors $[mag_x, mag_y, mag_z]$, and their trends. I first collect magnetic field data from magnetometer and then perform MFI. The subsequent data normalization sits on a solid foundation of data analysis and data training. Actually, I normalize images to generate data set that are the input of CNN training phase. The steps of data collection and normalization include:

Step 1: Walk at constant speed v_b to collect the magnetic field data in a multi-floor building;

Step 2: Convert the collected magnetic signals to waveform images, by keeping same distance and same coordinate scales; an example of MFI is in Figure 4-7.

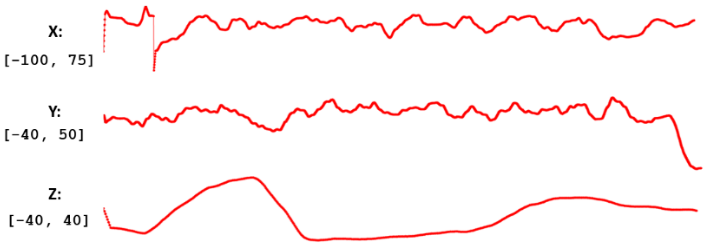


Figure 3-7 An example of MFI

Step 3: Convert the fingerprint images from RGB to grayscale using *rgb2gray* method;

Step 4: Generate crops of a given width w, and concatenate the x,y, z images in the same crop and across crops; hence crops overlap and a fingerprint image includes n crops, depending on crop width (Figure 4-8);

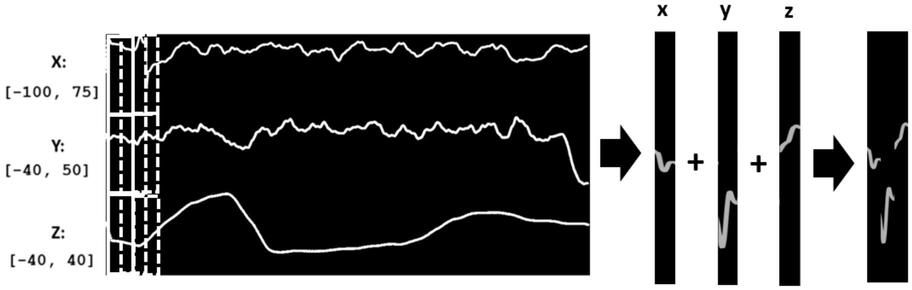


Figure 3-8 Crop and concatenate images

Step 5: Mark concatenated crops with the corresponding geo-reference information; the geo-reference data of each feature vector identify location. Then, I use the normalized magnetic field image set for the next training phase.

3.3.2.2 Fingerprint Clustering

Clustering the normalized fingerprints is the second step. MFI to waveforms are easier to be clustered. I use a distance-based approach, k-means technique, to select the best subset of features.

The image dataset is labeled by K-means algorithm that identifies several close points, each with the highest probability. Afterwards, the label results present the categories (clusters) of the ones in dataset. Thus, classification training of CNN is to be precise, because it should separate data that are surpassingly different: the classifier will perform better if the difference among data of different clusters is high. The magnetic field values are distributed within the building, and data collected are classified according to the clusters identified. The clustered labels of geo-referenced magnetic fields will be used for positioning in the online phase.

3.3.2.3 Fingerprint Classification

The layout of CNN is closer to the biological neural network [71]. In the structure of neural network, one end is the input layer or input neurons, the other end is output layer, while hidden layers are in between.

1) Artificial Neuron Model

Each input signal of a neuron has a weight, which is the connection strength between neurons. An artificial neuron is weighting sum of other neurons' input signals. Whether the connection is active or suppressed determining a positive or negative weight. Then, the activation of neurons can be obtained by the sum of the weights in each input of neurons. Assume that there are n inputs of a neuron, $x_1, x_2, \dots, x_n$,and their connection weights W are $\omega_{k1}, \omega_{k2}, \dots, \omega_{kn}$,

$$\mathrm{X} = (x_1, x_2, \dots, x_n)$$
$$\mathrm{W} = (\omega_{k1}, \omega_{k2}, \dots, \omega_{kn})^T$$

CNN learns through training data, furthermore, it can train for image processing in parallel with its special structure shared by local weights. Its layout is closer to the actual biological neural network [71]. In the structure of neural network, one end is the input layer or input neurons, the other end is output layer and hidden layers are in between.

2) Activate Function

Activation function mainly inserts non-linear factors to solve complex problems. Many activate functions, like threshold function, piecewise linear function, sigmoid function, ReLU (Rectified Linear Unit) [76], etc., are widely used in various neural networks. Compared with other non-linear activation functions, ReLU, which is activated by a simple zero-threshold matrix without requiring input normalization from saturating. It can greatly accelerate the convergence of the stochastic gradient descent. In a ReLU, a neuron has output f as a function of its input x is:

$$\mathrm{f(x)} = \max(0, x)$$

In the proposed CNN, I apply ReLU to standardize outputs. A neuron model can be presented by the following figure, and u_k represents the weighting sum of all the input signals:

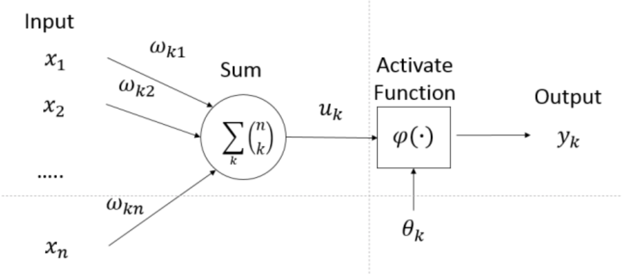


Figure 3-9 The architecture of artificial neurons

CNN learns through training data: furthermore, it can train the processed images by its special structure with shared local weights. The architecture includes convolution layer, pooling layer [73] and fully connected layer. It convolves the input image or the output of the upper layer by some filters. Then its output uses a non-linear activation function to apply pooling to produce the same features as the input of next layer. After convolution layers and pooling layers, fully connected layers are proceeded. Additionally, in recognition and classification work, the last fully connected layer is usually connected to a classifier and gives a feedback to the initial input. The parameters associated with some specific layers need to be trained; the number of parameters is also related to the filters.

Two properties of CNN, Local Receptive Fields (LRF) [72] and shared weights [73], are involved in training, which means that each filter with the same weight is used in entire image on convolution layer. That reduces the storage and has displacement invariance. A filter is a p × p matrix that is applied to a small area of the input image or particular hidden neurons moving with a defined stride length. For instance, an input with size I × I, p × p filter, and stride length s, then the size of output (feature map) is:

$$\frac{\mathrm{I} - \mathrm{P}}{\mathrm{s}} + 1$$

The Figure 4-10 presents a typical architecture of CNN. In the figure, the output of each stage is composed of feature maps. The input first goes through two convolution layers and two pooling layers, then goes through a convolution layer and a fully connected layer. Finally, the output, a feature map [74], presents some specific characteristics.

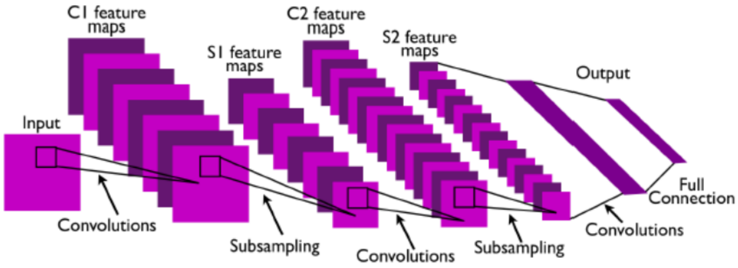


Figure 3-10 A typical example of CNN [155]

I propose a CNN-based method of magnetic field data classification, which is a 9-convolutional-layer CNN, as Figure 4-11 presents. Given the input image, my goal is to obtain one-dimensional array with a CNN, and then accurately get the classified result. The size of pre-processed fingerprinting images is 939*153, in order to adapt to the convolutional kernels, I downsample them to 200*48, which are the input of the first layer of my proposed CNN.

Table 4-4 details the structure of proposed 9-layer CNN with weights. The first layer is a convolutional layer, which filters images with a 9 x 9 kernel with input 1, output 128, and a stride of 1 pixel. Then the max-pooling layer follows. It is a sample-based discretization process. The objective is to down sample an input representation, reduce its dimensionality and allow assumptions to be made about features contained in the sub-regions. Then, the max-pooling layers follow the hidden convolutional layers with kernels 5 x 5, 5 x 5, and 21 x 2 respectively to down sample. I apply the ReLU to the output of every convolutional and pooling layer, and get a 500 output. Then the CNN connects two fully connections that connect all activations of the previous layer. Hence, these activations can be computed with a 43 matrix multiplication (number of labels) followed by a bias offset. Finally, it connects with a softmaxloss layer for predicting a single class of K mutually exclusive classes. The output of the last fully connected layer is fed to a 43-way softmax, which produces a distribution over the 43 class labels.

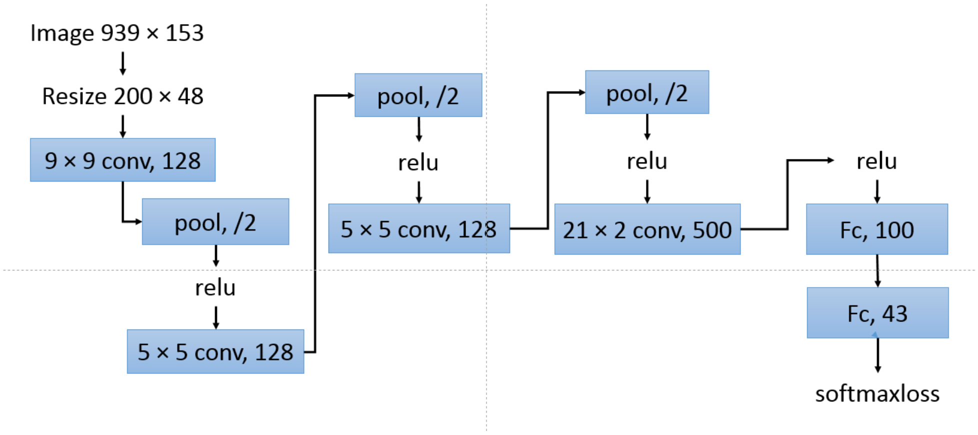


Figure 3-11 The proposed 9-layer CNN

Table 3-4 Description of proposed CNN layers

	Name	Kernel	Input	Output
Layer 1	Conv	9*9	1	128
Layer 2	MaxPooling			
Layer 3	Conv	5*5	128	128
Layer 4	MaxPooling			
Layer 5	Conv	5*5	128	128
Layer 6	MaxPooling			
Layer 7	Conv	21*2	128	500
Layer 8	FC	1*1	500	100
Layer 9	FC	1*1	100	43

Fully Connection (FC) [77] is applied after the ReLU in certain layers. The Fully Connected (FC) layer plays a "classifier" role in the entire CNN. If the convolution layer, pooling layer, and activation layer map the original data to the hidden layers in feature space, the fully connected layer acts as a function of mapping the "distributed feature representation" to the sample mark space. In practice, the fully connected layer can be realized by a convolutional operation: it is transformed by a convolution of 1x1 kernel. The fully connected layer converts a graph into a vector, which represents a classification result. Thus, I can consider that the previous convolutional layers are used for feature extraction, and the next fully connected layers are linear classifiers. Additionally, the performance of multi-fully connection is normally better than a single fully connection's.

The training process of a neural network is to minimize a loss function with respect to input data by adjusting the network parameters. The training dataset is divided into mini-batches, which are used to train this network. The system recognizes the locations using classified signals.

3.4 Evaluation on Indoor Positioning

Among the various sensors of smart phone, motion sensors are accelerometer, gravity sensor, gyroscope, rotation-vector sensor, etc., and positioning sensors are magnetometer, GPS, etc.

The testing device is the Samsung Galaxy S5 SM-G900F. The sensors that I use in this phone are listed below. Among them, the accelerator sensor counts the steps and walking distance, and geomagnetic field sensor collects and matches the geomagnetic fingerprints with orientation sensor for heading estimation, as Table 4−5 presents.

Table 3-5 The introduction of using sensors [153]

Sensor	Description	Functions
MPU6500 Acceleration Sensor	Measures the acceleration force in m/s2 that is applied to a device on all three physical axes (x, y, and z), including the force of gravity.	Motion detection.
Orientation	Measures degrees of rotation that a device makes around all three physical axes (x, y, z)	Heading estimation.
AK09911C Magnetic field Sensor	Measures the ambient geomagnetic field for all three physical axes (x, y, z) in µT (micro-Tesla).	Position measurement.

A magnetometer works well in clean magnetic environments, however, in real scenarios, it should be tested to ensure the received data is well calibrated. The main error of a magnetometer is sensor offset bias (the response between axes is not centered at the origin or the response sensitivity is different along each axis or sensor non-orthogonality). Since the heading direction and the vertical direction are considered, a plane sensor calibration can be considered as a 3D calibration for a complete rotation of the device.

The sensors are tested based on a compass swinging procedure, which rotates around three axes with a continuous series of angles. Figure 4−12 presents the testing results. In the figure, we can see that the origin located in the center of three planes, and the response sensitivity of each axis is equal. Thus, the testing sensor is well calibrated.

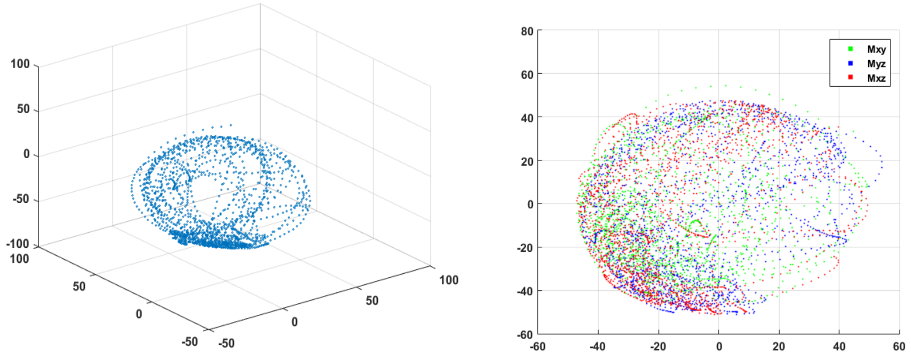


Figure 3-12 The calibrated testing result of magnetometer [153]

Then, I evaluate my two indoor positioning solutions individually based on the calibrated sensors.

3.4.1 Evaluation on RSML Solution

Variations in the ambient magnetic field can be used as features in indoor positioning and navigation. Its characteristics changes with location. The testing field is located in the engineering faculty of University of Pavia. I collect the magnetic fields of one testing area several times in different periods. As Figure 4−13 presents, I collected data six times on a corridor in three months, the magnetic field variations fluctuate steadily and the intensity is similar at the same location. Also, the changes of the geomagnetic field in a narrow area are significant. Thus, geomagnetic data of my testing field is feasible for accurate positioning.

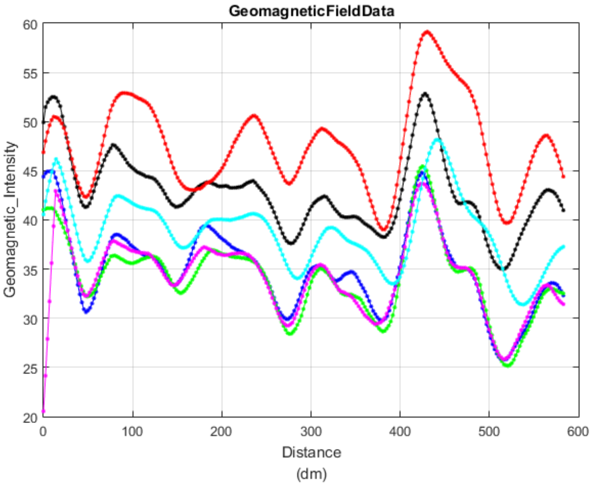
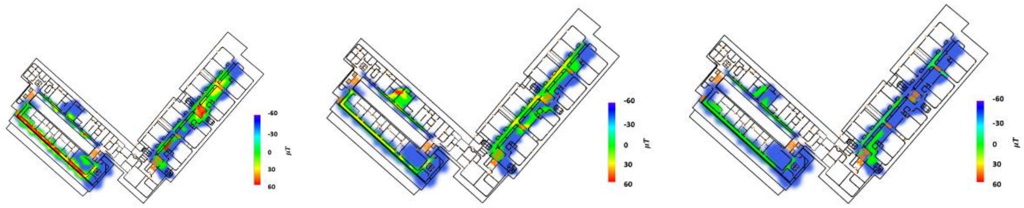


Figure 3-13 Different test of magnetic field intensity in one area [153]

The artificially generated magnetic field is not affected by most obstacles on the path [34]. It is necessary to map the collecting magnetic field data primarily for matching. The magnetometer sensor produces a three dimensional vector m $= [m_x, m_y, m_z]$ consisting of the three components, in units of μT, of the magnetic flux density in x, y, and z directions, respectively. Figure 4−14 presents the mapping data of x, y, z components.



(1) x-component (2) y-component (3) z-component

Figure 3-14 Indoor Magnetic Field Data [153]

The nearly unique magnetic field fingerprints are collected on several overlapping corridors on floor CD of our engineering building. I compared three methods: only PDR for positioning; PDR positioning with distance correction and, PDR positioning with distance and heading correction. From the testing results, as the number of steps increases, the accumulative errors of PDR sharply increase. PDR with distance correction improves the accurate result, and it (red line in Figure 4−15) shows that the accumulative errors slowly increase. However, the method of PDR with distance and heading correction is best performed after comparing.

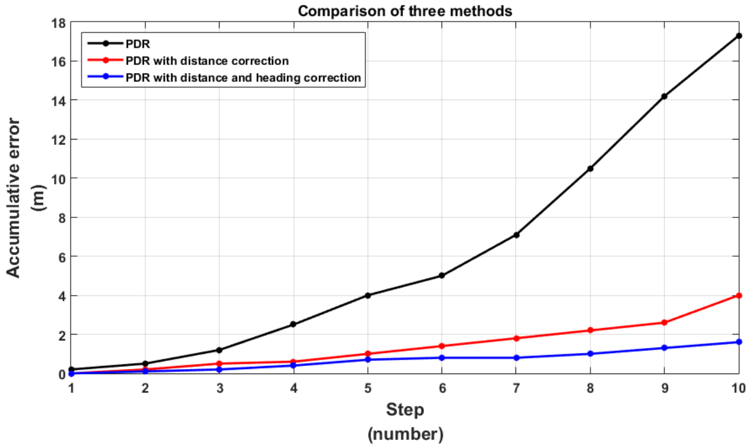


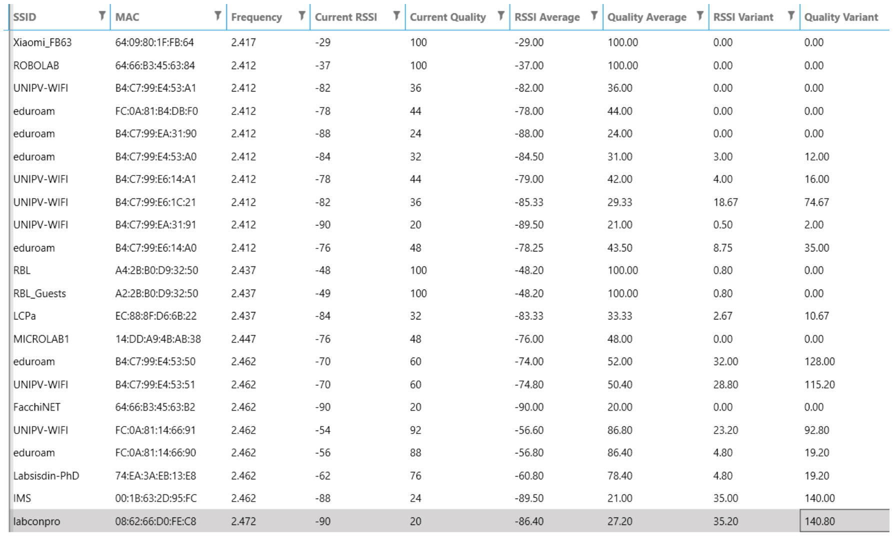
Figure 3-15 Different test of magnetic field intensity in one area [153]

I update and resample 100 particles in XPF to test the following points on different planes in the building. In floor A to D, the average localization error of random 68 testing points achieves 1.975m, 120 points achieve 2.142m, 205 point achieve 1.621 m, and 55 points achieve 1.716 m (in Table 4−6) respectively. The average accurate positioning rate of all the points achieves 1.864 m. Such testing result proves my integrated positioning approach presenting a good performance.

Table 3-6 Evaluation of accurate positioning rate [153]

	Floor A	Floor B	Floor C	Floor D
#. of testing points	68	120	205	55
Average bias (m)	1.975	2.142	1.621	1.716

I also tested the Wi-Fi floor estimation algorithm in engineering faculty of University of Pavia. Figure 4−16 presents an example of testing points' fingerprints.



SSID	MAC	Frequency	Current RSSI	Current Quality	RSSI Average	Quality Average	RSSI Variant	Quality Variant
Xiaomi_FB63	64:09:80:1F:FB:64	2.417	-29	100	-29.00	100.00	0.00	0.00
ROBOLAB	64:66:B3:45:63:84	2.412	-37	100	-37.00	100.00	0.00	0.00
UNIPV-WIFI	B4:C7:99:E4:53:A1	2.412	-82	36	-82.00	36.00	0.00	0.00
eduroam	FC:0A:81:B4:DB:F0	2.412	-78	44	-78.00	44.00	0.00	0.00
eduroam	B4:C7:99:EA:31:90	2.412	-88	24	-88.00	24.00	0.00	0.00
eduroam	B4:C7:99:E4:53:A0	2.412	-84	32	-84.50	31.00	3.00	12.00
UNIPV-WIFI	B4:C7:99:E6:14:A1	2.412	-78	44	-79.00	42.00	4.00	16.00
UNIPV-WIFI	B4:C7:99:E6:1C:21	2.412	-82	36	-85.33	29.33	18.67	74.67
UNIPV-WIFI	B4:C7:99:EA:31:91	2.412	-90	20	-89.50	21.00	0.50	2.00
eduroam	B4:C7:99:E6:14:A0	2.412	-76	48	-78.25	43.50	8.75	35.00
RBL	A4:2B:B0:D9:32:50	2.437	-48	100	-48.20	100.00	0.80	0.00
RBL_Guests	A2:2B:B0:D9:32:50	2.437	-49	100	-48.20	100.00	0.80	0.00
LCPa	EC:88:8F:D6:6B:22	2.437	-84	32	-83.33	33.33	2.67	10.67
MICROLAB1	14:DD:A9:4B:AB:38	2.447	-76	48	-76.00	48.00	0.00	0.00
eduroam	B4:C7:99:E4:53:50	2.462	-70	60	-74.00	52.00	32.00	128.00
UNIPV-WIFI	B4:C7:99:E4:53:51	2.462	-70	60	-74.80	50.40	28.80	115.20
FacchiNET	64:66:B3:45:63:B2	2.462	-90	20	-90.00	20.00	0.00	0.00
UNIPV-WIFI	FC:0A:81:14:66:91	2.462	-54	92	-56.60	86.80	23.20	92.80
eduroam	FC:0A:81:14:66:90	2.462	-56	88	-56.80	86.40	4.80	19.20
Labsisdin-PhD	74:EA:3A:EB:13:E8	2.462	-62	76	-60.80	78.40	4.80	19.20
IMS	00:1B:63:2D:95:FC	2.462	-88	24	-89.50	21.00	35.00	140.00
labconpro	08:62:66:D0:FE:C8	2.472	-90	20	-86.40	27.20	35.20	140.80

Figure 3-16 Fingerprints of testing points [153]

I collect 70 fingerprints on each floor and randomly take 91 testing points on each of three floors. Figure 4-17 demonstrates that there are 70 collecting fingerprints on the CD floor of our engineering faculty. We can see in Figure 4-18, when λ is -1.7, the accuracy rate is the highest in the testing case. The ratio of 1-meter accuracy achieves 57.51%. After testing, the floor estimation result with -1.7 (λ) achieves 93% accuracy rate, which means there are 254 points among the 273 testing points for correct floor detection.

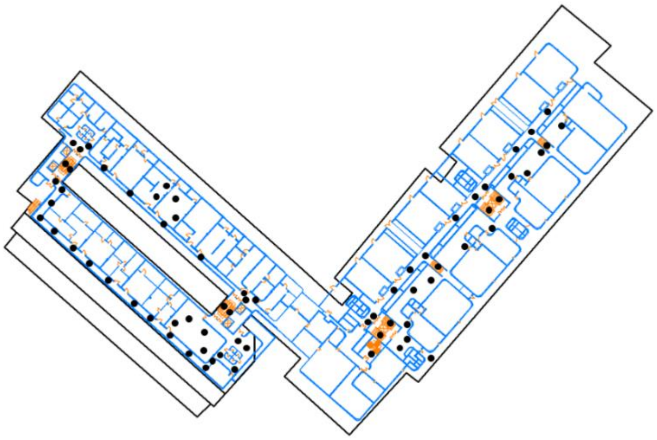


Figure 3-17 The 70 collected fingerprints on CD floor [153]

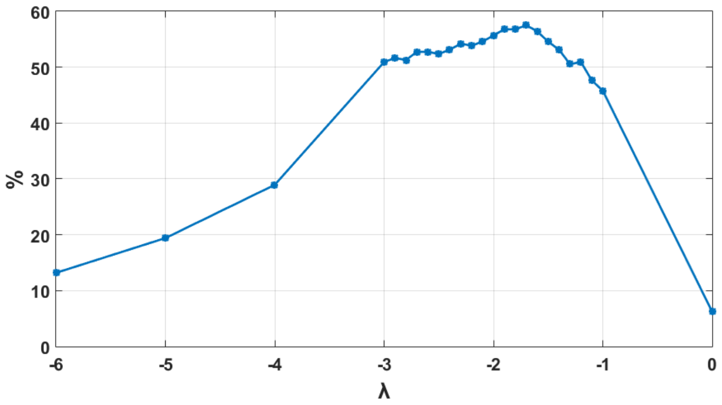


Figure 3-18 The accurate rate that within 1 meter with λ value [153]

3.4.2 Evaluation on CVFL Solution

3.4.2.1 Experiments and Evaluation on CNN

I use an equal learning rate for all layers and adjust it manually throughout training. The heuristic that I follow is tuning the learning rate when the validation error rate stops improving the current learning rate.

The magnetic field training data is collected from 8-floor Computer Engineering building in Harbin Institute Technology (HIT), China, and the fingerprinting image dataset contains 7063-labeled images under 43 categories/clusters. I partition the dataset, extract 1.000 images for model accuracy prediction, and train the rest ones. I train the proposed CNN for roughly 50 iterations through the training dataset, which takes about 11 hours on a 4-core (8-thread) Intel CPU. The fine-tuning history of CNN training process is presented in Table 4-7. Then I get a better situation with much lower error rate (Situation 7).

Table 3-7 Fine-tuning of the CNN training process

1	Learning rate	0.3		Train_Top-1	0.824
	Num epochs	60		Val_Top-1	0.827
	Batch size	150			
2	Learning rate	0.2		Train_Top-1	0.970
	Num epochs	50		Val_Top-1	0.985
	Batch size	150			
3	Learning rate	0.2		Train_Top-1	0.260
	Num epochs	10		Val_Top-1	0.295
	Batch size	150			
4	Learning rate	0.1		Train_Top-1	0.025
	Num epochs	30		Val_Top-1	0.103
	Batch size	150			
5	Learning rate	0.1		Train_Top-1	0.007
	Num epochs	60		Val_Top-1	0.080
	Batch size	150			
6	Learning rate	0.05		Train_Top-1	0.028
	Num_Epochs	60		Val_Top-1	0.073
	Batch size	150			
7	Learning rate	0.08		Train_Top-1	0.007
	Num epochs	60		Val_Top-1	0.077
	Batch size	150			

In Table 4−7, the training model 7 has a lower error rate with parameters: input image size is 28*28 with 1 channel, learning rate is 0.08, the number of epoch is 60, and the batch size is 150. Then, it is customary to report two error rates: top-1 and top-5. I achieve a winning top-1 test error of 0.077 and top-5 test error of 0.003 with an approach that averages the predictions produced from the training model on magnetic field fingerprints, as Table 4−8 presented.

Table 3-8 Estimation of the proposed CNN model

Method	Top-1	Top-5
Train	0.007	0.000
Val	0.077	0.003

I randomly field test the classification results in four floors of an 8-floor building based on my deep learning method CNN. In each floor, I randomly collect 280 fingerprinting images by uniform motion. Then I obtain its accurate result, as Table 4−9 presented. In Table 4−10, I also list the classification testing result based on SVM with a linear SVC (Support Vector Classifier), which has 80.6% accuracy on same training and testing dataset for classification prediction. After comparing two field-testing, we can see that, the classification accuracy of CNN achieves 94.6%, which is acceptable and much higher than SVM method, 78.2%.

3.4.2.2 Experiments and Evaluation on CVFL Solution

In the previous section, I discussed the off-line training phase. Here, I illustrate the online phase, which is a magnetic fingerprint matching process. Figure 4-19 presents the overall architecture of such positioning system. The online phase of proposed method includes the following steps:

Step 1: Collect real-time magnetic data on smart phone;

Step 2: Process real-time data. Image-rize magnetic fingerprint and normalize these images. To deal with user's velocity, different sizes of observed window are used to deal with signal collection under various velocity problem. Inspired by this idea, I resize the observed window to extract data from the raw magnetic field values based on the current and storing velocity. The real-time amount of extracted data is proportional to the speed ratio, as presented as follows, v_c is the current velocity, v_b is the velocity when data storage, w is the given width of image normalization.

$$\mathrm{Re}_{\mathrm{width}} = \frac{v_c}{v_b} \cdot w$$

Step 3: Classify the real-time images. System detects their proprioceptive clusters based on trained CNN;

Step 4: After detecting each cluster, match the image within its proprioceptive cluster. I use MSE (Mean Squared Error) method to match the fingerprint images;

Step 5: Obtain the most similar image in the cluster, and then continuously match the successive sensor data on this path. If such path is matched, consider the current matched image as the localization result; if not, abort current value and re-iterate.

Step 6: Retrieve positioning geo-information, and presents such information on the map in a smart phone.

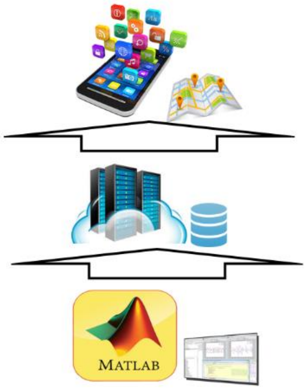


Figure 3-19 The architecture of the positioning system

Then, I tested the positioning results of 280 fingerprinting images in four floors. With the MSE algorithm, I had an average accuracy of 1-2 meters achieves in 92.9% with 1.572m average positioning result; conversely, the accuracy in SVM is 72.9% with 2.498 average positioning result, (Table 4-9, Table 4-10). Hence, CNN based positioning performs well in indoor environment.

Table 3-9 CNN based positioning testing result

Positioning Testing Result of CNN				
	Class_corr	**Class_wrong**	**Position_corr**	**Position_wrong**
1	255/280	25/280	250/280	30/280
2	266/280	14/280	260/280	20/280
3	270/280	10/280	266/280	14/280
4	268/280	12/280	262/280	18/280
# Aver	265/280	15/280	260/280	20/280
%Aver	94.6%	5.4%	92.9%	7.1%

Table 3-10 SVC based positioning testing result

Positioning Testing Result of SVM				
	Class_corr	**Class_wrong**	**Position_corr**	**Position_wrong**
1	201/280	79/280	194/280	86/280
2	217/280	63/280	197/280	83/280
3	243/280	37/280	220/280	60/280
4	213/280	67/280	205/280	75/280
# Aver	219/280	61/280	204/280	76/280
%Aver	78.2%	21.8%	72.9%	27.1%

3.5 Summary

The indoor positioning solution sits on a foundation of the complete framework, which covers the full phases of indoor navigation. In this chapter, I propose two magnetic indoor positioning solutions: Robot Simulation-oriented Mobile Localization (RSML) and Computer Vision-oriented Fingerprinting Localization (CVFL). They both maximize the correctness of positioning objective.

1) RSML, an integrated indoor positioning solution consist of ENS and INS which are based on information fusion from multiple sensors, magnetometer, accelerator, etc. It has three novelties, namely,

a. an integrated indoor positioning approach with floor estimation based on multiple sensors;
b. a novel Wi-Fi fingerprint based matching algorithm for floor estimation;
c. an eXtended Particle Filtering algorithm effectively combines both fingerprints matching and PDR.

2) CVFL, a CNN-based magnetic field fingerprints indoor positioning solution, has novelties that

a. it applies CNN to indoor positioning technology on magnetic fields and classifies the magnetic fingerprints;
b. MFI. It can continuously match fingerprints by matching fingerprinting images.

Chapter 4 Indoor Path Planning

4.1 Challenge on Indoor Path Planning

Common sense suggests that a path algorithm for a navigating user should be effective and efficient. To be effective means that algorithm should identify the most reasonable shortest way. To be efficient means that algorithm should minimize the time cost. Indoor Path Planning module may be based on various techniques as ACO (Ant Colony Optimization) [78], AC (Ant Colony), A* algorithm [79], etc. Based on these two assumptions, I here consider A* and Ant Colony Optimization algorithm.

Ant colony algorithm is suitable for dynamic path because it avoids the obstacles (walls) automatically. A* algorithm needs a large memory space to calculate the path distance, while ACO algorithm only relies on the pheromone of nearby nodes, and it plans the path dynamically based on indoor context. Thus, an optimized path algorithm that can be effective and efficient enough should be proposed in the field testing. Moreover, the segments and turning points of the optimized path can be extracted, which are the foundation/ resources for subsequent En-route Assistant module.

4.2 Related Work on Indoor Path Planning

A* algorithm [80] is used to solve the problem of the shortest path. It uses g(n) to indicate the path cost from the start point to the vertex n and h(n) to identify the estimation of cost from vertex n to the destination. h(n) is the heuristic estimate. A* balances the two when it moves from the start point to the goal. Each time through the main loop, it examines the vertex n that has the lowest f(n), It computes the cost of each vertex according to formula and finds the lowest value every time in the loop:

$$f(n) = g(n) + h(n)$$

A* algorithm is based on two heuristic estimate methods: Dijkstra algorithm [81] and Greedy Best-First-Search algorithm [82]. Dijkstra's algorithm calculates the shortest path from the starting point to nearby vertices. It checks the closest which contains candidate vertices and move vertices in it to the closet of examined vertex repeatedly until it reaches the end point. It guarantees to find the shortest path when terminating the searching processing. The Greedy Best-First-Search [82] works much faster than Dijkstra's algorithm but without any guarantee to find the shortest path. It has the same idea compared with Dijkstra that evaluates the cost of current searched vertex, except that this value is calculated by the elimination to the goal. This process is known as heuristic function. Using this function, it runs very quickly towards the destination. However, the Greedy Best-First-Search is “greedy” and tries to move towards the goal even if it is not the right path.

The main drawback of A* algorithm is its memory overhead. Since at least the entire point list must be saved, A* algorithm costs a lot of memory space.

Ant Colony Optimization (ACO) algorithm [83] mimics the behavior of "simulated ants" walking around the graph representing the path planning to solve. Pheromone trails is the way to communicate information among individuals regarding paths, and used to decide where to go. The more pheromone, the more attractive trails that would be likely to be followed. ACO algorithm does not need any memory to store the node list, but the search path is based on pheromone amounts of nearby nodes. It enables to find an optimal path through graphs based on this probabilistic technique. It cannot find the shortest path at the first try, however, Ant Colony Optimization (ACO) algorithm [83] still achieves the better path very quickly in an indoor environment. In an unfamiliar large-scale environment, like multi-floor buildings, the computational overhead of other algorithms would be very high and approximate optimal solutions cannot be obtained sometimes. I prefer ACO algorithm because of its strong optimized path planning ability. I propose an optimized ant colony algorithm in [84], which compared with and outperformed other algorithms, including, A* algorithm. Table 5−1 compares those algorithms.

Table 4-1 Summary of path algorithm research [84]

Algorithm	Description	Pros	Cons
A* with Dijkstra	Calculate the shortest path among nearby vertices.	Guarantee the shortest path	High time cost, large memory requirement
A* with Greedy Best-First-Search	No calculation but quickly find the next point.	Low time cost	Not guarantee the shortest path, large memory requirement
ACO algorithm	Wayfinding by previous pheromone information	No memory requirement, tradeoff between efficiency and effectiveness. Suitable for unfamiliar space.	Tend to the shortest path

4.3 Contribution on Indoor Path Planning

Path planning is the module that supports the navigation of the user across one or multiple floors. In one-floor navigation, it determines the user’s start and destination in order to plan the path. In the multi-floor path navigation, the system will firstly navigate user from any starting point to a specific destination point (e.g. elevators or stairs); at this point, system will switch to the new floor, and continue the path planning after certain time. The path planning navigation model is based on the Ant Colony algorithm, as I previously mentioned, is described below. The basic Ant Colony algorithm is well known and my description is a simple adaption of public available sources [85].

4.3.1 Basic Ant Colony Algorithm

Ants crawling through the obstacle on the way (Figure 5−1) suggests how to solve complex path planning problems. Assume that 30 ants are crawling from A to E, and 15 ants go each side because the two sides are equal (Figure 5−1a). When an obstacle appears at BD in the middle of the path

AE, these ants have to decide whether to turn right or left (Figure 5−1b). The choice is influenced by the intensity of the pheromone trails left by preceding ants. At the first time, the same number of ants turn right and left. During the same time, more ants going cross the shorter way BCD leave more pheromone than BHD. A higher level of pheromone on the right path gives a stronger stimulus and thus a higher probability to turn right. Finally, the optimal (shorter) path ABCDE is formed.

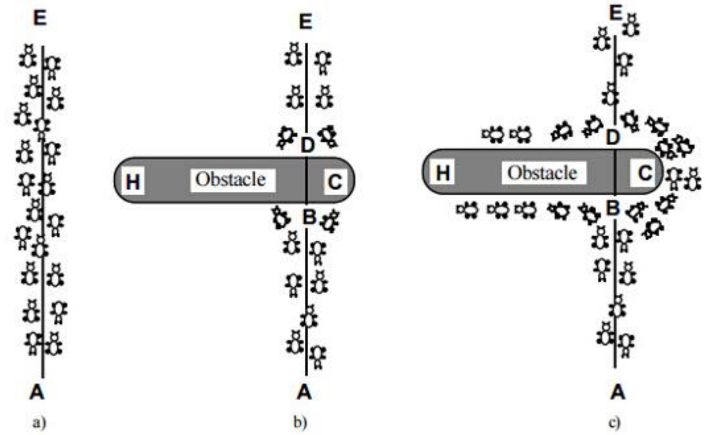


Figure 4-1 An example of real ant crossing the obstacles [85]

Consider the indoor map as a directed graph which contains nodes and arcs. The weight of arc is the length of the road. Ant algorithm will plan the path avoiding the obstacles accurately. The steps of the basic ant colony path algorithm are:

Step 1: Initialization of the moving paths

Assume 10 ants in each iteration and 20 iterations in total. Initialize the value of pheromone $\tau_{ij}(0)$ on edge (i, j) is 1. Also, initialize the value of the influence of the pheromone and the value of the influence of the distance.

Step 2: Adjustment of the quantity of pheromone trail

Let $\tau_{ij}(t)$ be the intensity of trail on edge (i, j) at time t. Each ant at time t chooses the next point, where the ant will be at time t+1, therefore, in the interval (t, t+1) every n iterations of the algorithm each ant has completed a tour. At this point the trail intensity is updated according to the following formula:

$$\tau_{ij} \leftarrow \rho\tau_{ij} + (1-\rho)\Delta\tau_{ij}$$

Where ρ is a coefficient such that (1 - ρ) represents the evaporation of trail between time t and t+n,

$$\Delta\tau = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

where $\Delta\tau_{ij}^{k}$ is the quantity per unit of length of trail substance (pheromone in real ants) laid on edge (i,j) by the k_{th} ant between time t and t+n. The value of the pheromone evaporation (1- ρ) is 0.7. When an ant is passing on a path, the increment of the pheromone ($\Delta\tau_{ij}^{k}$) is 10.

Step 3: Adjustment of each generated feasible path and generation of the moving paths of next moment for ant colony

The trail $\tau_{ij}(t)$ gives information about how many ants in the past have chosen that same edge (i,j). The visibility η_{ij} means the closer point on the map, the more desirable it is. Obviously, setting α= 0, the trail level is no longer considered, and a stochastic greedy algorithm with multiple points is obtained. After n iterations all ants have completed a tour, the values Δτij, k are updated and the shortest path found by the ants. According to the above information, the probability of next point will be calculated:

$$Prob_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{b\epsilon allowed_k(t)} \tau_{ib}^{\alpha}\eta_{ib}^{\beta}} & , \ if\ j\epsilon allowed_k(t) \\ 0 & , \ otherwise \end{cases}$$

Where η_{ij}= $1/d_{ij}$ is called a priori knowledge, which means the heuristic information of the path ij. α is the importance of residual information on the path ij. β is the importance of the heuristic information. Each ant finishes one iteration, the pheromone information will be updated. After all the ants finish the whole path, pheromone information is clear to prepare the next path planning.

4.3.2 Ant Colony Optimization Algorithm

Ant Colony algorithm plans the path avoiding the obstacles accurately without any memory to store the node list. It obtains the potential points by a stochastic greedy algorithm and updates the quantity of pheromone over time. Finally, system plans the path with highest probability of nodes and segment connections. It is much easier to find the shortest path when walking on the straight road. But how to find the shortest way in other conditions, such as corner, rugged detour, etc. In the basic ant colony system model, the ants always going around and around to search the way. But this kind of ways are not the optimal or shortest paths, which include many loops, as the Figure 5−2(a) shown. How to omit the useless loops to find the optimal path indoor becomes the main purpose of my ant colony optimization algorithm.

I use the ACO algorithm to further optimize the original one with effectivity and efficiency. Overall, I adjust the parameters, for instance, number of iterations, ant numbers, etc. to reduce the computation time, and filter the redundant cycling key points to shorten the path. The principle of the optimized ACO path algorithm is to find the straight path without going around. The initial idea is that connect the current point and the farthest point that no touching the wall as a path, which is a straight line. Then continue to wayfinding, from the end point of this path as a new starting point, looking for the next destination, in order to omit the unnecessary distance from starting point to the destination point. After testing, I found that, the path becomes shorter and better, but the curves still exist that still have room to improve the optimized ant colony algorithm. Figure 5−2(b) shows the better condition that the blue line is the optimal way.

According to such situation, in order to continue to improve the optimized ant colony algorithm for finding much shorter way. The new idea is that when I find out the farthest point that no touching the wall, continue to search next points, when I find more appropriate point that could connect the starting point, I consider this new point as the new search starting point and continue to search next point. Otherwise use the original destination point. Continue searching until the algorithm ends. Figure 5−2(c) shows the improved condition that the optimized path is much shorter and no more loops and curves. The detailed optimization is as follows.

Firstly, convert the map into a binary map, then, I set three parameters A, B, and C to calculate the distance between two nodes.

$$\begin{cases} A = \dfrac{n\_s.y - n\_e.y}{n\_s.x - n\_e.x};\ B = -1;\ C = n\_s.y - n\_s.x * A; & other \\ A = \ -1;\ B = 0; C = n\_s.x; & (n\_s.x - n\_e.x) = 0 \end{cases}$$

Among them, n_s means the starting node, n_e means the ending node, and x and y are the coordinates of the node on the map. Then, the distance dis between two nodes is:

$$\text{dis} = |\frac{A * n_i + B * n_j + C}{\sqrt{A^2 + B^2}}|$$

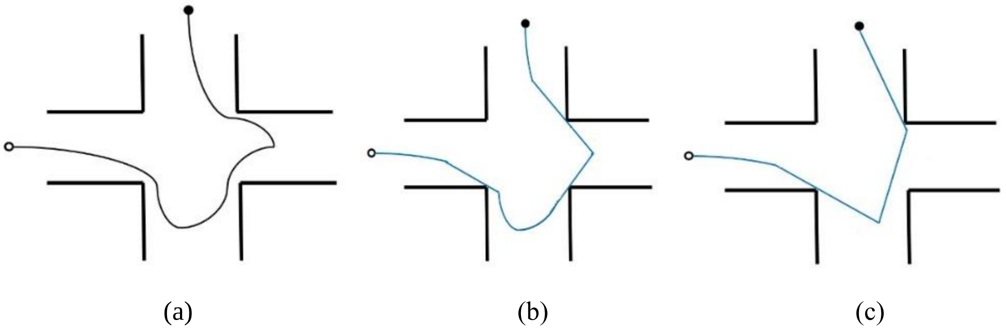
The obstacles while path planning can be detected as follows:

$$\text{is\_block} = \begin{cases} 1, & dis > 1 \\ 0, & dis \leq 1 \end{cases}$$

After detecting obstacles, system selects the optimized nodes. Then I connect the selected nodes in turn with straight segments. The straight line between the current point and the farthest point that no touching the wall with a limited distance. I define such limited distance Δnum equals with the distance of continuous 200 points [41].

$$\text{n(i)} = \begin{cases} n(i) + \Delta num, & (n(i+1) - n(i) \geq 200) \\ \quad n(i), & \qquad\qquad others \end{cases}$$

Figure 5-2 presents a real scenario of ACO path optimization process and shows the pseudo code below. Figure 5-2 (a) presents an example of AC path, Figure 5-2 (b) refers to a path connecting the current point and the farthest point that is not touching the wall, and Figure 5-2 (c) is the proposed optimized path, in which connects the next node with additional 200 points' distance. The key points of ACO path make AR indoor navigation feasible and simple. The pseudo code is shown in the following details the optimal algorithm.



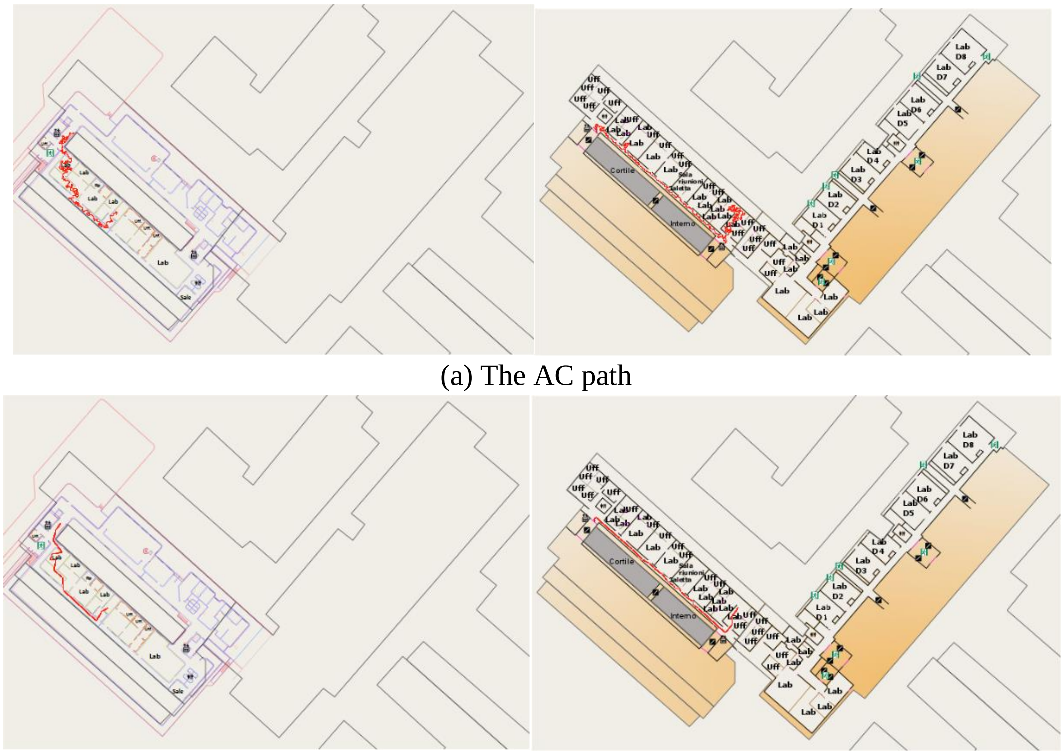
(a) (b) (c)

Figure 4-2 Optimized Path comparison

```
  Pseudo Code of ACO Path Algorithm
Begin
    Initialize the pheromone of each way
    Initialize the number of ants
    Initialize the number of iterations
    Initialize the influence of the pheromone, alpha
    Initialize the influence of the distance, beta
    Initialize the increment of the pheromone after an ant, Q
    for each iteration
        { for each ant
              chose next node on indoor map;
              judge the orientation of the ant;
              calculate the next possible node;
              store all the points in stackpath;
              update pheromone information of the points in stackpath;
          end for
         }
      end for
      if (stackpath is not null)
        {extract the points from the stackpath;
            if (line between two points intersects with the wall)
                {use the selected point as the end point;
                 }
            else{ while continue to search next point within 200 points do
                if (the next point intersects with the wall)
                      { use the original end point;
                       }
                 else {use the new selected point as the end point;
                       }
                 }
            return the best stackpath;
         }
       else { no way to destination.
             }
End
```

4.4 Evaluation on Indoor Path Planning

After testing the magnetic field technology, the error rate of magnetic indoor positioning is around 1 meter in our teaching building. With this highly precise positioning, a large number of navigation trails are also tested on indoor map of University of Pavia. The test effectively compares the basic ant colony algorithm, ant colony optimization algorithm and A* algorithm. The size of indoor map is 682×459. Here is an example path which across the floors as the Figure 5−3 shows, the trail starts from point (275, 135) of floor C to point (419, 361) of floor D. From them, Figure 5−3 (a) shows the AC trail which is non-optimized path, figure 5−3 (b) shows such optimized path. When a user arrives the stairs of floor C, the system will switch to floor D automatically and continue to navigate the user to destination.



(a) The AC path

(b) The ACO path

Figure 4-3 The path from floor C to floor D [84]

After giving a large set of testing between AC algorithm and ACO algorithm, the optimized results are apparently to see. I only extract a few testing results of navigation trails in University of Pavia, which are shown in the bellowing images and the comparison of them is shown in Table 5−2. From the testing result, I see that the distance of the trail is not the main factor of impacting the calculation time. Apparently, the distance trail based on ACO algorithm is much more reduced than AC distance. The average reduction rate is 69.9%. Even though the time cost of optimal path algorithm is a little higher than the basic one, it is acceptable. The average increase rate is only 26.2%. Also, I carry out another test about the accuracy of the algorithm. Starting from the point (53, 18) to (453, 1005) based on the map, as the trail shown in Figure 5−4, I test this trail for 10000 times and never failed to find the optimal path, which validates its high accuracy.

I implement the A* path algorithm with closest heuristic method as well. The brief conclusion is also listed in the following table. Two factors, time cost of path finding and the distance of the same trail are compared with the proposed ACO path algorithm. In general, the time overhead of A* algorithm is far more than ACO algorithm even the distance of A* algorithm is shorter than ACO. But in some extreme conditions, such like curved narrow roads in indoor environment, A* is not as accurate as ACO (A* algorithm cannot find the way in D-8 condition). In the tested path of Figure 5−4, the distance of A* algorithm is a little shorter than ACO algorithm, but the time of A*algorithm is eight times as long as ACO's, as Table 5−3 shown. As I mentioned before, A* algorithm needs enough memory to store the list of nodes are going to search. This is another reason of giving up A*.

Table 4-2 Testing results of basic ant colony model and optimization algorithm [84]

No.	Start Point (x,y)	End Point (x,y)	Time Cost of AC(ms)	Time Cost of ACO(ms)	AC Distance of trail (pixels)	ACO Distance of trail (pixels)	Distance Reduction Rate	Time increase rate	Time Cost of A* (ms)	A* Distance of trail (pixels)
A-1	(89, 98)	(268, 289)	327	429	1384	307	77.8%	31.2%	934	191
A-2	(241, 37)	(360, 170)	851	1020	1086	328	69.8%	19.9%	1066	210
B-3	(172, 78)	(269, 224)	432	552	754	234	69.0%	27.8%	981	217
B-4	(141, 77)	(258, 278)	369	492	1189	337	71.7%	33.3%	926	202
C-5	(154, 82)	(275, 135)	281	333	684	231	66.2%	18.5%	904	156
C-6	(247, 214)	(326, 195)	337	381	835	269	67.8%	13.0%	710	184
D-7	(155, 84)	(274, 266)	477	568	777	291	62.5%	19.0%	730	194
D-8	(59, 577)	(419, 361)	436	653	1800	566	68.6%	49.7%	Not Found	Not Found
Aver.	NA	NA	438	553	1064	320	69.9%	26.2%	893	194

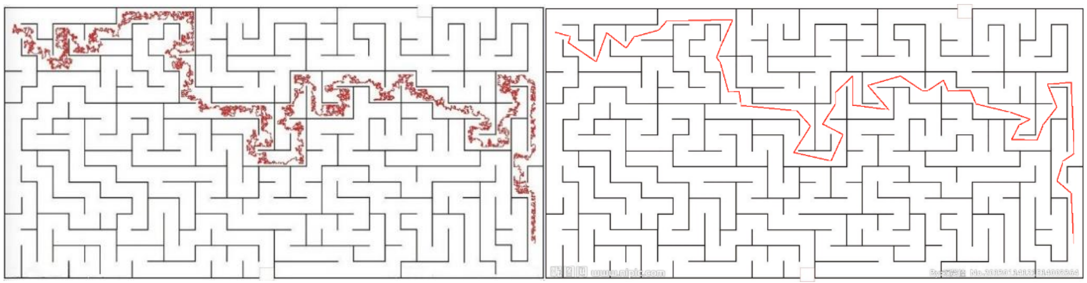


Figure 4-4 The trail of AC algorithm on typical map [84]

Table 4-3 The comparison of A* algorithm and ACO algorithm [84]

Time of A* algorithm (ms)	Distance of A* algorithm (pixels)	Time of ACO algorithm (ms)	Distance of ACO algorithm (pixels)
23207	1946	2874	2371

In order to determine the predict steps, I conduct a complete test on predicting step distance. This data of step distance analysis is related with the map that user chooses. The result differs with the chosen maps. Even the result of every calculation is different. In order to reflect the advantages of optimized Ant Colony algorithm, I make a complete testing about predict step distance on a typical map. After optimization, the path is shown in Figure 5−4. The distance of the original trails are more than 15000 pixels. When optimized the AC algorithm, the distance is reduced to only more than 2000 pixels (Table 5−4 shows). From the figure of predict step distance analysis, we can see that when the predict step distance is 200, the path is relatively shorter, as Figure 5−5 shows. The data that I choose is much more relatively reasonable. As the ACO algorithm is based on random algorithm theory, there is no optimal solution, but the feasible sub-optimal solution can be obtained. So each iteration is approaching the optimal solution.

Table 4-4 The comparison of AC distance and ACO distance [84]

No.	AC Distance of trail (pixels)	1 step of ACO	5 steps of ACO	10 steps of ACO	20 steps of ACO	30 steps of ACO	40 steps of ACO	50 steps of ACO	100 steps of ACO	150 steps of ACO	200 steps of ACO	300 steps of ACO	500 steps of ACO
1.	15897	2633	2502	2537	2447	2445	2458	2435	2412	2383	2366	2377	2377
2.	16026	2485	2346	2316	2279	2328	2341	2353	2273	2286	2245	2272	2272
3.	15488	2414	2354	2290	2349	2289	2346	2411	2387	2341	2329	2322	2322
4.	16947	2501	2304	2338	2283	2315	2302	2304	2273	2288	2282	2282	2282
5.	16770	2363	2294	2285	2239	2276	2295	2337	2382	2390	2371	2374	2374

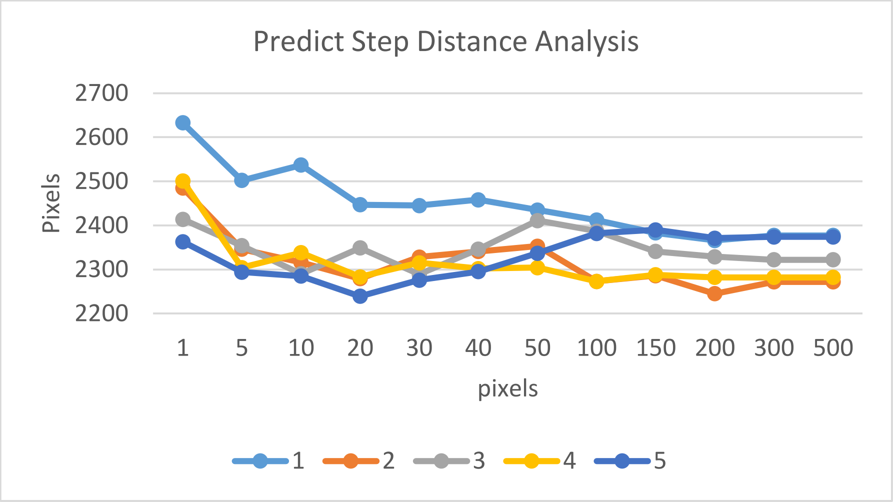


Figure 4-5 Predict step distance analysis of 5 paths [84]

4.5 Summary

The proposed indoor navigation system can work with geomagnetic fields and a novel optimization of an existing algorithm (namely Ant Colony algorithm). Also, the concept is the result of merging different competences.

The popular A* algorithm is low efficient in the testing indoor layout. Conversely, Ant Colony is efficient but it does not optimize the distance. So I work out an optimization in which I interpolate the loops of ants by straight segments and build the optimal way by a recursive process. Results are encouraging. Indeed, the resulting of ACO algorithm is not only efficient but it can be used also with other kind of positioning e.g., vision-based/ image recognition approaches.

Chapter 5 En-route Assistant

5.1 Challenge on En-route Assistant

En-route Assistant module provides the services in various devices, handed system [87], wearable system [88], etc. It can be used for an enhanced navigation. The en-route assistant relies on accurate positioning and effective path planning. In real-time navigation, it provides instructions to a convenient trip for citizens and safe trip plan for VIPs and MIPs (Mobility Impaired People).

5.1.1 Challenge on AR Navigation

In order to enhance convenient navigation, I consider Augmented Reality. It represents a real indoor scene and overlays navigation directions on the scenery of the real world. With Augmented Reality (AR) [89], the user does not need to know the route to destination, because the system displays where to go. Finally, AR builds a virtual environment, where users can "immerge".

AR is a key technology of navigation service. It includes: 1) Geo-localized AR, which is based on real-time positioning, and it uses positioning and motion sensors, and it operates under real surrounding environment [90]; and 2) AR-Tags, where AR recognizes pre-defined markers, stored on devices. Such markers may be digital media, as photos, videos, 3D models combined with audio files, which can add interactivity experience.

Actually a positioning-based navigation is based on north-oriented initial sensor [91]. Combining the positioning technology and the north-oriented relative space transformation is my novelty and contribution in AR navigation. A Geo-localized AR [90] solution, based on built-in multi-sensors, compass and accelerometer is effective for indoor navigation scenario. Specifically, the user will be accompanied to the destination by an arrow displayed on the screen of the smart phone, which tells go right, straight, or left.

5.1.2 Challenge on Navigation for VIPs

There are about 280 million people (4.25% of the world's population) around the world affected by varying degrees of visual impairment. Nearly 90% visually impaired people (VIPs) come from low-income families. Over 80% of them are over 50 years old with high morbidity [67]. In Europe, there are about 30 million visually impaired persons, accounting for 3.3% of total European population as high as 5.6% of China, which is ranking first in the world [67]. Moreover, the proportion of European people over the age of 65 will sharply increase from 14% in 2010 to 25% in 2050 [67].

Due to the increasing VIPs, the projects and researches of assistive technologies for VIPs have been increased in recent years. The important issue is how to help VIPs to move free and safely in emergency situations. An independent mobile ability should include the following four cognitive functions: walking, orientation, navigation and spatial cognition. Visual disturbances affect the mobility of VIPs, and it is difficult for most VIPs to walk fast and safely avoiding collisions, bumps and falls. Currently, in most cities, due to the lack of appropriate infrastructure, normal public transport system cannot be used by VIPs, especially in narrow residence areas. Moreover, people stay longer in indoor environments, which have more complex layouts, for instance, working in the office, studying in the school, shopping in the mall, eating in the hotel. Thus, helping VIPs to move free and safely in such indoor environment, and improve their urban lives becomes an important topic in our society.

The VIP indoor mobility issue has to be addressed at both physical and information levels. At physical level, non-barrier indoor public services are provided, such as smart facilities which include sidewalks or ramp ways for wheelchairs, smart elevators, telephones, toilets, handrails and wheelchair seats inside the buildings. At the information level, the most popular navigation systems (Amap, Baidu Maps, Google Maps Navigation, etc.) do not support VIPs. Many navigation applications for VIPs integrate POIs (Points of Interest) and accessibility data with geographic maps, without considering path security. Moreover, the voice alert is for enhancing display effects, rather than improving functions.

For safe navigation, I will consider the functions of auditory sense and accessibility [92] to fill the gap in VIP mobility support, which refers to the opportunity for disabled people to reach a given building under safety conditions. A VIP deals with an automated support is needed to receive, process, plan, guide and monitor an obstacle-free itinerary. Thus, dynamic obstacles, such as, the coming pedestrian should be detected for avoiding collision, etc.

5.2 Related Work on En-route Assistant

I here discuss works on the technology for En-route Assistance module. Assistive technologies have played an essential role in the facility, convenience, safety, and independence of people with disabilities. Here, I focus on the technologies of supporting convenient and safe ways for users, especially visually impaired users.

5.2.1 Related Work on AR Navigation

Actually, Augmented Reality (AR) navigation leads to a convenient trip for users. AR is widely used both indoor and outdoor. Some solutions address the development of wearable systems [93], some address the mobile phones. Mobile AR solutions use various outdoor and indoor positioning techniques, as GPS, computer vision, WiFi, mobile phone sensing. I summarize navigation solutions in Table 6‑1.

MARS (Mobile Augmented Reality Systems) [94], one of the first mobile AR systems, allows users to access indoor and outdoor spatial information while walking. For outdoor navigation, GPS relies on satellite connection [95]; however, its accuracy is low, ranging from 3m to 10m. For indoor navigation, MARS relies on Radio frequency identification (RFID) [96]. RFID is precise, but requires infrastructure. Hence, MARS does not fit the requirement of a navigation system which can run with a sufficient precision without any infrastructure.

Computer vision [97] [98] can identify the position by processing images; it does not require infrastructures, and it is scalable. However, image matching is relatively time consuming. In order to reach the desired level of accuracy in matching images, the user should sometimes change the position of the device. Let us comment some solutions based on computer vision. JongBae Kim and HeeSung Jun [99] propose an indoor navigation system that recognizes a location from image sequences. Tomasz Oleksy [100] uses historic images to restore the original view to users who are walking in a historic site. Walther-Franks and Malaka [101] give a photograph-based navigation that is providing a predefined route. A computer vision-based AR control of the Unmanned Aerial Vehicles (UAV) is also proposed [102]. Analogue to Computer Vision systems, HearThere [103] is a vision for indoor and outdoor auditory augmented reality (AAR). It uses a head tracking and bone conduction headphones, and it gives a sensory experience.

Hamid M. Ali and Zahraa T. Noori [104] propose an AR indoor navigation based on WiFi. The Reference Points (RPs) are previously stored in database, which records also the paths between each RP and its neighbor RPs. However, the quality of WiFi positioning [105] is rather low and the precision ranges from 1 to 20 meters depending on signal strength.

Finally, the smart phone itself can be used for both positioning and en-route assistance. Few years ago, Schall [106] presented a north-centered orientation tracking on smart phones [106] for outdoor AR navigation, where a vision-based tracker provides a stable orientation, and fusion of the sensor with magnetometers and accelerometers. Even if this solution is independent from infrastructure and provides a sufficient precision, it only displays the values but does not guide the user by a computer-human interface.

This solution enhances that concept from various view points, 1) it implements it on an indoor map of a multi-floor building; 2) it adds an interface namely, a real-time arrow that guides the user to destination; 3) It relies on geomagnetic positioning [8], which is based on data collected by geomagnetic sensor. Such positioning creates a fingerprint unique to a building with [107] a 2m accuracy [8].

Table 5-1 Summary of AR navigation research [154]

AR Navigation System	Solutions	Positioning Techniques	Positioning features
MARS	Indoor and outdoor positioning	GPS and RFID	GPS: not for indoor RFID: depend on device, Fast read/ write, Low integrate capacity
Magnified AR 3D display	Outdoor navigation	Computer vision-based AR	Scalable; Cheap; No device required; High time complexity.
North-centered Orientation tracking on Mobile Phones	North-centered Orientation tracking	GPS and other sensors	GPS: not for indoor; North-centered Orientation tracking: efficient, fast and ideally suited to handheld AR applications.
Indoor Way Finder Navigation System	Based on stored reference points (RPs)	WIFI	Acceptable; Cheap; Accurate range:30-40m; Not stable signal;
A vision-based location positioning system	Indoor Navigation	Computer vision-based (image sequence matching)	Scalable; Cheap; No device required. High precise algorithm needed; High time complexity.
HearThere	Indoor auditory AR	Fine-grained (20-cm) Ultra-Wide Band (UWB) radio tracking	High precise and depend on infrastructure.

5.2.2 Related Work on Navigation for VIPs

The current assistive tools, like white canes [108], dog guides [109], and smartphones are widely used by visually impaired people to aid in navigation [109]. Some research efforts create a variety of guide robots for this purpose.

The most ubiquitous and universally identifiable navigation tool for VIPs is the white cane, which is used for obstacles avoidance [110]. Another tool, dog guide, assists VIPs a safe, and smooth guidance in navigation. Talking signs also are easy for VIP to guide to the destinations. However, these cannot present an end-to-end navigation with a shortest, convenient and independent path in unfamiliar indoor environments. The feature-packed mobile phones provides the new interaction paradigms between human beings and indoor environment. Smart phones can also give users physical interface for a safe and end-to end navigation. The contained accessible functions with multi-sensors, for instance, text-to-speech, gesture recognition, and localization by sensors, etc., which can enhance VIPs' daily navigation life.

Serrão, M., et al. [111] propose a GIS navigation for blind persons in buildings. They detect visual landmarks, such as doors, stairs, signs, and fire extinguishers, etc. for tracing and validating a route for user's navigation by camera without a speech-based interface. BlindNavi [112] provides a smart voice feedback: short vibration means keep going straight; long vibration means turn right or left; continuous vibration means not on the right path. Each notification comes with a corresponding voice message. Moreover, its touch-based interface presents a dynamic and vivid navigation mode to users. Also, a system, proposed by Apostolopoulos, Ilias, et al. [113], real-time calculates the user's location and provides audio instructions to VIPs. A visual semantic interface proposed by Joseph, Samleo L., et al. [114] triggers the head-mounted smart glass which is comprised of camera and inertial measurement unit (IMU) for safe blind navigation. GoBraille [115] is supported by Braille device and accessible information of public transportation, such as, bus stops, etc. for blind and deaf-blind people on navigation. Listen2dRoom [116] is an ultrasonic sensor-based indoor wearable electronic travel aid with speech and non-speech sounds for auditory displays (e.g., obstacle: non-speech sounds, direction: speech sounds) to expect an obstacle to avoid. A Cell Phone-based Wayfinding for the Visually Impaired proposed by Coughlan, et al. [108] that works with camera to read the barcode. If system detects a barcode, it will give a speech sign for VIPs as a real-time wayfinding aid. Table 6−2 presents a summary of VIP navigation related work.

Table 5-2 Summary of VIP navigation research

VIP Navigation System	Solutions	Features
A GIS navigation for blind persons	Detect visual landmarks	Detect visual landmarks for tracing and validating a route for user's navigation by camera without a speech-based interface.
BlindNavi	A smart voice feedback	Each notification comes with a corresponding voice message.
System proposed by Apostolopoulos, Ilias, et al.	Audio instructions	Real-time calculates the user's location and provides audio instructions to VIPs.
System proposed by Joseph, Samleo L., et al.	A visual semantic interface.	It triggers the head-mounted smart glass with camera for safe blind navigation.
GoBraille	Braille device	A braille device and accessible information of public transportation for blind navigation.
Listen2dRoom	A wearable electronic travel aid	An ultrasonic sensor-based indoor wearable electronic travel aid with speech and non-speech sounds for auditory displays.
System proposed by Coughlan, et al.	A Cell Phone-based Wayfinding	It works with camera to read the barcode and give a speech sign for VIPs.

From above, we can see that researchers try to find the solution for VIPs' navigation from other senses, such as auditory sense, touch sense. I propose an en-route assist navigating for VIPs in unfamiliar indoor environments safely, which can avoid dynamic obstacles, incoming pedestrians, based on a speech and vibration alert.

Pedestrian detection is one subject of object detection. Support Vector Machine (SVM) [117] is widely used for object classification and recognition, which needs high CPU and memory capacity. However, Cascading Classifier [118] is another solution without requiring a high capacity of CPU and memory load. SVMs are supervised learning models with associated learning algorithms to analyze data used for classification and regression. Moreover, non-Linear SVM is for many overlap categories, which can be used for pedestrian detection. Cascading is a particular case of ensemble training by several Classifiers concatenation, using all collected information from the output of a given classifier for the next classifier in the cascade. Pedestrian cascading training is a kind of feature based cascading for positive and negative pedestrian detection. I choose boosted Haar-based feature pedestrian cascading on mobile phones because of a memory and CPU runtime consideration.

5.3 Contribution on En-route Assistant

5.3.1 Knowledge Base of Sensing Mobility

My AR navigation solution is driven by an IoT approach, which consists of mobile compass and magnetometer. The arrow displayed on interface is based on a north-oriented relative space system that calculates the angle between the sensor direction and the north. Here I illustrate the basic knowledge of sensor methodology.

Definition 1: Heading and Heading-Relative Bearing

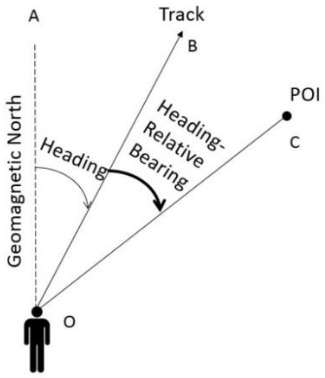


Figure 5-1 Depiction of navigational terms [154]

Heading OB is the course/direction that the tracker is following. The angle AB is the value of heading, which is the angle between the direction that the tracker faces and the north.

During navigation, absolute bearing refers to the angle between the magnetic north and an object or a POI. [7] Here I refer absolute bearing depends on a magnetic north bearing.

Heading-relative bearing is the bearing of a landmark/a POI with respect to the traveler’s heading. The angle BC is the heading-relative bearing, as Figure 6−1 shows. Afterwards, I name heading-relative bearing “bearing” for short in this chapter.

Definition 2: Coordinate system of handheld smartphone (compass)

The coordinate system of 3D compass is x,y,z. Assume that a smart phone is perfectly horizontal, as shown in Figure 6−2. X-axis is directed left to right at the top of the screen. Y-axis is directed top to bottom along the left side of the phone. Z-axis is directed to the sky at the left top point. The sensor has 3 values measured in degrees, as follows:

1. Values [0]: they represent the orientation of the phone around the Z-axis. 0 indicates North; 90 represents East; 180 represents South; 270 represents West.
2. Values [1]: they measure inclination of the phone on the X-axis. Values [1] range from -180≤values [1] ≤180. If the table is perfectly horizontal, value [1] is 0 (Figure 2).
3. Values [2]: they measure the angle on Y-axis, and range: -90≤values [2] ≤90. If the phone is perfectly horizontal, values [2] are zero.

A compass calculates all three angles: pitch is values[2], roll is values[1], yaw is values[0], and compass heading. Next, I illustrate how to calculate heading by pitch, roll, and yaw.

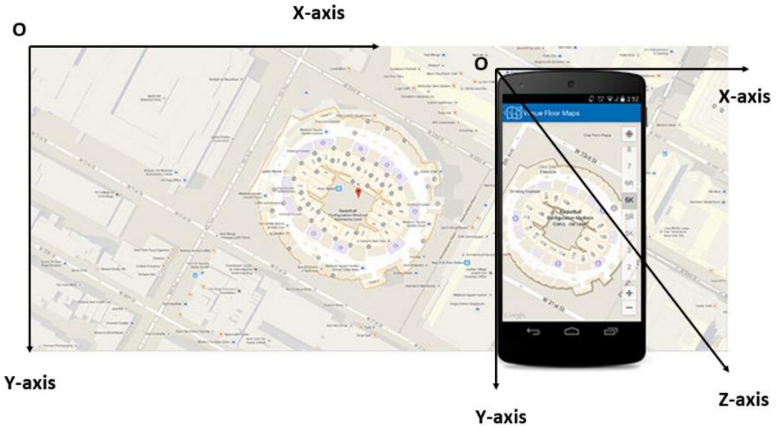


Figure 5-2 The 3D coordinate system of handheld sensor [154]

As in Figure 6−2, the space system of the smart phone is a sub-space of the map, i.e., a relative space. When the smart phone is moving or rotating, the point of the map will be recalculated on the relative space.

Definition 3: Heading Calculation

The ideal situation is compass working on a flat surface. Thus, the heading θ can be calculated as,

$$\theta = tan^{-1}(\frac{Y}{X})$$

However, a user cannot hold the smart phone on a horizontal way when he/she walks. The compass in the phone always rotates or declines. When the compass works on a titled surface, the heading θ will be calculated as follows:

$$\begin{cases} X_h = X * cos\,\alpha + Y * sin\,\gamma * sin\,\alpha - Z * cos\,\gamma * sin\,\alpha \\ Y_h = Y * cos\,\gamma - Z * sin\,\gamma \end{cases}$$

Among it, α is pitch, β is jaw, and γ is roll. The pseudo code of heading θ is calculated as follows.

Pseudo Code of Heading Calculation

```
Begin
      Heading θ =arctan(Yh/Xh);
         if (Xh<0)
              {heading θ =180-heading θ;}
         elseif(Xh>0 && Yh<0)
              {heading θ =-heading θ;}
         elseif(Xh>0 && Yh>0)
              {heading θ =360-heading θ;}
         elseif(Xh==0 && Yh<0)
              {heading θ =90;}
         elseif(Xh==0 && Yh>0)
               {heading θ =270;}
End
```

Definition 4: Relative space coordinates transformation

A map can be considered as a two-dimensional space by two Cartesian coordinates [118]. In a coordinate rotation, point P (x,y) in the map presents in Cartesian coordinate system XY. Let θ represents the heading of the user; a user’s coordinate system based on heading θ is a relative space of the whole Cartesian coordinates. Once the relative space change while walking, a certain conversion relationship is used to transform the previous coordinate space into a new one (translated and transformed one). This conversion relationship can be represented by a transformation matrix. The new matrix will be generated when the sensor rotates, and, hence, the space coordinates.

There is a 2D coordinate space transformation between user’s coordinate space and the map plane. In Figure 6-3, the parallel movement of the coordinate space XOY obtains the coordinate space X ‘‘O’Y’’. Let Point O be the origin of the coordinate space XOY, and move O to the Point O' (a, b) in coordinate space X"O'Y".

So,

$$\begin{cases} x - a = x' \\ y - b = y' \end{cases}$$

After translating the coordinate space, the relative space rotates. As Figure 6-3 presents that, the coordinate space X"O'Y" coincides with the origin of X'O'Y'. X"O'Y" counter-clockwise rotating by heading θ obtains the X’O ‘Y’ system.

So,

$$\begin{cases} x - a = x' \cdot cos\theta + y' \cdot sin\theta \\ y - b = y' \cdot cos\theta - x' \cdot sin\theta \end{cases}$$

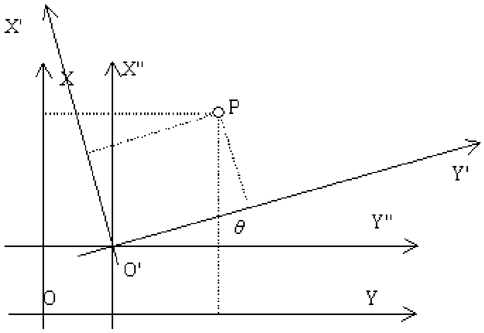


Figure 5-3 The translated and rotated 2D coordinate space [154]

Thus,

$$\begin{pmatrix} x-a \\ y-b \end{pmatrix} = R_z(\theta)\begin{pmatrix} x' \\ y' \end{pmatrix}$$

Among them,

$$R_z(\theta) = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix}$$

Then the new relative coordinate system (x', y') will be obtained.

5.3.2 AR Indoor Navigation

An optimized indoor path planning is the foundation of AR navigation. The path is made of segments and key points (turning points). Obviously, the optimal indoor path fast plans the fewer key points with the minimized segment length, which means the efficient (cost) and the effectiveness (length). Thus, those key points of ACO path make AR indoor navigation feasible and simple. The performant results have already been evaluated in [120], even the distance is not shortest enough (but it is accepted and ACO is an Iterative optimization process, it will be optimized enough), the time cost is largely reduced.

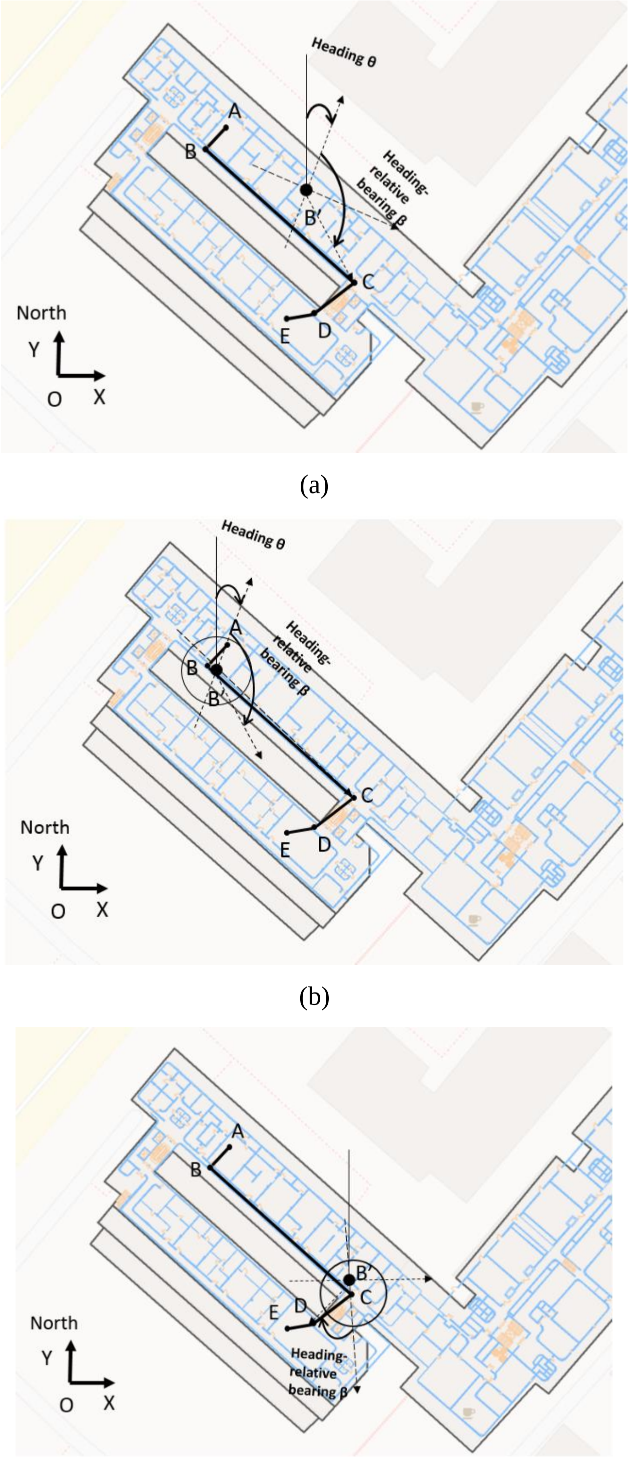
The AR indoor navigation relies on a north-oriented coordinate transformation. Specifically designed for smartphones, it builds an indoor floor map and tracks the movement of pedestrians. No matter the user is on the correct or incorrect path, the system shows the right direction. The main elements of the proposed AR navigation solution are heading and bearing that are based on the compass. Based on the combination of the compass rotation and indoor positioning technology (magnetometer of mobile phone), my AR indoor navigation proceeds as follows:

Step 1: The optimized path, which has been previously calculated by path planning services, is received. That path contains all the key points of the path, A, B, C, D and E, as Figure 6-4 (a) presented;

Step 2: User positioning by magnetometer is based on magnetic fields matching. The system positions the user on current point B' on the map (Figure 6-4 (a));

Step 3: Estimate the offset of the initial orientation of the user versus the geographic north, by magnetic compasses and accelerometers. After obtaining the current 2D space coordinates, heading θ is calculated;

Step 4: The subsequent key point C is found and the transformed matrix is calculated on the updated 2D space coordinates. The bearing (heading-relative bearing) β is based on the new transformed coordinates and the direction is displayed on the smart phone screen.


Heading θ
A
B
B'
Heading-relative bearing β
C
D
E
North
Y
O
X
Heading θ
A
B
B'
Heading-relative bearing β
C
D
E
North
Y
O
X
A
B
B'
C
D
E
Heading-relative bearing β
North
Y
O
X

(a)

(b)

(c)

Figure 5-4 AR Navigation Process, this figure describes user steps into B’, the bearing β will be calculated by B’ and C.

In short, from the above figures, when a user steps into the Point B’, the heading θ is the degree that the smart phone rotates around Z-axis. The new space system is transformed from the Point O to B’ and rotated by θ. The subsequent key point C and the current relative space calculate the heading-relative bearing β. The system then presents the directional arrows to the user, e.g. turn left, turn right, or go straight. In short, the angle/direction is calculated on the smart phone relative space system.

Additionally, as user passing by, the key points of the path are updated. As the user progresses on the path, the travelled path will be cancelled. If a user steps into some place that is within the range of the subsequent point, the previous point will be cancelled. So I only monitor if the user position is within a given range (e.g., 50 pixels).

As Figure 6-4 (b) shows, when the user goes from B to B’, B’ becomes the current point, while the user is in B’s range. The system calculates the heading-relative bearing of B’ and the next key point C. When the user is close to point C’s range, the heading relative bearing is calculated on the target D (Figure 6-4 (c)). In other words, the key points (turning points) are updating while walking on the path. The updating process works alike a queue in and out.

5.3.3 Voice Alert Indoor Navigation for VIPs

The objective is leading the VIP user to the destination with a safe and convenient route. Combination of the incoming pedestrian detection and a voice feedback in the en-route assistant can guide VIPs to avoid dynamic obstacles, and find the best route. The core researches of blind indoor navigation are visual obstacle detection and voice alert. The device I use is mobile phones, with limited CPU power and memories, which have to deal with a sophisticated real-time process of image sequence and a text-to-speech mechanism. When system detects the pedestrian, the voice interface of the system communicates with the user the direction of incoming obstacles (right pedestrians, left pedestrians, center pedestrians, etc.). Hence, the proposed voice alert interface of indoor navigation for VIPs can be designed into two parts:

1. Pedestrian optimization detection: system can detect the incoming pedestrians by a real-time video from inertial camera sensor. I use the Haar cascading classifier based on Android OpenCV library for pedestrian detection and then get the pedestrians' coordinates (top-left coordinate and right-bottom coordinate) on mobile screen. The classifier is a real time function, which means it can deal with every frame of a video. The optimization process deals with every pedestrian detection area to continue to refine/precise the detected area within the first detected one.
2. Voice alert interface: system will present a speech to VIPs based on Android TextToSpeech tool to tell them the positions of incoming pedestrians, which facilitates them walking and keep them safe. In addition, this voice interface is not only a speech interface, it is also a tactile one, which means it can run with vibrations to alert the coming pedestrians.

5.3.3.1 Pedestrian Detection Optimization Process

Normally, a pedestrian is initially detected by Haar cascading classifier. The Haar cascading classifier can offer a rough detected area if pedestrians appear. The matter is that the detected area is not accurate in such limited mobile screen. Thus, system cannot tell an accurate obstacle location to users. Thus, I focus on the accurate detection of coming pedestrian area. I try to filter the blank areas and the obstacle part. The optimized pedestrian detection method is illustrated as follows (Figure 6−5):

Step 1. Convert the received RGB images to gray images. Then, the Gaussian Filter makes the received image smoother. It outputs a `weighted average' of each pixel's neighborhood with an average weight, which is higher than the one of central pixels. Then, I will get image with less noises;

Step 2. I use Otsu’s algorithm to convert the obtained images to binary images. The Otsu's algorithm performs clustering-based image threshold or gray level-based reduction. This algorithm can remove background areas, so that I can get the foreground/important regions of images;

Step 3. The edge of the object in images is obtained by Canny algorithm. The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. Canny algorithm will maximize the important information of object edges and inhibit other useless features;

Step 4. Median Filter is very widely used in digital image processing. Under certain conditions, it preserves edges while removing noise. I use Median Filter to remove noises again so that I can keep the feature points.

Step 5. The feature points are processed, and I get the minimal detected area (rectangle) that includes the max important feature points.

Step 6. Finally, system returns the optimized detection result.

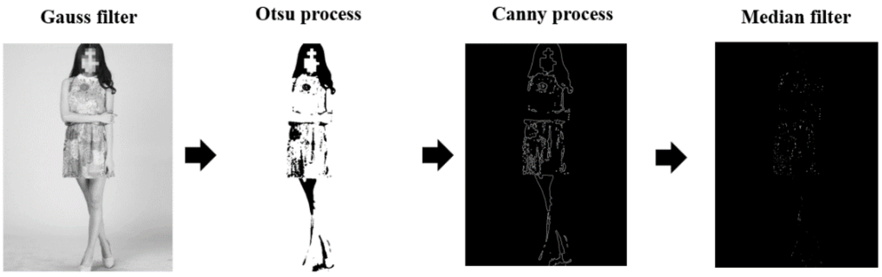


Figure 5-5 The optimized process of pedestrian detection

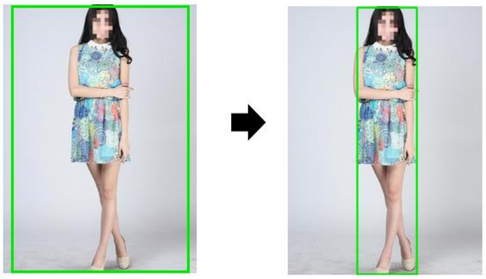


Figure 5-6 Comparison of original detection result and optimized result

In Step 5, I suppose an optimized algorithm to find a minimal rectangle that contains the important/foreground features. The smaller detected area, the more contained features, the more accurate pedestrian detection result. Figure 6-6 presents a comparison of original detection result and the optimized result. We can see that a more accurate result will be obtained after the real-time optimization process. The pseudo code of optimization shows below.

```
Pseudo Code of Haar Cascading Classifier Optimization Pedestrian Detection
Begin
    Initialize the camera view
    Initialize the media sound service
    Initialize the text to speech

    for each CameraViewPicture
           {
                   OrighinalPicture = PreProcess(OrighinalPicture);
                   OrighinalPicture = PedestrianClassifier(OrighinalPicture);
                   image = DetectionAreaCapture(OrighinalPicture);
                   image = GaussFilter(image);
                   image = Otsu(image);
                   image = Canny(image);
                   image = MedianFilter(image);
                   Coordinate = RectangleCapture(image);
                   OrighinalPicture = DrawRectangle(coordinate);
                   PedstrianDistributeType = Distribution (Coordinate);
                   Show(OrighinalPicture);
                   AlertText = ChooseText(PedstrianDistributeType);
                   TextToSpeech(AlertText);
           }
        end for
End
```

5.3.3.2 A Voice Alert Design for VIP Navigation

After detecting pedestrians, I need to present a voice alert to VIP user during his/her navigation. The multi-threaded voice alert on smart phone runs every three seconds. Also, a high-quality, real-time text-to-speech synthesizer handles an unlimited vocabulary with a media speaker, which requires a minimum of memory and computational power. Such speech synthesizer can process the multi-lingual text that even not exists in the dictionary. The efficient voice alert interface manages two modes, voice mode and vibration mode. After evaluation, integration of vibration and voice can be better for safe and convenient trips. Such voice alert interface is designed in two parts:

1. Alert text: Different texts according to different detect results are based on detected coordinates of a mobile screen. For instance, if a coming pedestrian occurs on the left side of the screen, interface presents the alert: “Left finds people”. Same with right side, center side, etc.

2. Text to speech: The voice interface will translate the warning text to speech to VIP users. I use Google’s TextToSpeech tool, since it supports multi-languages. Meanwhile, the interface combines with vibration alert.

Figure 6-7 (a) presents an example of a pedestrian detection result with a green rectangle showing the dynamic obstacles’ positions on VIPs’ ways. Thus, the system presents a speech to alert VIP users: “left and center find people”. Another case (Figure 6-7 (b)) will alert: “right and center find people”. Figure 6-8 alerts: “center finds people”.

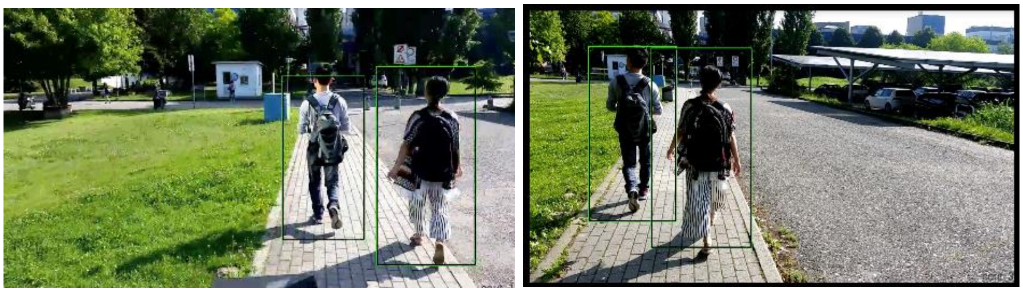


Figure 5-7 Pedestrian detection result (left and center (a)) (right and center (b))

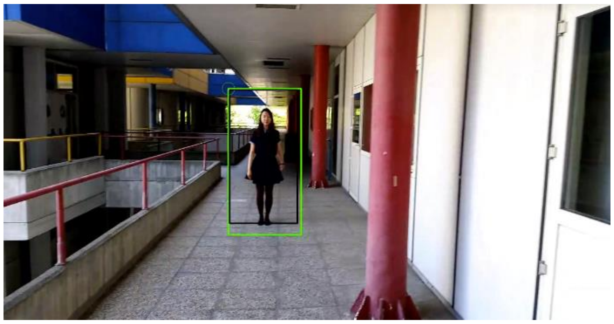


Figure 5-8 Pedestrian detection result (center)

5.4 Evaluation on En-route Assistant

5.4.1 Evaluation on AR Navigation

After optimizing the Ant Colony path algorithm, system only extracts and stores key points in a list. Then it connects every two key points in a sequence, as a straight line, and finally gets the shortest path. Based on the shortest path, AR Indoor Navigation is performing well, because only key points along the path and the course are managed. This system plots the course on a camera view, which displays the real scene. So, the arrow in the camera view augments reality. The camera works as the user’s eye to explore the indoor surroundings and position. After testing 100 trails on the indoor map, the optimization rate of two modes is shown in Figure 6-9.

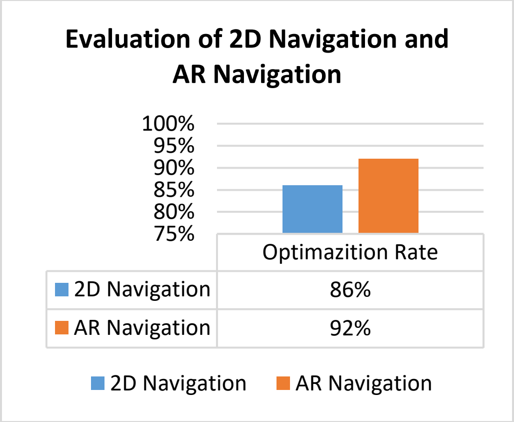
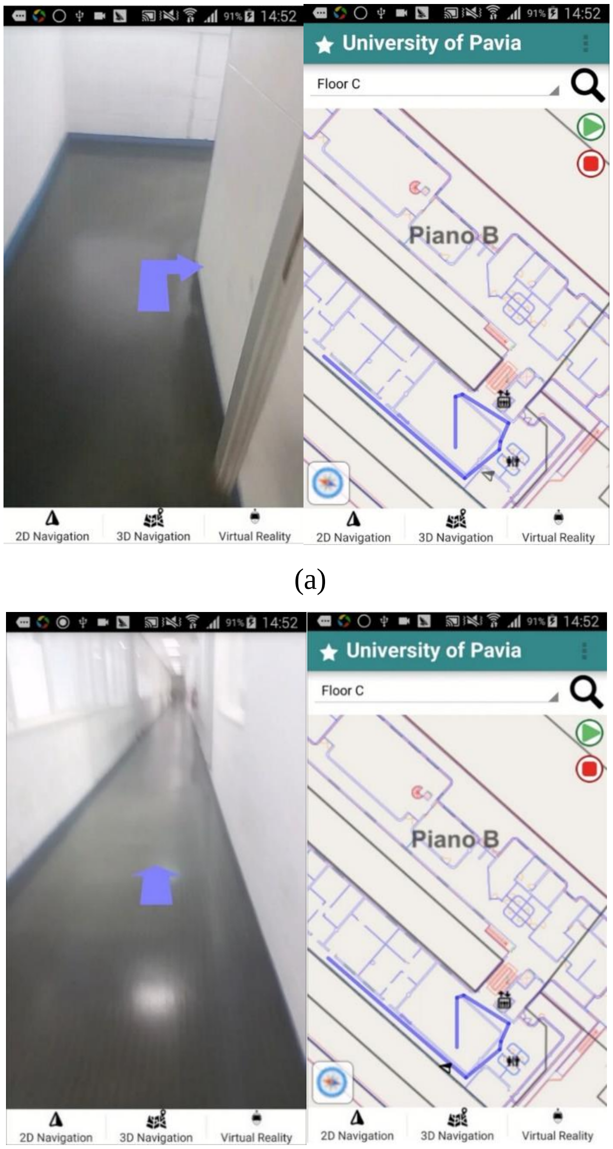


Figure 5-9 Evaluation between 2D navigation and AR navigation [154]

Figure 6-10 shows experimental results of indoor scenes, which compare the user's locations between indoor 2D navigation and AR navigation. At the first time, the vision tracker is walking in the narrow corner of Piano C, and the turn right direction is displayed in the camera view. After his turning right, the arrow changes immediately and gives the go straight direction to go along the corridor. Meanwhile, the user's position on 2D map is updated in real-time and showed the correct location. From the figure, I see that the system achieves precise positioning and direction indication.



(a)

(b)

Figure 5-10 Location comparison between 2D navigation and AR navigation [154]

5.4.2 Evaluation on Voice Alert Indoor Navigation for VIPs

Here, I field test this pedestrian detection optimization algorithm in real scenarios. The ideal sample can perform a better testing result. I have made a series of testing based on some samples. The testing results are presented in Table 6-3, which lists 10 testing results. Based on the testing results, the pedestrian detections are optimized in various degrees, and the reduction of detected area ranges from 4.0% to 44.4%. The average of optimization achieves 23.1%, which validates the good performance.

Table 5-3 Detection result before optimization VS after optimization

Sample ID	Original width (pixel)	Original Height (pixel)	Optimization width (pixel)	Optimization height (pixel)	Reduce percentage
1	203	309	197	274	13.9%
2	384	578	382	558	4.0%
3	49	108	46	64	44.4%
4	120	263	112	224	20.5%
5	133	399	127	320	23.4%
6	98	215	96	159	27.6%
7	72	196	67	129	38.8%
8	101	238	97	196	20.9%
9	90	164	81	146	19.9%
10	75	210	71	182	18.0%
Aveg.	133	268	128	225	23.1%

I present the testing results of optimized pedestrian detection in Table 6-3, Figure 6−11 and Figure 6−12. The original pedestrian detection results of Haar cascading classifier are presented in the first line with green rectangle; while the optimized results are represented in the second line with black rectangle in Figure 6−11. The figures intuitive display the optimization results and compare the original pedestrian detection results and original ones.

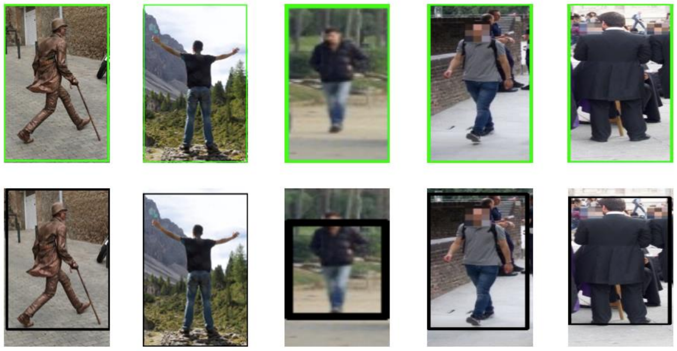


Figure 5-11 Detection result original VS optimized (sample 1-5)

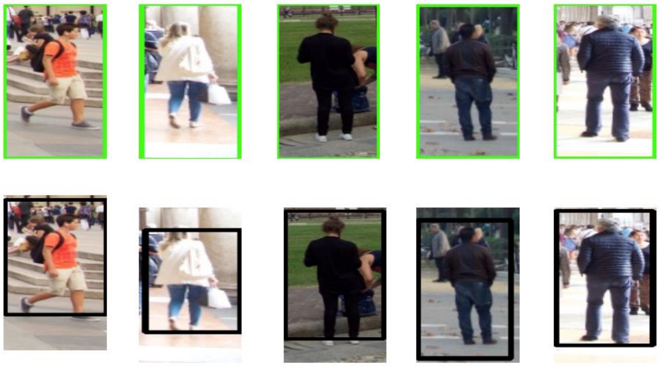


Figure 5-12 Detection result original VS optimized (sample 6-10)

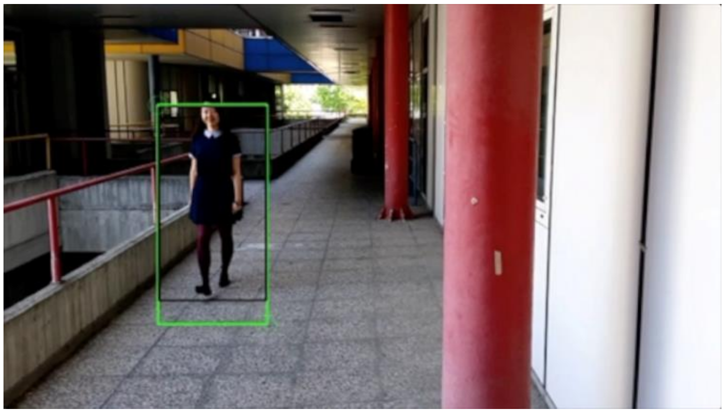
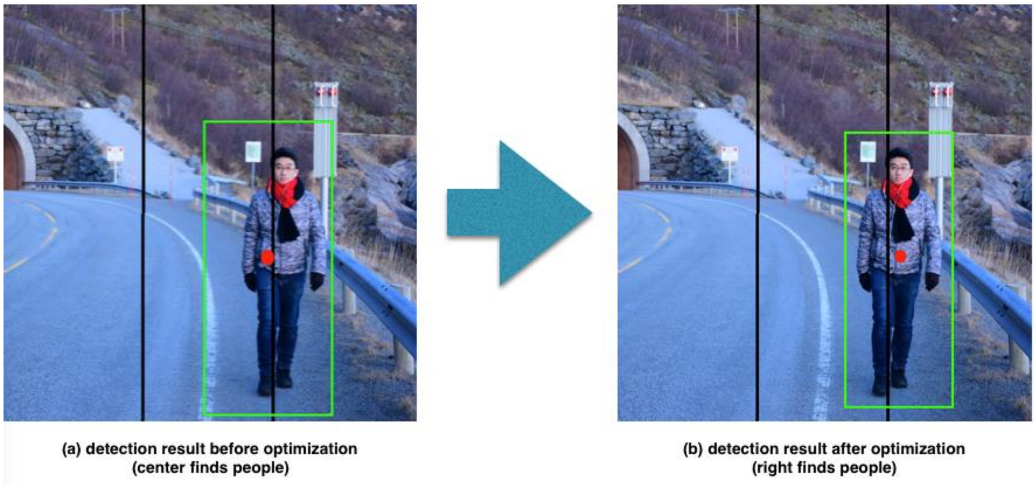


Figure 5-13 Pedestrian detection optimization result

In real indoor scenario, as Figure 6−13 presents, I can get a detected area that is more accurate in the real environment. The smaller black rectangle is the result area after optimization, while the bigger green rectangle is the original detection result of Haar cascading classifier. The filters have removed the non-pedestrian area (black rectangle) in this case, which presents a significantly improvement.



(a) detection result before optimization
(center finds people)

(b) detection result after optimization
(right finds people)

Figure 5-14 An optimized and practical example

Figure 6−14 presents an example, which indicates the practical effect of optimization. I divide the screen into three parts as I mentioned before (left, center and right). I use the center point of pedestrian detected area for judging. The voice interface presents a speech: "center finds people" before optimization (presented in Figure 6−14 (a)), which is not correct. After the optimization of pedestrian detection, the voice alert will be: "right finds people" (presented in Figure 6−14 (b)). In other words, in Figure 6−14 (a), the VIP voice interface makes an incorrect judgment because of the inaccurate detected region. In Figure 6-14 (b), after optimization, the interface alerts the VIP users that the coming pedestrian locates on the right side. From above, we can see that the optimization algorithm corrects the pedestrian detected region and presents a correct warning to VIPs during their navigation. Thereby, the good performance of the proposed voice alert interface enhances indoor navigation services for VIPs.

5.5 Summary

En-route assistant includes effective AR Navigation and Voice Alert Navigation for VIPs. In particular, AR transforms the relative space system and displays on the user's camera a directional arrow with surrounding POI information, thus acting as an artificial eye. It supports a real-time match between 2D and AR indoor navigation (building structure). Additionally, the pedestrian voice alert plays a major role in meeting VIPs' indoor services demands, improving them indoor experiences and reducing the stress of searching paths. For voice alert VIP indoor navigation, I illustrate the optimized algorithm, which is based on the Haar cascading. After a series of testing results, I validate this method with more accurate detection results and safer navigation solutions.

Chapter 6 Indoor Data Recommendation

6.1 Challenge on Indoor Data Recommendation

The urgent needs of personal recommendation are growing with enlarging and complicated indoor environment. For instance, large-scale shopping malls in Milan are visited by travelers, who don't know what they can find and what they are looking for. Visitors are interested in discovering new places or products in a mall. Frequent customers may regular visit some identical POIs often following patterns. Therefore, a POI recommendation service is necessary within a navigation lifecycle, which should discover user's potential Points of Interest (POI).

In recent years, POI recommendation is a hot topic that researches the interest intensity of a person for a given thing. Since the attitude towards a given thing may be different, system will not recommend the same POI to users with different preferences. In this context, a personalized POI recommendation should reflect the different preferences and user behaviors, which are addressed through an analysis, modeling, and forecasting process [124] [125]. In short, a personalized POI recommendation algorithm should be proposed for indoor context.

Location-based Services (LBS) of Indoor Mobility provide mobile users with the current locations and environment-related services or information. A typical Location-Based Social Network (LBSN) [121] provides a platform for user communication, allows users to share and evaluate in the current location [122], and provides a large amount of user behaviour data, such as user social relationships, historical sign-up data, evaluation information [123], etc. Currently, in widely used LBSN, such as foreign Facebook, Foursquare, Weibo, etc., Point of Interest (POI) recommendation becomes a common and practical personal service.

A complete recommendation system is usually composed of three modules: data acquisition module, behavior analysis module, and recommendation algorithm module [126]. Data acquisition module collects user (behavior) information, such as, historical places, user friends' locations, etc. Behavior analysis module analyzes the type of POIs and user interests. Recommendation algorithm model analyzes user preferences and recommends different types of POIs based on various factors, e.g., geographical factor [127], time factor [128] and social relations factor [129]. Among these factors, geographical factor considers the distance between user and POI, time factor considers the time of choosing POI, and social relations factor considers user friends' affinity.

On the other hand, good indoor data quality is determined as reliable data to precisely predict POI pattern. Data quality can be affected by the way of data achievement and data management. In terms of indoor data itself, it is measured in a number of dimensions, including, but not limited to accuracy, completeness, consistency, uniqueness, validity, and timeliness. However, with websites and other campaigns collecting so much data, getting zero errors is challengeable. Which to determine if information is complete and accurate, and what do I do when finding errors or issues? Sina Weibo is one of the biggest social media platforms of China launched in 2009 and it most dominants source of news content, where netizens come for information acquisition, sharing and commenting. In 2015, it has over 600 million registered users, and monthly active users reaching 390 million in September of 2016 [156]. In Weibo, the users are part of a "follower-followee network". The relationship between followers and followees is unidirectional; one can 'follow' an individual and read their 'weibos' (posts), like and share them, without being followed back [156]. Those real-name registered users guarantee the good quality of post data and data security so that Weibo can support precise indoor data analysis and POI prediction.

Moreover, as presented in Figure 7-1, the geography and time factors are fully studied, and fewer studies research on the relations factors [130]. However, in real life, everyone's social relationship is complex, in addition, in the indoor environment, the distance is short and social relation plays an important role. However, the characteristic friend relations of Weibo can also provide massive information of network for indoor data mining. In the next section, the Weibo social data is extracted and designed for indoor data recommendation analysis.

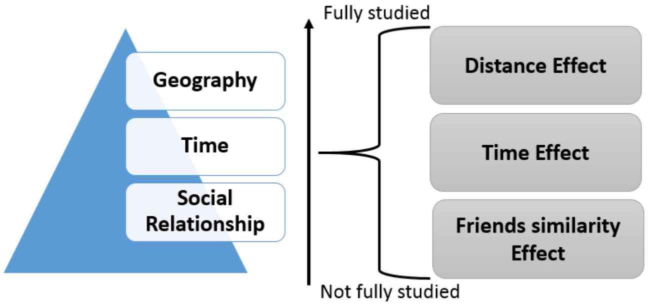


Figure 6-1 Research on the current states of recommended algorithms

6.2 Related Work on Indoor Data Recommendation

Most indoor applications only consider the geo-location of the user [131], and few with emotion context. For instance, Smart Signs, an indoor navigation application, developed by Lijding et al. [132], is able to plan the path based on user preferences, time, weather and current state. An indoor navigation application that developed by Peternier et al. [133] is based on 3D virtual characters to guide user to the destination. C. Barberis et al. consider the user preferences, e.g., elevator preferences, stairs preferences, etc., [120], and they provide the accessible path for disabled users.

Typical LBS social networks, e.g., Foursquare [134], Gowalla [135], Facebook Place [136], etc., allow users to check-in, leave messages, and share their experiences on typical POIs [137]. Thus, social relationship and historical POIs are critical factors for user preferences mining [138].

Check-in data also reflects the user preferences [139]. Berjani et al. [140] propose a POI recommendation algorithm based on check-in data by regular matrix factorization. They define a POI hierarchy to analyze user preferences.

Some researchers regard the POI as a product to recommend. Ye et al. [124] propose a product and user-based fusion recommendation algorithm. This algorithm assumes that similar users have similar preferences, thus, it recommends the user with the highest similar rate without data sparse. Finally, their recommendation algorithm performs well.

Geographic factor mainly considers the impact on distance between user and POI. Yuan et al. [128] rank the Bayesian probability on check-in distance and recommend POI with highest ranking. Cheng et al. [129] consider users always check-in around somewhere. Therefore, they find the center of interesting area based on Gaussian model and a greedy clustering algorithm. Ye et al. [141] propose a friend-based collaborative filtering algorithm that considers the friend's preference based on friend affinity. In their later work [128], they continue to improve their algorithm with the user's social relationship and POI check-in data. Yuan et al. propose a time-aware recommendation algorithm [128], which predicts the time of going to POI. They classify the historical check-in data and calculate the user similarity by time period. A POI recommendation algorithm based on matrix factorization is proposed by Gao et al. [142]. System can recommend users different POIs in different time periods.

The research [149] on Facebook shows that 40% of friends watched the same movies. Bo et al. [150] conduct the same analysis on the same data set [149], and find that only 20.6% of friends co-check at one POI. Bo et al. [150] extracted the check-in POI with a set of tags, i.e., “music”, “shopping” or “food” in Foursquare. They used LDA (Latent Dirichlet Allocation) to obtain the topic distribution of the dataset, by assuming POI as “documents” and tags as “words”. Results show that a very high percentage (about 80%) of friends shares topics. From above, social relationship occupies larger in POI recommendation.

In fact, social information, which includes social relations and social topics, is rich, heterogeneous in social network. Social topic model is applied to discover the topic distribution of social relationships and the topic distribution of POI. Here, I data mining on both the social relationships from social networks and the POI information from indoor navigation to propose a recommended algorithm. It uses social relations to enhance the accuracy of the recommended algorithm from a large number of user behavior data in social network. Indoor Mobility also provides a social relationship mining model to recommend POI by a classification mapping between social relationship and POIs.

6.3 Contribution on Indoor Data Recommendation

6.3.1 Social Relations Analysis

In this section, I classify user social relations based on user intimacy, roles and interests from massive user information and social information in social network.

- User information, includes the following information (followers and followings) and profile information (user personal information, such as basic information, educational information, job information, and user preferences of tag information, etc.). The following information reflects the social affinity: a unilateral following is less intimate than a mutual one.

- Social information, refers to the user's post, that is, interaction/communication between friends. Then, I can classify user’s social relations, such as, classmates, colleagues, relatives, etc.

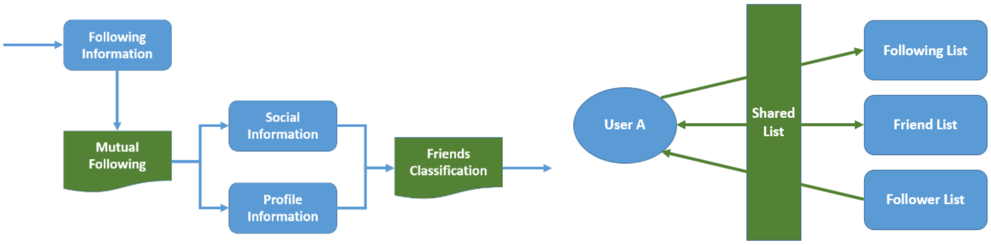


Figure 6-2 Framework of social relations model

Figure 7-2 presents the framework of social relations model. In the model, blue boxes indicate the user behaviors, and green boxes indicate the analysis of user behaviors.

6.3.1.1 User Information Analysis

I analyze the user similarity by various attributes: 1) Affinity. I define affinity with a threshold, and here I only consider the affinity group; 2) Gender. Human interest is usually gender-related, therefore, gender will be used as an additional factor to analyze user similarity; 3) Experience. I also analyze collect user education and working experience. Thus, the user information similarity between $user_i$ and $user_j$ can be calculated by above attributes:

$$\mathrm{sex}Sim_{i,j} = \begin{cases} 1, & for\ sex(i) = sex(j) \\ 0, & for\ sex(i) \neq sex(j) \end{cases}$$

$$\mathrm{age}Sim_{i,j} = \frac{1}{|age(i) - age(j)| + 1} + sexSim$$

$$\mathrm{edu}Sim_{i,j} = \begin{cases} \dfrac{1}{\lg(edu_i - 1) + 1} + simSex, for\ edu(i) = edu(j) \\ 0, \qquad\qquad\qquad\qquad for\ edu(i) \neq edu(j) \end{cases}$$

$$\mathrm{job}Sim_{i,j} = \begin{cases} \dfrac{1}{\lg(job_i - 1) + 1} + simSex, for\ job(i) = job(j) \\ 0, \qquad\qquad\qquad\qquad for\ job(i) \neq job(j) \end{cases}$$

edu_i means the number of people in user $user_i$’s school, and job_imeans the number of $user_i$’s colleagues. The tag similarity tag$Sim_{i,j}$ is complex, and it can be calculated by following two steps:

Step 1: Calculate similarity between two tag words by word2vec;

Step 2: Define tag similarity as tag$Sim_{i,j}$ and get the highest similar tag pairs, which calculates as follows:

$$\mathrm{tag}Sim_{i,j} = maxSim\ (\omega_m, \omega_n)$$

Among it, m means the all tag words of $user_i$, n means all tag words of $user_j$.

In fact, in many LBS social networks, some user information is missing because of no mandatory and privacy. I mine the user information in order to better understand user relations with the greatest possible classification.

6.3.1.2 Social Information Analysis

Social information mainly relates to the user's posts and interactions (forward, comment, compliment, and mention someone). These contain rich content, such as, user behaviors, user preferences, intimacy, activities, trips, emotions, etc. Here, I divide user social relations into two categories: intimate relationship, such as, parents, relatives, lovers, children, etc., and normal relationship, such as, colleagues, classmates, etc.

In order to completely analyze the user social relations by such rich content, I propose a Subject Clustering Model (SCM) and two mining algorithms: 1) POI-based relations mining algorithm, which analyses the intimate relations from interactions; and 2) subject-based relations mining algorithm that analyses the normal relations from user group activities.

6.3.1.2.1 Subject Clustering Model (SCM)

The proposed Subject Clustering Model (SCM) is based on LDA (Latent Dirichlet Allocation), which is a 3D Bayesian probability model for document generation. LDA consists of three-tier structure of words, subjects, and documents. The probability of each word appears is calculated by LDA model that can be represented by a matrix presented in Figure 7-3.

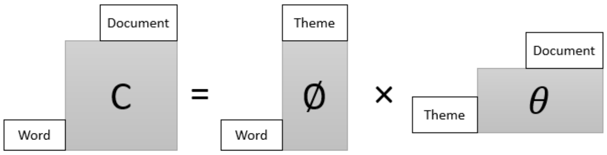


Figure 6-3 LDA Probability Formula

C represents the word frequency in each document, and Ø represents the probability of word occurrence in each subject; and θ represents the probability of subject occurrence in each document. Thus, the detailed LDA algorithm is designed as follows:

Step 1: Select a document d from document set D (d ∈ D), remove the meaningless words and get the frequent ones. Then, I obtain the expected word set W $= \{w_1, w_2, w_3, \ldots, w_n\}$;

Step 2: Statistic of the words $w_i(w_i \epsilon\ W)$, and then get probability of each word appearance p (w_i|d);

Step 3: Randomly assign the word $w_i(w_i \epsilon\ W)$ to a subject as an initial subject;

Step 4: Resample the subject t for word w_i (w_i ∈ W) by Gibbs Sampling;

Step 5: Repeat the process from step 2 to step 4 until Gibbs Sampling converges. The result of the convergence is the probabilistic distribution of the subject-word;

Step 6: Back to step 1 and re-select other documents d (d ∈ D), and repeat from step2 to step5 process until all the documents D are complete;

The above process can be illustrated by a model, as Figure 7-4 presents:

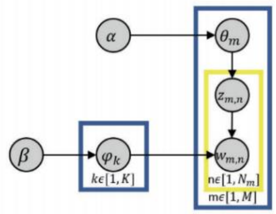


Figure 6-4 LDA Probability Formula

In Figure 7-4, α is a priori parameter that denotes the polynomial distribution of Dirichlet among subjects; β denotes the one among words in each subject. θ_m presents the subject distribution under m^{th} document. φ_k presents the word distribution under k^{th} subject. $z_{m,n}$ presents the n^{th} word in the m^{th} document. K presents the number of subjects, M presents the number of documents, and N presents the total number of words in all documents. Thus, the two main processes of LDA model can be divided:

1. $\alpha \rightarrow \theta_m \rightarrow z_{m,n}$. This process shows that in α case, I randomly select the subject distribution θ_m in the m^{th} document, and then randomly select the $z_{m,n}$ subject in the *n*th word.
2. $\beta \rightarrow \varphi_k \rightarrow w_{m,n}|k = z_{m,n}$. The process selects the subject of k = z_m among K subjects to get the word distribution of this subject. Then, I get the word $w_{m,n}$ from this word distribution.

Actually, the whole process is a Dirichlet-Multinomial conjugate structure. Therefore, the formula can be obtained:

$$p(z_m|\alpha) = \frac{\Delta(n_m + \alpha)}{\Delta(\alpha)}$$

Among it, $n_m = ((n_m)^1, (n_m)^2, \dots, (n_m)^K)$, $(n_m)^k$ presents the number of words in k^{th} subject of m^{th} document. In addition, the subjects are generated independently in documents M, I continue to generate probabilities in whole corpus, as the formula presents:

$$\mathrm{p(z|\alpha)} = \prod_{m=1}^{M} p(z_m|\alpha) = \prod_{m=1}^{M} \frac{\Delta(n_m + \alpha)}{\Delta(\alpha)}$$

Same, under k = $z_{m,n}$, we can see the second process is still based on Dirichlet-Multinomial conjugate structure:

$$\mathrm{p}(w_k|\beta) = \frac{\Delta(n_k + \beta)}{\Delta(\beta)}$$

Thus, the probability of word generation in the whole corpus is:

$$\mathrm{p(w|z,\beta)} = \prod_{k=1}^{K} p(w_k|z_k,\beta) = \prod_{k=1}^{K} \frac{\Delta(n_k + \beta)}{\Delta(\beta)}$$

From above, I can get the formula of LDA Joint Probability:

$$\mathrm{p(w,z,|\alpha,\beta) = p(z|\alpha)p(w|z,\beta)} = \prod_{m=1}^{M} \frac{\Delta(n_m + \alpha)}{\Delta(\alpha)} \prod_{k=1}^{K} \frac{\Delta(n_k + \beta)}{\Delta(\beta)}$$

In LDA model, there are some unknown parameters to be trained. They are the topics distributions over document θ_d and the words distributions over topic φ_K. The widely used learning method with these two parameters is Gibbs sampling.

Gibbs sampling is a kind of Markov chain Monte Carlo with a particular stationary distribution. It is usually used to calculate the mobility probability of two parallel points in the n-dimensional spaces. For example, there are two two-dimensional points, $a(x1, y1)$ and $b(x1, y2)$, the mobility probability $p(a-> b)$ from point a to point b is $p(y2|y1)$. It can be extended to n-dimensional spaces. In the model, the word corresponding document and the topic can be regarded as two dimensions, so that each word in different document and topic can be treated as different points in 2D spaces. With Gibbs sampling, the two points transfer, which is called re-sampling. Resample until the Gibbs sampling equation is convergent. Then I obtain the topic distributions over document θ_d and the word distributions over topic φ_K. The equation of Gibbs sampling is showed below.

$$p\left(z_{m,n} = k|{\sim}Z_{m,n}, W\right) \propto p\left(z_{m,n} = k, w_{m,n} = t|{\sim}Z_{m,n}, {\sim}W_{m,n}\right) = \theta_m \cdot \varphi_k$$

In the formula, $z_{m,n}$ means the topic of word *n* in document *m* in the topic library *Z*, and ${\sim}Z_{m,n}$ denotes topics exclude $z_{m,n}$. $w_{m,n}$ means the word *n* in document *m*, and ${\sim}W_{m,n}$ denotes words exclude $w_{m,n}$. θ_d and φ_K are demonstrated by the following formulas.

$$\theta_m = \frac{n_{m,\sim(m,n)}^{(k)} + \alpha k}{\sum_{t=1}^{V}(n_{m,\sim(m,n)}^{(k)} + \alpha k)}$$

$$\varphi_k = \frac{n_{k,\sim(m,n)}^{(k)} + \beta k}{\sum_{t=1}^{V}(n_{k,\sim(m,n)}^{(k)} + \beta k)}$$

Then it is obtained that the equation of Gibbs sampling is expressed below.

$$p\left(z_{m,n} = k|{\sim}Z_{m,n}, W\right) \propto \frac{n_{m,\sim(m,n)}^{(k)} + \alpha k}{\sum_{t=1}^{V}(n_{m,\sim(m,n)}^{(k)} + \alpha k)} \cdot \frac{n_{k,\sim(m,n)}^{(k)} + \beta k}{\sum_{t=1}^{V}(n_{k,\sim(m,n)}^{(k)} + \beta k)}$$

Actually, the right side of the formula equals $p(t|d) \cdot p(w|t)$. Because there are K topics, so the physical significance of Gibbs sampling is sampling in the K routes. Therefore, I can rank the probability of belonging subject based on the subject classification.

6.3.1.2.2 POI-based Social Relations Mining Algorithm

POI-based social relations mining algorithm discovers intimacy users. This algorithm focuses on frequent user interactions in particular scenarios, e.g., festivals, birthday parties, anniversaries, etc. The algorithm is designed as follows:

Step 1: Define a subject corpus T, T = $t_1, t_2, \dots, t_n$ for a specific POI, such as, Valentine, love and gifts, etc.;

Step 2: Extract post collections pub_{texts} from the user, which contain the text @$_{\text{texts}}$ and user information @$_{\text{users}}$;

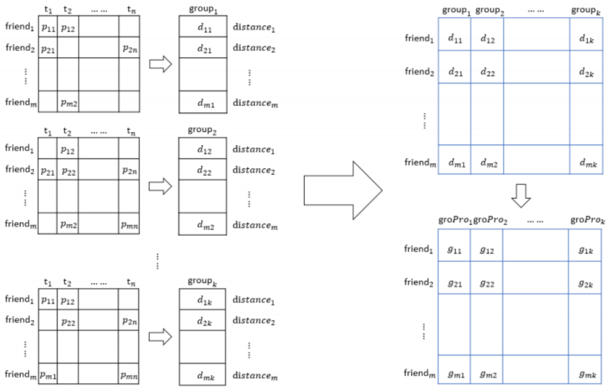
Step 3: The extracted user ID as the primary key in user collections, and then classify the text. Actually, each user's text can be regarded as a group;

Step 4: Analyze the related text collections of user u (u ∈ @$_{\text{users}}$) by subject clustering model, then get the subject probability distribution $U_{subjects}$.

Step 5: Select the specific subject $Subject_T$, and calculate the distance dis between the user and POI;

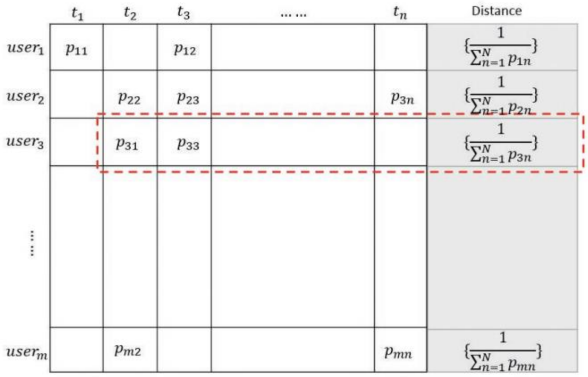
Step 6: Repeat the process from Step 4 to Step 5 until all the users are completed;

Step 7: Rank the calculated distance, and get the nearest user û and his/her intimate relationship r̂.



t_1 t_2 …… t_n
$friend_1$
$friend_2$
$friend_m$
p_{11} p_{12}
p_{21} p_{2n}
p_{22}
p_{m1} p_{m2} p_{mn}
$group_1$
$group_2$
$group_k$
d_{11} d_{21} d_{m1}
d_{12} d_{22} d_{m2}
d_{1k} d_{2k} d_{mk}
$distance_1$
$distance_2$
$distance_m$
$group_1$ $group_2$ …… $group_k$
$groPro_1$ $groPro_2$ …… $groPro_k$
g_{11} g_{12} g_{1k}
g_{21} g_{22} g_{2k}
g_{m1} g_{m2} g_{mk}

Figure 6-5 The schema of friends relation distance calculation



	t_1	t_2	t_3	……	t_n	Distance
$user_1$	p_{11}		p_{12}			$\{\frac{1}{\sum_{n=1}^{N} p_{1n}}\}$
$user_2$		p_{22}	p_{23}		p_{3n}	$\{\frac{1}{\sum_{n=1}^{N} p_{2n}}\}$
$user_3$		p_{31}	p_{33}			$\{\frac{1}{\sum_{n=1}^{N} p_{3n}}\}$
⋮						
$user_m$		p_{m2}			p_{mn}	$\{\frac{1}{\sum_{n=1}^{N} p_{mn}}\}$

Figure 6-6 The schema of distance between specific POIs

Figure 7-5 details the calculation process of the user-subject probability distribution, where the row of matrix presents the $@_{users}$, the column presents the corpus T based on specific POI o_i, and N is the number of T set. The value $p_{i,j}$ in the matrix presents the probability of user i in subject j. In the figure, the $user_i$ locates in maximum distance (marked in red in Figure 7-6). The distance Dis_i between the user i and particular POI is the reciprocal of the sum of all subject probabilities. The smaller Dis_i, the greater the probability belonging to current POI. The formula is defined as follows:

$$Dis_i = \sum_{n=1}^{N} \frac{1}{p_{i,n}}$$

According to the above algorithm, I can get the user's intimate social relations, such as parents, relatives, lovers, etc. The intimacy labels each user in separated groups. The Pseudo Code of this algorithm is:

```
Pseudo Code of POI-based Relations Mining Algorithm
Input:
    The corpus T of specific POI;
    The user;
Output:
    Intimate relationship $(\hat{u}, \hat{r})$       //Return the intimate relations of user
    $T_@ \leftarrow get(pub_{texts}, @_{texts})$     //Get text collections from the user
    $U_@ \leftarrow get(pub_{texts}, @_{users})$     //Get user collections
    $U_{texts} \leftarrow group\ by(userID)$    //Group the text from the same user
    for each i ← 1 to $U_@$ do
        $U_{topics} \leftarrow get(U_{text}, LDA\ model)$  //Get the distribution of subject
        while t ∈ $U_{topic}.equals(t \in T)$ do
            add Subject$_T$                    //Select related subject
            dis ← cal($Topic_T$)         //Calculate the distance between POIs
        end while
    end for
```

6.3.1.2.3 Subject-based Social Relations Mining Algorithm

The subject-based social relations mining algorithm focuses on normal social relations, such as, classmates, colleagues, interest groups, etc., which mainly covers several events, such as, classmates' meetings, hiking, some public activities, etc. The algorithm is designed as follows:

Step 1: Get all the posts Texts of user's friends. I define them as $\text{Friends} = \{friend_1, friend_2, \dots, friend_n\}$, n is the number of friends;

Step 2: Extract the user's interactive information and the relevant text with friends that are defined as $\text{FriTexts} = \{friTexts_1, friTexts_2, \dots, friTexts_k\}$, $friTexts_i$means the text set of i th friend, k means the number of texts;

Step 3: Cluster the subjects from all text collections $friText_i$ by LDA model, and access to the probability distribution of this friend's subject, $\text{ProDis}[friTopic_i]$;

Step 4: Filter and remove the meaningless text, then, I obtain a new subject probability distribution, $\text{NewProDis}[friTopic_i]$;

Step 5: Calculate the distance between each friend $friText_i$ and his/her group $Group_s = \{group_1, group_2, \dots, group_m\}$, and return the nearest grouping $group_i$, which means this friend belonging to current group, m is the number of all groups.

- System calculates the distance between each subject and each group.
- System calculates the probability of the friend and each group, as Figure 7-6 presented.

$\text{T} = \{t_1, t_2, \dots, t_n\}$ presents the subject set of the group $group_k$. The unclassified friend set is $\text{Friends} = \{friend_1, friend_2, \dots, friend_k\}$, p_{ij} presents the probability of friend i belonging to subject j. d_{ij} presents the distance between i^{th} friend and j^{th} group. Finally, the probability of i^{th} friend belonging j^{th} group $g_{i,j}$ is calculated, as formula presents:

$$Dis_i = \sum_{n=1}^{N} p_{i,n}$$

$$groPro_i = \frac{g_{i,j}}{\sum_{k=1}^{K} g_{i,k}}$$

I set the threshold value θ_g 0.3. If the probability of friend $friend_i$ belonging to the groPro$_{j}$, $g_{ij} > \theta_g$, then, $friend_i$ belongs to $groPro_j$.

Step 6: Repeat Step3 and Step5, until it completes all the text of friends FriTexts.

The Pseudo Code of this algorithm is:

```
Pseudo Code of Subject-based Relations Mining Algorithm
Input:
    The user $user_s$;
Output:
    Friends group $Group = \{group_1, group_2, \dots, group_m\}$
        //Return friend groups
    $Text_s \leftarrow get(pub_{texts}, user_s)$    //Get text collections of user's posts
    $Friend_s = \{friend_1, friend_2, \dots, friend_n\} \leftarrow get(friend_s, user_s)$
        //Get unclassified followings
    $FriText_s = \{friText_1, friText_2, \dots, friText_k\} \leftarrow get(Text_s, Friend_s)$
        //Get the text from above user information
    for each i $\leftarrow 1$ to $n$ do
        $ProDis[firTopic_i] \leftarrow LDA(friText_i)$
            //Get the probability distribution of friends-subject
        $NewProDis[firTopic_i] \leftarrow filter(ProDis[friTopic_i])$
            //Filter out the meaningless words and get new distribution
        for each i $\leftarrow 1$ to $n$ do
            Dis$_{ij} \leftarrow$ cal$(friText_i, group_j)$
                //Calculate the distance between POIs
        end for
    end for
```

6.3.2 Personal POI Recommendation Algorithm Design

The proposed personal POI recommendation is based on a mapping relationship between POI categories and friend categories by two algorithms, POI-based social relations and subject-based social relations. First, I mine social relations by the proposed social relations model and classify the POIs based on indoor context. Then, I establish the mapping relationship between different POI categories and social relation types. Finally, system recommends the particular POI with the corresponding social relationship using the improved collaborative filtering recommendation algorithm. The framework of the personal POI recommendation algorithm is presented in Figure 7-7.

However, the critical issues of this POI recommendation algorithms are: 1) how to classify POIs; 2) how to map the social relation categories and POIs, and 3) how to recommend the accurate POI to the user based on his/her social relations.

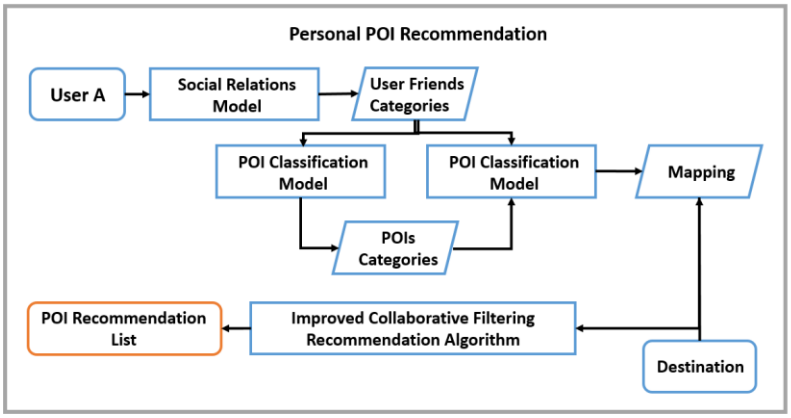


Figure 6-7 The framework of POI recommendation algorithm

6.3.2.1 POI Classification Model

The POI classification follows the user social relations classification. Generally, a POI contains several attributes: $< POI_{id}, POI_{name}, \text{Image}, \text{Description}, \text{lat}, \text{lon} >$. I train and classify the POI data through machine learning in Figure 7-8: 1) characterize the POI data as features; 2) select the conducive feature for model prediction; 3) select the model, and train the sample features for classification.

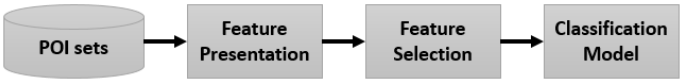


Figure 6-8 The framework of POI classification

1. **POI Feature Presentation**

The attributes of POI provide valuable information. I apply the traditional Vector Space Model (VSM) to analyze the attributes $\text{V} = \{w_1, w_2, w_3, \ldots, w_n\}$. A feature in the model is a semantic unit of text. $w_1$denotes the weight of the vector V that corresponding to the i^{th} characteristic term t_i.

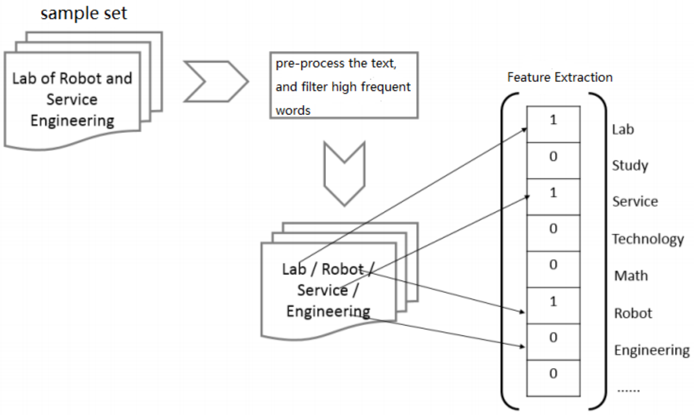

sample set
Lab of Robot and Service Engineering
pre-process the text, and filter high frequent words
Feature Extraction
Lab / Robot / Service / Engineering
1
0
1
0
0
1
0
0
Lab
Study
Service
Technology
Math
Robot
Engineering
......

Figure 6-9 A SVM model dictionary generation

The VSM model trains the sample at first to get a dictionary. Figure 7-9 presents an example of a dictionary generation. During the process, I first pre-process the sample text collection, remove some high frequency words (modal and conjunction words), then generate features. The process of vectorising text is as follows:

Step1: Create a new vector whose length is the same as the length of the dictionary. Each position of vector corresponds to a word in the dictionary.

Step2: Find the corresponding position of the feature in the vector, and fill the weight of features by Boolean operation.

2. POI Feature Selection

The dictionary contains all the POI features, for instance, Lab, Engineering, Science, Math, Center, etc. Firstly, I extract the meaningful features and remove meaningless words, like “Center”, by information gain method based on information influence. The greater changes of information influence, the more importance of the feature word in the document.

In this section, I use information entropy to define the amount of information. $T_i = \{T_1, T_2, \dots, T_n\}$ are the information sources corresponding to the probabilities $P_i = \{P_1, P_2, \dots, P_n\}$. Various situations are independent with each other. The definition is shown in the following equation:

$$\mathrm{H(T)} = -\mathrm{E}[-\mathrm{log}p_i] = -\sum_{i=1}^{n} p_i log p_i$$

Information entropy formula presents probabilities of information occurrences, then, the distribution is obtained. The higher probability, and the more widely propagated, which means the more valuable information. The following conditional probability formula represents the amount of information:

$$\mathrm{H(T|X)} = \sum_{x} p(x) H(T|X = x) = -\sum_{x}\sum_{t} p(x,t) log_2 p(t|x)$$

The information gain is obtained as follows:

$$\mathrm{IG(X)} = \mathrm{H(T)} - \mathrm{H(T|X)}$$

As can be seen from the above, the information entropy is an important measure of the entire text content. Afterwards, I rank the information gain of each feature and select the Top N features with a higher correlation among classification attributes.

3. Classification model

I construct the classification model by Naive Bayesian based on the selected features. Naive Bayesian is simple, requires less estimated parameters, not sensitive to the missing data, and supports the classification training by independent attributes. Naive Bayesian model is quit suitable for this method, since the text information of POI are nouns (names or introductions), the correlations between them are not very high. It calculates the posterior probability under the prior probability for maximum one. The Naive Bayesian Classification Model is as follows:

$$\mathrm{P}\left(\mathrm{Y} = y_j \middle| \mathrm{X}\right) = \frac{\prod_{i=1}^{r} P(X_i|Y = y_j) P(y = y_j)}{P(X)}$$

In addition, the evidence P(X) is a constant, for all classes Y, P (X) is fixed, so I can simplify the above formula:

$$\mathrm{P}\left(\mathrm{Y} = y_j \middle| \mathrm{X}\right) = argmax_{y_i} P(Y = y_j) \prod_{i=1}^{r} P(X_i | Y = y_j)$$

Since the features are discrete, the conditional probability $\mathrm{P}(X_i|\mathrm{Y} = y_j)$ in Naive Bayesian Classification model can be estimated by the proportion of the training examples (the attribute in the class y_j equals to x_i). After the above process, I get the detailed POIs classification results.

6.3.2.2 POI Classification Mapping Algorithm

Table 7-1 presents a mapping relationship after a series calculation on relationship between social relations and POI categories. One category of friends can be associated with more than one categories of POIs. Thus, the mapping relationship of POIs and friends is n:1. I name the mapping algorithm “FtP(Friends to POIs)”:

Step 1: Get the POI collections from each friend, as $POI_s = \{POI_1, POI_2, \dots, POI_n\}$, Calculate the distribution of POIs of each friend in a category. The probability $p_{i\ jk}$ (the friend i in category j and select the POI k), can be calculated using the following formula:

$$p_{i\ jk} = \frac{Num_k}{\sum_{k=1}^{K} Num_k}$$

Where, Num_k represents the number k of POIs that interests the friend j, and the denominator indicates the number of all POIs. Then, I can get the POI distribution of friend j $P_{ij} = p_{ij1}, p_{ij2}, \dots, p_{ijK}$, K is the number of POI categories. That distribution can be presented by the following matrix:

$$\begin{bmatrix} p_{i11} & \cdots & p_{i1k} \\ \vdots & \ddots & \vdots \\ p_{im1} & \cdots & p_{imk} \end{bmatrix}$$

Step 2: Calculate the distribution of POIs in each category of friends. The correlation between friend i and POI category k is:

$$R_{ik} = \prod_{j=1}^{m} p_{i\ jk}$$

In this equation, I can get the correlations between all the friends and POIs, which is displayed in the following matrix:

$$\begin{bmatrix} R_{11} & \cdots & R_{1k} \\ \vdots & \ddots & \vdots \\ R_{n1} & \cdots & R_{nk} \end{bmatrix}$$

Each column represents the correlation between a POI category and a friend category. Then, I generate a table to represent the relationship mapping, as shown in Table 7−1 below.

Table 6-1 The relationship mapping of friend categories and POI categories.

	POI Category 1	POI Category 2	….	POI Category K
Friend Category 1		⇋	….	
Friend Category 2	⇋		….	
….	….	….	….	….
Friend Category N			….	⇋

6.3.2.3 Personal POI Recommendation Algorithm

In this section, I introduce a personal POI recommendation algorithm that combines the relationship mapping with the improved collaborative filtering recommendation algorithm.

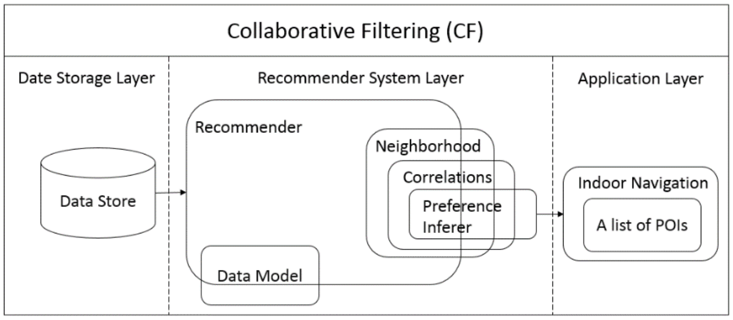


Figure 6-10 The framework of Personal Recommendation Algorithm

As Figure 7-10 presents, the three layered framework of personal recommendation algorithm is:

1) Data storage layer. It is mainly used for storing data, including the searched destinations, POI classifications, friend categories, posts, etc.
2) Recommendation layer. This is the core layer of the framework. Collaborative filtering mines minor data with high similarity to the target user, then according to their preferences, lists POIs to the target user. From the figure, I first need to build a data model to standardize the data, then calculate the user preferences between target user and his/her friend categories. Finally, it gets the correlations between them.
3) Application layer. It makes the adjustments for POI recommendation in the specific scenarios.

According to the different references, the Collaboration Filtering (CF) algorithm can be divided into two categories. One is collaboration filtering based on the users and the other is collaboration filtering based on the POI. The collaboration filtering based on the users focus on the similarity between the target user and his/her friends. That is, this kind of algorithm consider the correlation between users from the perspective of people. The algorithm recommends POI mainly based on the different preference degree of the target user towards indoor POI, which is extracted from user's historical POI collection. The correlation values between the target user and his/her friends are calculated based on the different preference values over a same topic. Figure 7-11 and Table 7-2 present a user-based collaborative filtering recommendation algorithm. In this example, we can see that compared with B, C is more similar to the A. The core idea of this algorithm is to discover the relevance of the user's preference through POI histories, and then select the one with highest relevant.

The historical POIs are listed in the Table 7−2. From the table, we can see that the similarity between POI A and POI C is high. For the target user C, I will recommend POI C in the case of only selecting POI A.

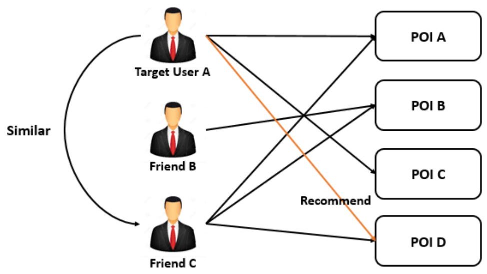


Figure 6-11 The figure of user-based collaborative filtering recommendation algorithm

Table 6-2 An example of POI set for the target user and his/her friends

	POI A	POI B	POI C	POI D
Target user A	♡		♡	
Friend B		♡		
Friend C	♡		♡	♡

For the two collaboration filtering methods, they both have their own benefits and the drawbacks. The collaboration filtering based on the user method is suitable for the situation that the number of users are much bigger than the number of the indoor POI, and the indoor POI data is quite stable. Therefore, the comprehensive collaboration filtering method combined with the collaboration filtering is adopt, which is based on the users method and POI method. The collaboration filtering based on the POI method can improve the diversity of POI recommended, while the collaboration filtering based on user method can increase the accuracy and precision of recommendation. Therefore, in some ways, the two methods are complementary. Comprehensive collaboration filtering method is utilized in POI recommendation.

1. **Preference definition**

Generally, the preference of users often means the users' interested degree towards different things. In the application scenarios of most recommendation algorithms, the preference values are marked by the users themselves. For example, in the movie recommendation, the film scores are usually marked by the users. However, for the indoor navigation systems, there are not many uses in the indoor navigation systems who often score indoor POI, although the indoor navigation systems allow the users to score. Therefore, a specific method is designed to calculate the users' preference values toward each POI.

The users' preference calculation toward each POI is based on the social categories information and indoor POI categories. The structure of indoor POI categories is illustrated as the Figure 7-12. In the figure, Num_i represents the number of all the user's historical visited records toward POI_i.

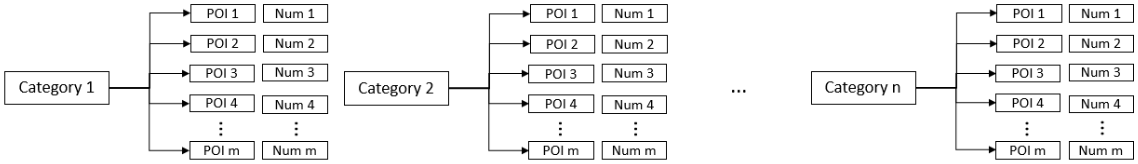


Figure 6-12 Demo of the structure of POI categories

Based on the structure of POI categories, the probability P_{C_i} of user selecting the POI category P_{C_i} can be calculated:

$$P_{C_i} = \frac{N_{C_i}}{\sum_{i=1}^{n} N_{C_i}}$$

Parameter N_{C_i} denotes the number of user's all the historical visited records of the POI category C_i. The n represents in the number of all the POI categories. The calculation formula of N_{C_i} is as below:

$$N_{C_i} = \sum_{j=1}^{m} N_{p_j}$$

In the formula, parameter N_{p_i} denotes the number of all the historical visited records of the POI p_j, and m is the number of all the POI in the POI category C_i.Based on the two formulas above, the probability P_{C_i} of user selecting the POI category P_{C_i} can be represented by the following formula.

$$P_{C_i} = \frac{\sum_{j=1}^{m} N_{p_j}}{\sum_{i=1}^{n} \sum_{j=1}^{m} N_{p_j}}$$

Besides, the probability P_{p_i} of user selecting the POI P_{j} can be calculated with formula below.

$$P_{p_i} = \frac{N_{p_j}}{\sum_{j=1}^{m} N_{p_j}}$$

From the above formulas, the preference pre_j of user over POI P_j can be calculated:

$$pre_{\mathrm{j}} = P_{C_i} \times P_{P_j}$$

After the series of formulas, the preference value of users toward each indoor POI can be obtained.

2. Correlation calculation

In the section, a definition about the correlation values among the target users and his/her friends is given, as well as among POI. The correlation values are usually called similarity. Based on the above section, the preference values of users over each indoor POI have been obtained. For each user, the preference values toward all the indoor POI can be treated as a one-dimensional vector. As shown in the Table 7-3, for the target user A, his/her one-dimensional preference vector towards all the indoor POI is $user_A = \{pre_{AA}, pre_{AB}, pre_{AC}, pre_{AD}\}$.

Table 6-3 Demo of users' preference values over indoor POI

	Indoor POI A	Indoor POI B	Indoor POI C	Indoor POI D
The target user A	pre_{AA}	pre_{AB}	pre_{AC}	pre_{AD}
Friend B	pre_{BA}	pre_{BB}	pre_{BC}	pre_{BD}
Friend C	pre_{CA}	pre_{CB}	pre_{CC}	pre_{CD}

In order to measure the similarity of each two vectors, Cosine similarity formula is adopted to calculation the similarity values. The cosine similarity $cos(u,\ f_i)$ of each two vectors can be expressed by the following formula, where *n* is the number of all indoor POI.

$$\cos(u, f_i) = \frac{\sum_1^n (u \times f_i)}{\sqrt{\sum_1^n u^2} \times \sqrt{\sum_1^n {f_i}^2}}$$

Besides, as the previous speaking, user's friends are divided into the intimate friends and the common friends. The two kinds of friends have different influence degree to the target user. So whether the friends are intimate friends is checked firstly. If the friends are intimate friends, an extra 1 is added to the similarity value. Therefore, the similarity value between the target user and his/her friends can be defined as the following formula.

$$Sim_{users} = \begin{cases} \cos(u, f_i) + 1, if\ f_i\ is\ user's\ intimate\ friend \\ \cos(u, f_i), \qquad\qquad\qquad\qquad\qquad others \end{cases}$$

For the indoor POI, as the Table 7-3, the preference value of all users over an indoor POI can be treated as a one-dimensional vector. For example, the one-dimensional vector of indoor POI A is $POI_A = \{pre_{AA}, pre_{BA}, pre_{CA}\}$. Hence, the similarity of the two POI vectors can also be calculated with cosine similarity formula, as formula presents, where m is the number of all the users.

$$Sim_{POI} = cos(p_i, p_j) = \frac{\sum_1^m (p_i \times p_j)}{\sqrt{\sum_1^m {p_i}^2} \times \sqrt{\sum_1^m {p_j}^2}}$$

3. Neighbour selecting

In the section, the neighbors of user are defined as the friends who have high similarity with the target user. In the last section, it is obtained that the social similarity Sim_{user} between the target user and his/her friends, and the indoor POI similarity Sim_{POI} among indoor POI, shown in the Table 7-4.

Table 6-4 Demo of the social similarity and the indoor POI similarity

	Indoor POI A	Indoor POI B	Indoor POI C	Indoor POI D
User A	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$
Friend B	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$
Friend C	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$	$(\mathrm{Sim}_{\mathrm{user}},\ Sim_{POI})$

After that, ordered by the values obtained by multiply Sim_{user} by Sim_{POI} from small to large, the indoor POI is ranked. Finally, top-N indoor POIs are picked up as the recommended POI to the user. The process is illustrated as Figure 7-13.

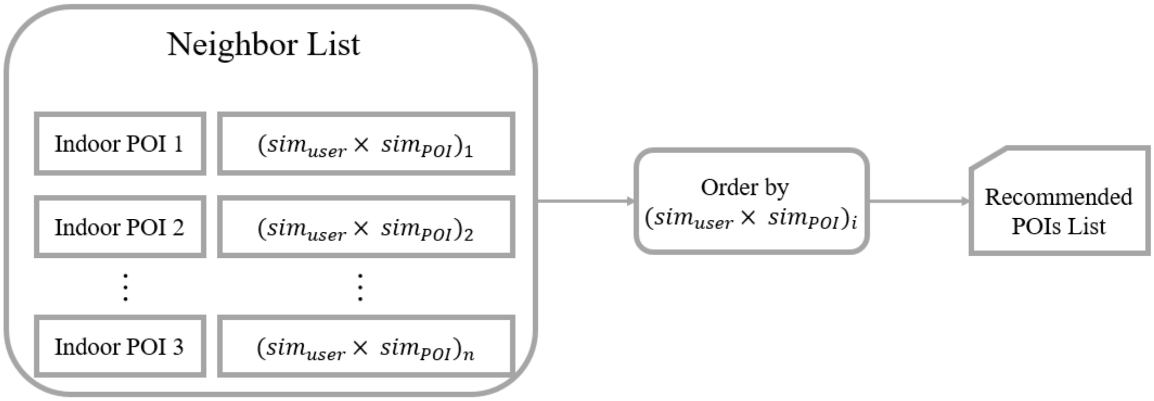

Neighbor List
Indoor POI 1
$(sim_{user} \times sim_{POI})_1$
Indoor POI 2
$(sim_{user} \times sim_{POI})_2$
Indoor POI 3
$(sim_{user} \times sim_{POI})_n$
Order by $(sim_{user} \times sim_{POI})_i$
Recommended POIs List

Figure 6-13 Sketch diagram of selecting neighbours

6.4 Evaluation on Indoor Data Recommendation

The testing requires big social network data and POI data; Weibo data is crawled by WebCollector as my testing data set. Experiments are performed on windows 8.0 of 1.83GHz frequency, 4 CPU, 8GB memory, 1TB hard drive. The experimental data are presented in Table 7−5.

Table 6-5 Experimental data from Sina Weibo

Type of data	Users	Posts	Comments	Likes	POIs
Amount	363,891	34,687,228	36,458,269	17,736,854	8,437,477

The indoor POI dataset is obtained from the teaching building in the University of Pavia, Italy. There are 126 indoor POI in the building, which include several kinds of labs, several classrooms, several study-rooms, several meeting rooms, and several leisure rooms, e.g., coffer bar. Sometimes, some public activities, clubs or lectures are held on this building. The details are presented in Table 7-6.

Table 6-6 Experimental indoor POI dataset from local indoor environment

Data Types	Quantity
The number of total indoor POI	126
The number of textual information of all the POI	528
The number of comments generated by users	721

In the social dataset, Figure 7-14 demonstrates the changing of logP(w|K) with different K values in Gibbs sampling. From the Figure 7-14, it can be observed that the degree of fitting LDA model over social dataset is best, so the performance of generating document of LDA model is strongest at the moment. So the K value is 160. Table 7-7 presents the detail parameters values of LDA model.

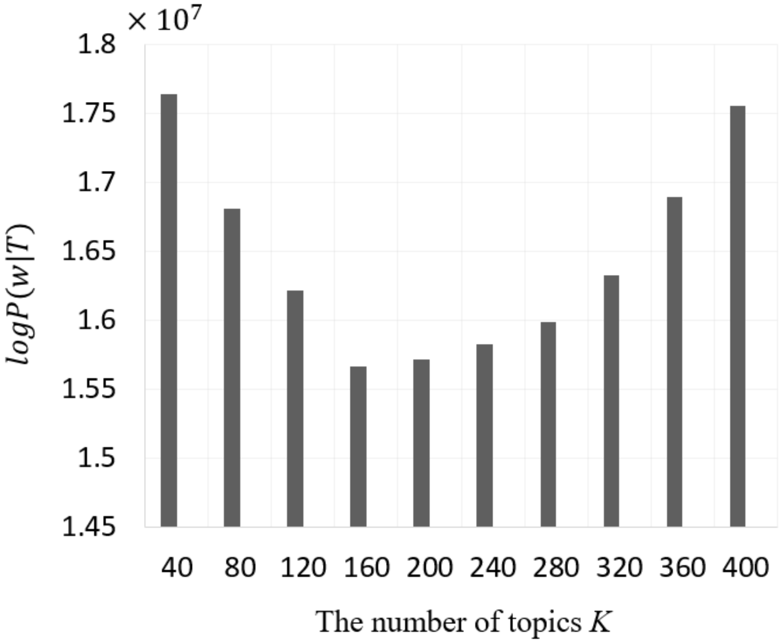


Figure 6-14 K values selecting

Table 6-7 Parameters values of LDA model

α	50/K
β	0.01
The number of topics K	160
The sampling times of Gibbs sampling M	2000
The number of iteration intervals	200

The evaluation index refers to the measurement of algorithm. There are three evaluation indexes: Precision, Recall and F score. Table 7−8 details these concepts. The accuracy rate p means the ratio of the number of correct results to the total numbers, which can be expressed by the following formula.

Table 6-8 The matrix of evaluation indexes

	Positive Result	False Result
Actual Positive	TP	FN
Actual False	FP	TN

$$\mathrm{p} = \frac{TP}{TP + FP}$$

“Recall” means the percent of actual positive in the total correct number. Recall r can be defined as:

$$\mathrm{r} = \frac{TP}{TP + FN}$$

The F value is a comprehensive evaluation of precision and recall, since they are often inversely proportional. The higher F value, the lower recall. The F-value is calculated as:

$$\mathrm{F} = \frac{2pr}{p + r}$$

In addition, the classification algorithms, including social relations classification algorithm and POI classification algorithm, define categories. Each category C_i has its own accuracy p_{ci}, recall r_{ci} and F value.

6.4.1 Evaluation on Social Relations Analysis

The TF - IDF - VSM model is one of the commonly used Subject Clustering Models. The model is mainly composed of VSM model and TF - IDF algorithm. VSM presents the text as a vector V in n-dimensional feature space. TF - IDF algorithm distinguishes different text by the semantics frequency in this text and other text, which can define the characteristics of each text for subject clustering. In this method, TF refers to the frequency of the basic semantic unit occurrence, the greater the frequency, the larger TF value. IDF is the reverse of text frequency (the probability of word occurrence in other texts). The greater IDF (the less the probability of semantic unit occurrence in other texts), which can distinguish from other texts. The TF value of the i_{th} feature is:

$$TF_i = \frac{t_i}{\sum_{j=1}^{n} t_j}$$

Where t_i presents the frequency of the i_{th} characteristic in the eigenvector, and n is the number of all the characteristics. The inverse text frequency IDF_i of the i_{th} characteristic in the eigenvector is:

$$IDF_i = log_2 \frac{\sum_{i=1}^{n} num_i}{num_i}$$

Where num_i presents the number of the i_{th} feature vocabulary occurrences. TF--IDF is calculated as follows:

$$w_i = TF_i \times IDF_i$$

After calculating the TF - IDF value for each text, I cluster them by K-means. I randomly select and label 1000 microblogs (posts) belonging to five subjects, as shown in Table 7−9. Then cluster them using LDA model and TF - IDF - VSM model respectively. LDA model needs to define the parameters firstly, as presented in Table 7−10. I take the maximum subject probability of each text as the classification result.

Table 6-9 Testing data set

Subject	Valentine's Day	Mother’s Day	Classmate Meeting	Colleagues travel	Concert
# of text	138	153	284	252	173

Table 6-10 Parameter definition in LDA model

Parameter	α	β	# of subjects	# of iterations
Number	0.5	0.1	5	500

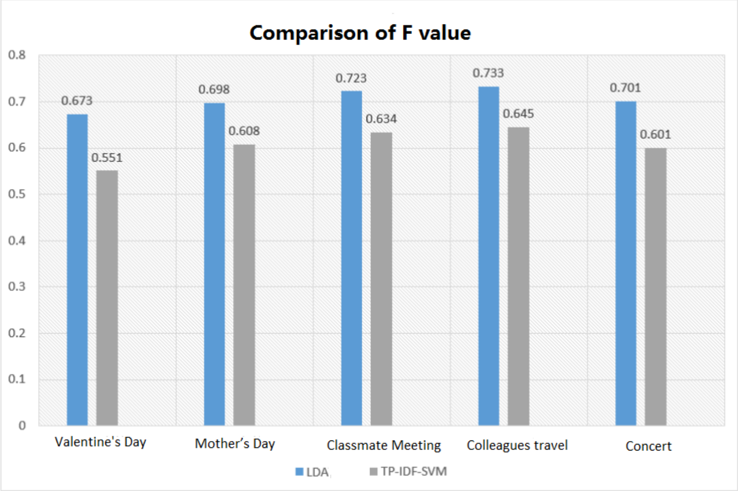


Figure 6-15 The comparison of F value in LDA and TP-IDF-SVM models

The obtained F value of two experimental results are shown in Figure 7-15, the average F value of the LDA model is 0.706, and the one of TF - IDF - VSM model is 0.608. From the comparison results, the F value of LDA model is about 16.1% higher than the one of TF – IDF - VSM model. Therefore, LDA model is more suitable for my case study.

6.4.2 Evaluation on POI-based Relations Mining Algorithm

This algorithm aims at the user's intimacy, such as relatives, lovers, etc. In order to verify the algorithm, I invite 100 Weibo users and ask them to classify their intimate friends. Then I put the labeled ones as the sample data of this experiment. The F-value of the algorithm is presented in Figure 7-16, where every five users as a basic unit, and y-axis is the average F value.

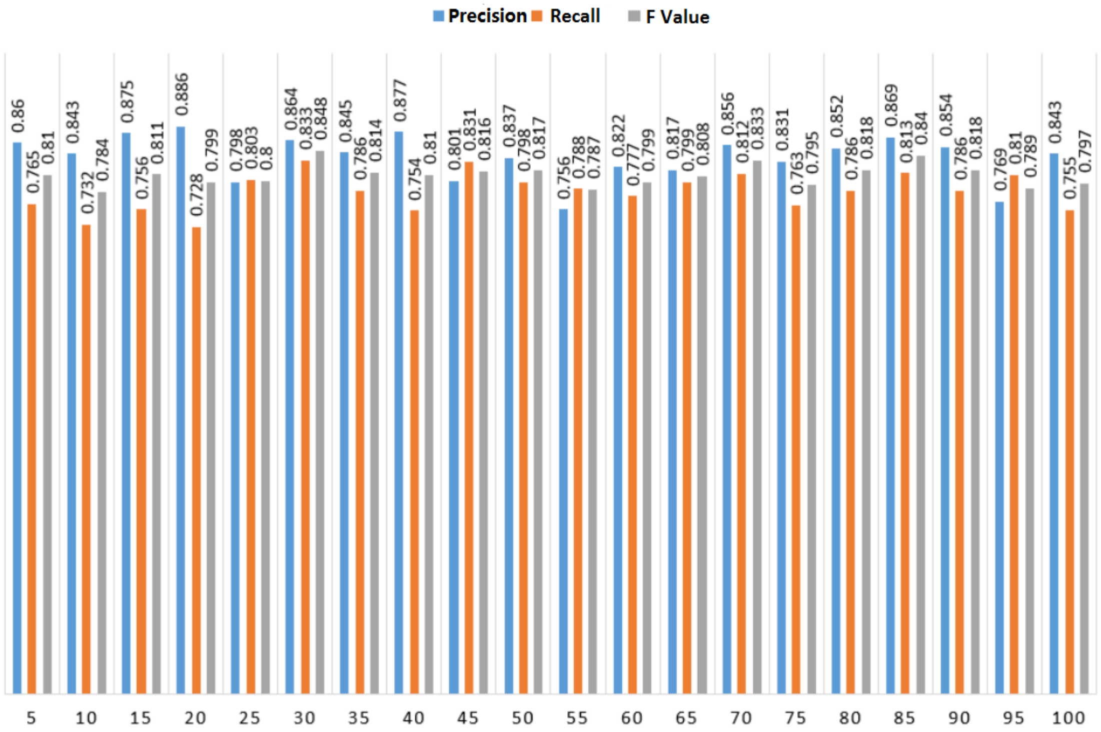


Figure 6-16 The graph of POI-based relations mining algorithm evaluation

As figure presented, the average accuracy rate of the algorithm is 0.838, the average recall rate is 0.784, and average F value is about 0.810. Sometimes, the text may not contain only one '@' symbol. Then, I improve the algorithm, that is, when the user forwards the other text, if the only one '@' symbol is in front of the '//' symbol, it is also considered as text.

After optimizing, the average accuracy of the algorithm is 0.870, the average recall rate is 0.810, the average F value is 0.838 (Figure 7-17). Compared with the pre-optimized algorithm, the average accuracy increases by 3.8%, the average recall rate increases by 3.2%, while the average F value increases by 3.4%.

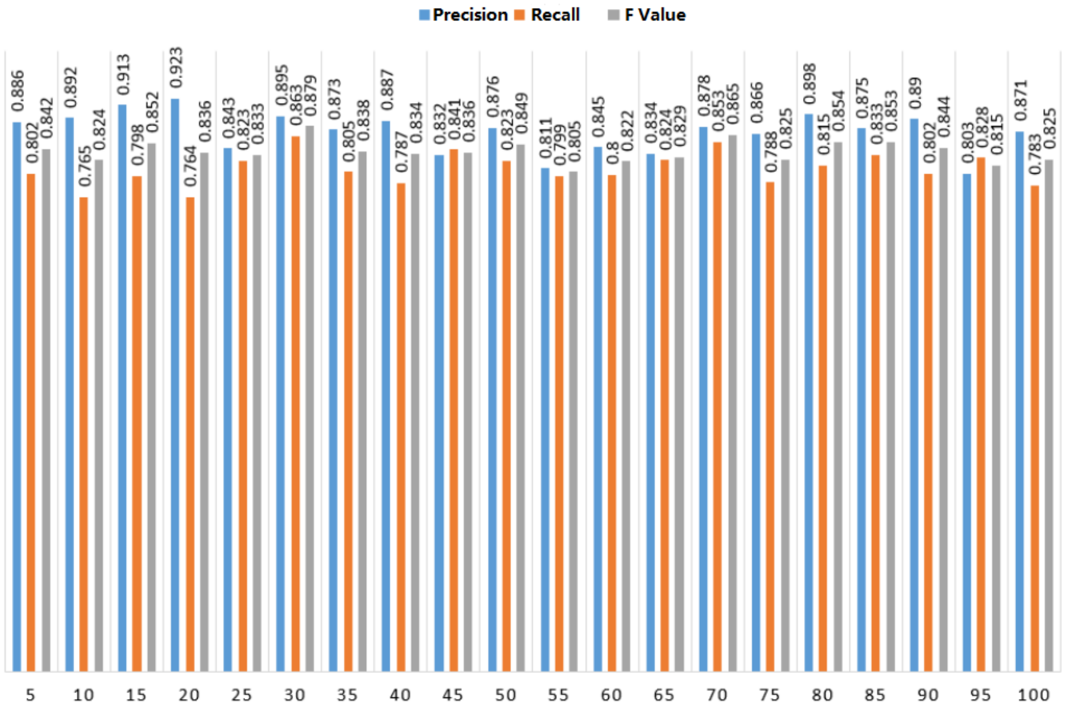


Figure 6-17 Evaluation of optimized POI-based relations mining algorithm

6.4.3 Evaluation on Subject-based Relations Mining Algorithm

In order to verify the algorithm, I extract more than 80% of the user complete information. The number of users occupies only 17% of the total users, that is, 61,861 users. Then, I select and classify 100 users as the test sample among them, as university students, colleagues, girlfriends/ buddies, fellows and relatives. The F value after testing is presented in Figure 7-18. From the figure, I can get the average F of each class is: 0.790, 0.789, 0.794, 0.784, 0.773, so the average F value is 0.786.

After analysis, I found that, some users and their friends do not have many interactions, it will lead to the reduced accuracy. I later optimized the algorithm and improve its performance. I pre-processed the user friends' data, and removed some of the friends with little interactions. After optimizing, I got the average F value of each algorithm is 0.822, 0.811, 0.814, 0.812 and 0.803, and the average F value is 0.812, as presented in Figure 7-19.

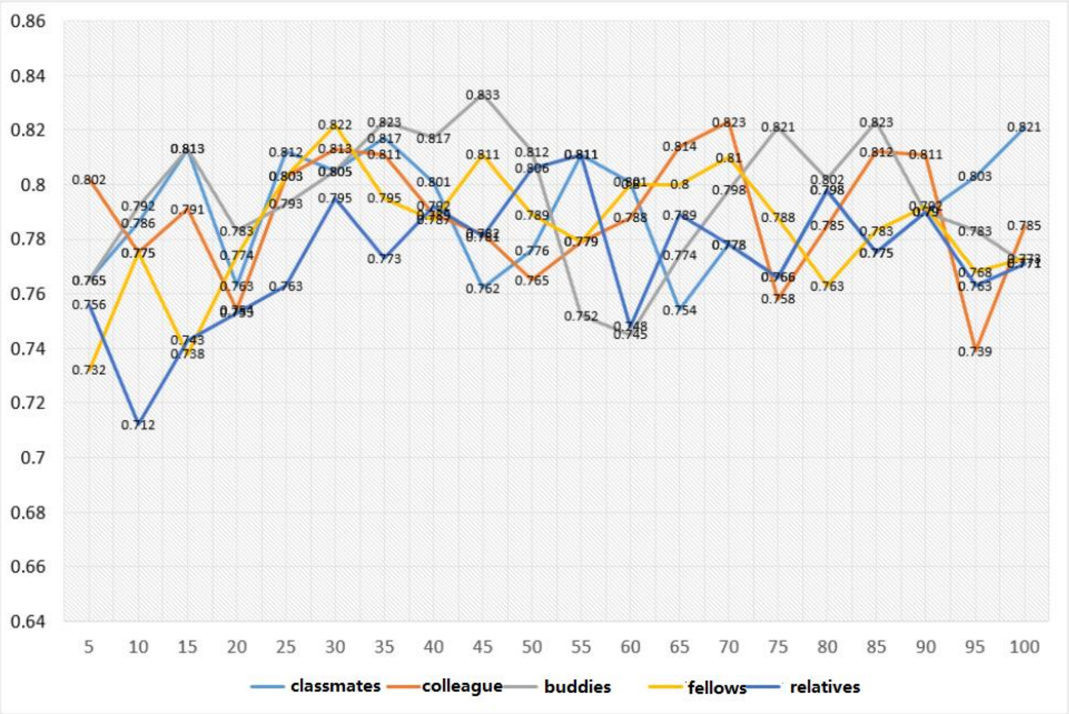


Figure 6-18 F value of subject-based relations mining algorithm

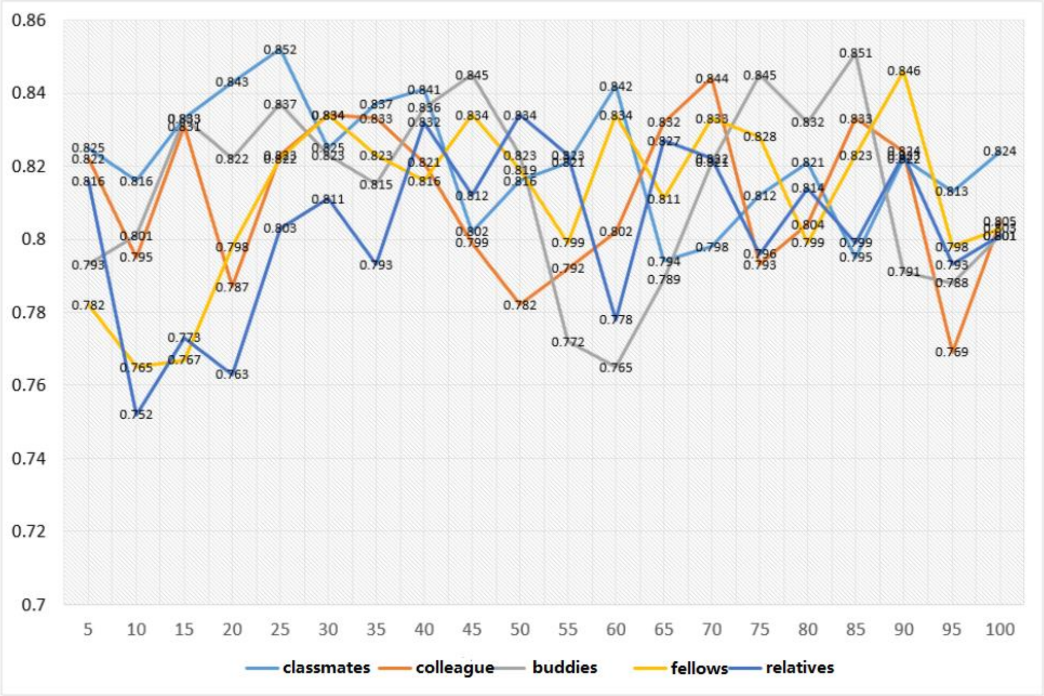


Figure 6-19 F value of optimized subject-based relations mining algorithm

6.4.4 Evaluation on Personal POI Recommendation Algorithm

I select and label 1200 POIs, which is divided into six categories, and then I get the F value of each class of POIs, as presented in Table 7−11. According to the table, I can conclude that the average F value is 0.691.

Actually, the performance is not good enough. Because I ignore some kinds of connection between a single POI and its surrounding POIs. For instance, a shopping mall contains a large number of clothing stores (POI) and food stores (POI). According to the above problems, I first select the POIs of all users, DBSCAN to cluster all the POIs by latitude and longitude, and get different areas of POIs. After optimizing, and the F values for each category of interest are shown in Table 7−12. The average F value of the algorithm is 0.817, which is nearly 18.23% higher than that before optimization.

Table 6-11 POI testing data set

POI	Shopping Mall	School	Food Court	Attractions	Government	Offices
No.	232	161	298	173	159	177
F value	0.653	0.667	0.696	0.708	0.732	0.691

Table 6-12 Optimized POI testing data set

POI	Shopping Mall	School	Food Court	Attractions	Government	Offices
F value	0.818	0.834	0.802	0.838	0.814	0.795

The function of the personal POI recommendation in indoor navigation enhances the user experience by providing the user with the potential POIs, giving the user high quality service.

Since the number of users in this case study, indoor navigation of University of Pavia, is not big enough to build their own social network. Therefore, the system is supported through Sina Weibo to log in. The social users can be classified as presented in Figure 7-20.

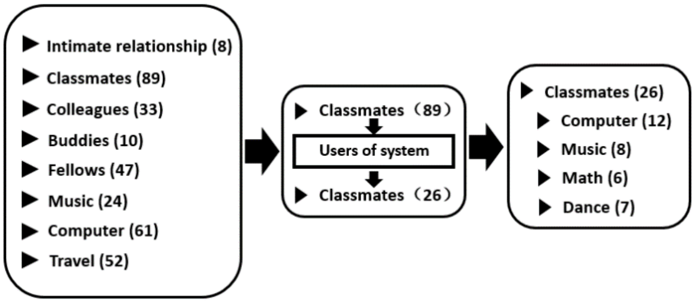


Figure 6-20 The diagram of user friend classification and selection

In addition, I classify the POIs into 8 categories: classroom, leisure room, study room, community activities group, conference room, laboratory, professor office and administrative office. After obtaining the categories of user friends and POIs, I match them by this friend and POI relationship mapping algorithm, as presented in Figure 7-21.

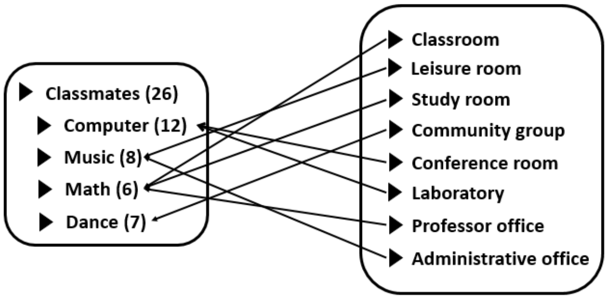
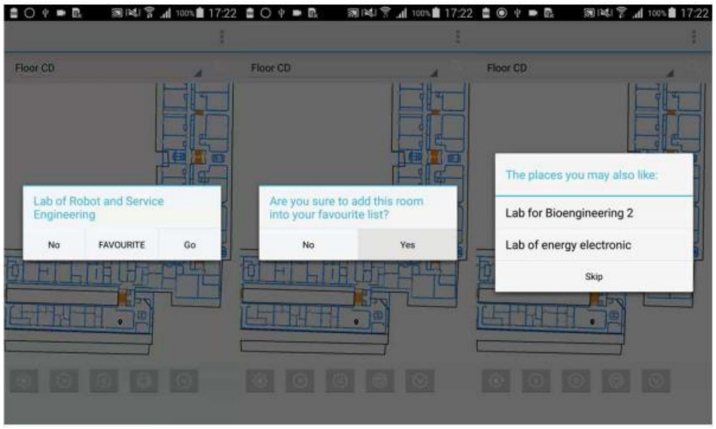


Figure 6-21 The mapping of friend classification and POI classification

This POI recommendation algorithm runs based on the stored POI, which is triggered when adding POI to favorite folder. System will return a recommended POI list, which is shown in Figure 7-22. Users can click on the list of POIs to view their details and can be navigated to the destination.



Lab of Robot and Service Engineering

Are you sure to add this room into your favourite list?

The places you may also like:

Lab for Bioengineering 2

Lab of energy electronic

Figure 6-22 Diagram of adding POI list to the favourite folder

As it is said before, the proposed personalized POI recommendation algorithm combining collaborative filtering recommendation method achieves the POI recommendation function. The validation of the algorithm is the average adoption rate.

The adoption rate a_i is the ratio of adding history number Num_{rp} to the total recommended historical POIs Num_p. The average adoption rate avg_a is the sum of adopting users divided by total users N, as the formula presented.

$$a_i = \frac{Num_{rp}}{Num_p}$$

$$avg_a = \frac{\sum_{i=1}^{N} a_i}{N} = \frac{\sum_{i=1}^{N} \frac{Num_{rp}}{Num_p}}{N}$$

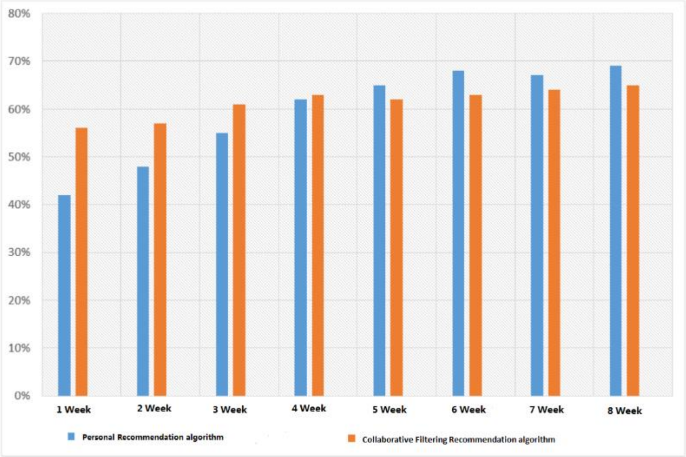


Figure 6-23 Comparison of recommendation algorithm adoption rates

After two months testing, the adoption rate for each week is presented in Figure 7-23. We can see that in the early stage, collaborative filtering recommendation algorithm is better than the proposed algorithm proposed because of less amount of users, and the less significance of the social relations classification. However, with the system putting into use longer, the more users, more user's behavior data, and the better proposed algorithm. In this case, the adoption rate of the recommended algorithm is high. In addition, with the continuous development of indoor navigation system, I will not only confine to a comprehensive teaching building in a campus, the POI recommendation data analysis will play an important role in the indoor environment expansion as well.

6.5 Summary

In this chapter, my proposed personal POI recommendation can help users quickly in a large and responsible indoor environment to find their real needs. The recommendation algorithm by analyzing intricate social relations is designed to improve the accuracy and enhance the user experience. The main work of this chapter is as follows:

1) I study the research status of social recommendation system. Also, the shortcomings and improvement points of existing research models are researched and summarized.
2) Based on the user's complex social relations, I design a social relations analysis model that is used to mine and classify the social relations.
3) A classification algorithm of POIs and a mapping algorithm between POIs and friends are proposed.
4) An integrated personal recommendation approach is proposed. It first uses the social analysis model to classify the social relations of the users, then classifies the POIs, and mapping two of them. Finally, the improved collaborative filtering recommendation algorithm is used to generate a list of recommendations to the user to find his/her own potential POI.
5) The evaluation proves that this method has a good performance. In addition, POI classification algorithm can be used in other scenarios.

Chapter 7 Conclusion and Future Works

Indoor navigation is an innovative extension of the classic outdoor navigation to multi-floor complex building navigation. The system that I have illustrated plays an important role of meeting users' indoor services demand, improving users' indoor experiences and reducing the stress of searching positions and paths. An object-oriented thought and a comfort design of indoor navigation system bring various benefits to a smart city. Finally, a user validation testing result of Indoor Mobility is presented in Table 7-1 that determines if the basic requirements of each navigation phase are met. Table 7-1 defines a set of basic activities conducted in the full phases of indoor navigation services and presents an acceptable result that suits real scenarios.

Table 7-1 A user validation testing of Indoor Mobility

No.	Usability Testing Task	Actual Result	Pass/Fail	Comments
1.	Step into a testing building, look into the 2D and 3D layout of this building.	Sensitive operation on 2D and 3D vivid indoor scenarios.	Pass	Vivid rendering and high performance of map operation (e.g., zoom in, zoom out, rotation, etc.) are important.
2.	Random positioning on the map	Precisely locate user (1.6m) on the map	Pass	
3.	Continuously positioning while user walking	Precisely locate with rarely delay.	Pass	Keep on precisely positioning while walking is the foundation of efficient navigation.
4.	Normal navigation on 2D map	Plan the shortest path to the user.	Pass	Effective path planning and positioning support high quality navigations.
5.	AR navigation and voice alert testing to the destination	Present the correct (voice) direction.	Pass	Improve the effectiveness and efficiency of navigation
6.	Predict the potential POI by network	90% (45/50) correctly predict the user potential POI. successfully on the first attempt.	Pass	

In particular, I summarize the full phases of indoor navigation services, as I have underlined:

- Indoor Mapping: namely 2D and 3D Indoor Mapping, which provides a realistic virtual indoor environment for navigation. This indoor mapping approach provides a solid foundation for indoor management and navigation services. Also, indoor SVG map design and rapid 2D-to-3D conversion can be further extended. In my opinion, a richer indoor visualization will be further addressed, in the aspects of, e.g., enrich rendering materials, more indoor content description, etc.

- Indoor Positioning: two solutions namely, RSML and CVFL. RSML fuses multiple sensor data, Magnetic fields, Wi-Fi communication, Accelerator, Dead reckoning etc. CVFL uses CNN to classify the magnetic fingerprints for positioning. Various solutions both provide accurate positioning results and flexible indoor navigation. I compare these two solutions, CVFL achieves a higher positioning accuracy than RSML's, while, the computational complexity of CVFL is higher than RSML's with 1.864 m. The research on indoor positioning technology is complete. But for future work, two aspects can be further considered: 1) Combination of these two solutions not only improves the positioning accuracy but also reduces computational complexity; 2) Enrich the training dataset. Collect more magnetic fingerprints in multi-buildings.
- Indoor Path Planning: the proposed ACO path algorithm can detect the boundary of building, and plan short, safe and convenient path for multiple stakeholders. The ACO algorithm still has room to improve, e.g., 1) When searching the farthest point that is not touching the wall, expand the range of point analysis; 2) Increase the ant crawling distance (one step three pixels instead of one step one pixel) to accelerate the processing; 3) Increase appropriate number of pixels against and thicken the wall so that no more points against the wall appear; 4) Seamless integration of indoor and outdoor for further development.
- Indoor En-route Assistant: the AR and voice alert for VIP navigation solutions are effective. AR navigation facilities the common users (citizens) for way finding, while, VIP navigation can facilities VIPs. The overall framework can be the conceptual skeleton of an eco-system of services for indoor navigation, alike e015 digital ecosystem [143], which Regione Lombardia released for Milano EXPO 2015 to provide a set of services on mobility and smart city. So far, there is still space for improvement. I see several directions of future work: 1) Enhance spatial reasoning and analysis between the key points for a step-by-step navigation; 2) Extend specific interface for blind people, e.g., wearable systems, braille, etc. Specific advanced value propositions can be developed for blind people, wheelchair people, and, on the other side, for robots; 3) Estimate the distance of pedestrians according to their heights for blind users; 4) Detect more obstacles for VIPs, not limited with pedestrians, such as door detection, stairs detection, elevator detection, etc.
- Indoor Data Analysis: I analyze indoor POIs and extract social subjects and relationships from social network. I recommend POIs based on social behaviors as a navigation prompt. The testing results validate a good performance of the proposed POI recommendation algorithm. The future work can be summarized as the following aspects: 1) Extend the social network or build an independent social network in the indoor navigation system; 2) Extend the indoor scenes, e.g., large shopping malls, international airports, etc.

References

[1] Kaplan, Elliott, and Christopher Hegarty, eds. Understanding GPS: principles and applications. Artech house, 2005.

[2] Vanclooster, Ann, and Philippe Maeyer. "Combining indoor and outdoor navigation: the current approach of route planners." Advances in Location-Based Services (2012): 283-303.

[3] "Year to date Passenger Traffic". ACI. 2015-06-22. Retrieved 2015-06-23.

[4] Study of the future development of the national airport network - One Works, Nomisma, Kpmg (2010).

[5] “Investigation report Xidan commercial street traffic” (2004).

[6] Clara Guibourg, Chris Parmenter. “City A.M.’s analysis of figures from the London Datastore” (2015)

[7] Carlson, Laura A., et al. "Getting lost in buildings." Current Directions in Psychological Science 19.5 (2010): 284-289.

[8] Subbu, Kalyan Pathapati, Brandon Gozick, and Ram Dantu. "LocateMe: Magnetic-fields-based Indoor localization using smartphones." ACM Transactions on Intelligent Systems and Technology (TIST) 4.4 (2013): 73.

[9] Zhang, Chuanrong, Tian Zhao, and Weidong Li. "Geospatial Data Interoperability, Geography Markup Language (GML), Scalable Vector Graphics (SVG), and Geospatial Web Services." Geospatial Semantic Web. Springer International Publishing, 2015. 1-33.

[10] Wang, Junchen, et al. "Augmented reality navigation with automatic marker-free image registration using 3-D image overlay for dental surgery." IEEE transactions on Biomedical Engineering 61.4 (2014): 1295-1304.

[11] Lv, Zhihan, et al. "3D visual analysis of seabed on smartphone." arXiv Browne, Aidan F., and David Vutetakis. "Localization of Autonomous Mobile Ground Vehicles in a Sterile Environment: A Survey." (2015).

[12] Williams, Brian, and Ian Reid. "On combining visual SLAM and visual odometry." Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE, 2010.

[13] Parsley, Martin P., and Simon J. Julier. "Exploiting prior information in GraphSLAM." Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.

[14] Milstein, Adam. "Occupancy grid maps for localization and mapping." Motion Planning. InTech, 2008.

[15] Georgiou, Christina, Sean Anderson, and Tony Dodd. "Constructing contextual SLAM priors using architectural drawings." Automation, Robotics and Applications (ICARA), 2015 6th International Conference on. IEEE, 2015.

[16] Lee, Jooyeon, et al. "Peripersonal Space in Virtual Reality: Navigating 3D Space with Different Perspectives." Proceedings of the 29th Annual Symposium on User Interface Software and Technology. ACM, 2016.

[17] Ren, Gang, et al. "3D freehand gestural navigation for interactive public displays." IEEE computer graphics and applications 33.2 (2013): 47-55.

[18] Indraprastha, Aswin, and Michihiko Shinozaki. "Constructing virtual urban environment using game technology." 26th eCAADe Conference: Architecture in Computro Antwerpen. 2008.

[19] Konrad, Janusz, Meng Wang, and Prakash Ishwar. "2d-to-3d image conversion by learning depth from examples." 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, 2012.

[20] Harman, Philip V., et al. "Rapid 2D-to-3D conversion." Electronic Imaging 2002. International Society for Optics and Photonics, 2002.

[21] Tam, Wa James, and Liang Zhang. "3D-TV content generation: 2D-to-3D conversion." 2006 IEEE International Conference on Multimedia and Expo. IEEE, 2006.

[22] Zhang, Chuanrong, Tian Zhao, and Weidong Li. "Geospatial Data Interoperability, Geography Markup Language (GML), Scalable Vector Graphics (SVG), and Geospatial Web Services." Geospatial Semantic Web. Springer International Publishing, 2015. 1-33.

[23] Collins, Mark J. "Scalable Vector Graphics." Pro HTML5 with Visual Studio 2015. Apress, 2015. 209-235.

[24] Li, M., et al. "Comparative Study on Visualizing Vector Graphics in WebGIS Applications with SVG and Flash Technologies." (2015).

[25] Cristie, Verina, et al. "CityHeat: visualizing cellular automata-based traffic heat in Unity3D." SIGGRAPH Asia 2015 Visualization in High Performance Computing. ACM, 2015.

[26] Merlo, Alessandro, et al. "3D model visualization enhancements in real-time game engines." J. Boehm, F. Remondino, T. Kersten, T. Fuse, D. Gonzalez-Aguile (a cura di), 3D-ARCH (2013): 181-188.

[27] Dörrie, Jan Wilken, and Michael Kohlhase. "OpenMathMap: Accessing Math via Interactive Maps." Subjects and Issues in Electronic Publishing (2013): 81.

[28] Goh, Kim Nee, Siti Rohkmah Mohd Shukri, and Rofans Beleam Hilisebua Manao. "Automatic Assessment for Engineering Drawing." International Visual Informatics Conference. Springer International Publishing, 2013.

[29] Jackson, Wallace. "Digital Painting Software: Corel Painter and Inkscape." Digital Painting Techniques. Apress, 2015. 1-11.

[30] ZHANG, Yin. "School of Art Design, Guangdong Industry Technical College;; Research on modeling method of fading surfaces in product design based on Rhinoceros [J]." Journal of Machine Design 2 (2014).

[31] Lin, Lin, Zhihui Tian, and Shan Zhao. "Construction of campus 3D scene based on 3dsMax and Unity3D." Hydraulic Engineering III: Proceedings of the 3rd Technical Conference on Hydraulic Engineering (CHE 2014), Hong Kong, 13-14 December 2014. CRC Press, 2014.

[32] De Araùjo, Bruno R., Géry Casiez, and Joaquim A. Jorge. "Mockup builder: direct 3D modeling on and above the surface in a continuous interaction space." Proceedings of Graphics Interface 2012. Canadian Information Processing Society, 2012.

[33] Batra, Vineet, et al. "Accelerating vector graphics rendering using the graphics hardware pipeline." ACM Transactions on Graphics (TOG) 34.4 (2015): 146.

[34] Li, Binghao, et al. "How feasible is the use of magnetic field alone for indoor positioning?." Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on. IEEE, 2012.

[35] Kassim, A. M., et al. "Indoor Navigation System based on Passive RFID Transponder with Digital Compass for Visually Impaired People." INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS 7.2 (2016): 604-611.

[36] Kulyukin, Vladimir, et al. "RFID in robot-assisted indoor navigation for the visually impaired." Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. Vol. 2. IEEE, 2004.

[37] Möller, Andreas, et al. "A mobile indoor navigation system interface adapted to vision-based localization." Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia. ACM, 2012.

[38] Fang, Yeqing, et al. "Application of an improved K nearest neighbor algorithm in WiFi indoor positioning." China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III. Springer Berlin Heidelberg, 2015.

[39] Bekkelien, Anja, Michel Deriaz, and Stéphane Marchand-Maillet. "Bluetooth indoor positioning." Master's thesis, University of Geneva (2012).

[40] Li, Binghao, et al. "How feasible is the use of magnetic field alone for indoor positioning?." Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on. IEEE, 2012.

[41] Pham, Van-Tang, et al. "Thermal Stability of Magnetic Compass Sensor for High Accuracy Positioning Applications." Sensors & Transducers 195.12 (2015): 1.

[42] Chung, Jaewoo, et al. "Indoor location sensing using geo-magnetism." Proceedings of the 9th international conference on Mobile systems, applications, and services. ACM, 2011.

[43] Haverinen, Janne, and Anssi Kemppainen. "A global self-localization technique utilizing local anomalies of the ambient magnetic field." Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. IEEE, 2009.

[44] Pratama, Azkario Rizky, and Risanuri Hidayat. "Smartphone-based pedestrian dead reckoning as an indoor positioning system." System Engineering and Technology (ICSET), 2012 International Conference on. IEEE, 2012.

[45] Wroble, Kyle. "Performance Analysis of Magnetic Indoor Local Positioning System." (2015).

[46] Pratama, Azkario Rizky, and Risanuri Hidayat. "Smartphone-based pedestrian dead reckoning as an indoor positioning system." System Engineering and Technology (ICSET), 2012 International Conference on. IEEE, 2012.

[47] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature521.7553 (2015): 436-444.

[48] Lee, Daewon, and Jaewook Lee. "Domain described support vector classifier for multi-classification problems." Pattern Recognition 40.1 (2007): 41-51.

[49] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[50] Ciresan, Dan C., et al. "Flexible, high performance convolutional neural networks for image classification." IJCAI Proceedings-International Joint Conference on Artificial Intelligence. Vol. 22. No. 1. 2011.

[51] Pasku, Valter, et al. "Magnetic field analysis for distance measurement in 3D positioning applications." Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International. IEEE, 2016.

[52] Wu, Falin, et al. "A robust indoor positioning system based on encoded magnetic field and low-cost IMU." 2016 IEEE/ION Position, Location and Navigation Symposium (PLANS). IEEE, 2016.

[53] Barsocchi, Paolo, et al. "A multisource and multivariate dataset for indoor localization methods based on WLAN and geo-magnetic field fingerprinting." Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on. IEEE, 2016.

[54] Montoliu, R., J. Torres-Sospedra, and O. Belmonte. "Magnetic field based Indoor positioning using the Bag of Words paradigm." Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on. IEEE, 2016.

[55] Du, Yichen, Tughrul Arslan, and Arief Juri. "Camera-aided region-based magnetic field indoor positioning." Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on. IEEE 2016.

[56] Robertson, Patrick, et al. "Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments." Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on. IEEE, 2013.

[57] Cadena, Cesar, et al. "Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age." arXiv preprint arXiv:1606.05830 (2016).

[58] Liu, Hung-Huan, and Yu-Non Yang. "WiFi-based indoor positioning for multi-floor environment." TENCON 2011-2011 IEEE Region 10 Conference. IEEE, 2011.

[59] Wang, Xuyu, et al. "DeepFi: Deep learning for indoor fingerprinting using channel state information." Wireless Communications and Networking Conference (WCNC), 2015 IEEE. IEEE, 2015.

[60] Wang, Xuyu, Lingjun Gao, and Shiwen Mao. "CSI phase fingerprinting for indoor localization with a deep learning approach." IEEE Internet of Things Journal 3.6 (2016): 1113-1123.

[61] Kendall, Alex, Matthew Grimes, and Roberto Cipolla. "Posenet: A convolutional network for real-time 6-dof camera relocalization." Proceedings of the IEEE international conference on computer vision. 2015.

[62] Kendall, Alex, and Roberto Cipolla. "Modelling uncertainty in deep learning for camera relocalization." Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE, 2016.

[63] Zhu, Yuke, et al. "Target-driven visual navigation in indoor scenes using deep reinforcement learning." arXiv preprint arXiv:1609.05143 (2016).

[64] Zhang, Wei, et al. "Deep neural networks for wireless localization in indoor and outdoor environments." Neurocomputing 194 (2016): 279-287.

[65] Herbst, Edward, and Frank Schorfheide. "Sequential Monte Carlo sampling for DSGE models." Journal of Applied Econometrics 29.7 (2014): 1073-1098.

[66] Douc, Randal, and Olivier Cappé. "Comparison of resampling schemes for particle filtering." ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.. IEEE, 2005.

[67] Pascolini, Donatella, and Silvio Paolo Mariotti. "Global estimates of visual impairment: 2010." British Journal of Ophthalmology (2011): bjophthalmol-2011.

[68] Koo, Jahyoung, and Hojung Cha. "Localizing WiFi access points using signal strength." IEEE Communications letters 15.2 (2011): 187-189.

[69] Park, DoWoo, and Joon Goo Park. "An enhanced ranging scheme using WiFi RSSI measurements for ubiquitous location." Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on. IEEE, 2011.

[70] Bose, Atreyi, and Chuan Heng Foh. "A practical path loss model for indoor WiFi positioning enhancement." Information, Communications & Signal Processing, 2007 6th International Conference on. IEEE, 2007.

[71] Niu, Xiao-Xiao, and Ching Y. Suen. "A novel hybrid CNN–SVM classifier for recognizing handwritten digits." Pattern Recognition 45.4 (2012): 1318-1325.

[72] Liu, Tianyi, et al. "Implementation of training convolutional neural networks." arXiv preprint arXiv:1506.01195 (2015).

[73] Abdel-Hamid, Ossama, et al. "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition." Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012.

[74] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[75] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[76] Zeiler, Matthew D., et al. "On rectified linear units for speech processing." Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.

[77] Sainath, Tara N., et al. "Convolutional, long short-term memory, fully connected deep neural networks." Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, 2015.

[78] Ting, Ching-Jung, and Chia-Ho Chen. "A multiple ant colony optimization algorithm for the capacitated location routing problem." International Journal of Production Economics 141.1 (2013): 34-44.

[79] Laarabi, Mohamed Haitam, et al. "Real-timefastest path algorithm using bidirectional point-to-point search on a Fuzzy Time-Dependent transportation network." Advanced Logistics and Transport (ICALT), 2014 International Conference on. IEEE, 2014.

[80] Seet, Boon-Chong, et al. "A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications." Networking 2004. Springer Berlin Heidelberg, 2004.

[81] Höllerer, Tobias, et al. "Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system." Computers & Graphics 23.6 (1999): 779-785.

[82] Imai, Tatsuya, and Akihiro Kishimoto. "A novel technique for avoiding plateaus of greedy best-first search in satisficing planning." Fourth Annual Symposium on Combinatorial Search. 2011.

[83] Sapin, Emmanuel, Edward Keedwell, and Timothy Frayling. "An Ant Colony Optimization and Tabu List Approach to the Detection of Gene-Gene Interactions in Genome-Wide Association Studies [Research Frontier]."Computational Intelligence Magazine, IEEE 10.4 (2015): 54-65.

[84] Liu, Kaixu, et al. "Multi-floor indoor navigation with geomagnetic field positioning and ant colony optimization algorithm." Service-Oriented System Engineering (SOSE), 2016 IEEE Symposium on. IEEE, 2016.

[85] Dorigo, Marco, and Thomas Stützle. "Ant colony optimization: overview and recent advances." Handbook of metaheuristics. Springer US, 2010. 227-263.

[86] Bao J, Zheng Y, Wilkie D, et al. Recommendations in location-based social networks: a survey. Geoinformatica, 2015, 19(3):525–565

[87] Billinghurst, Mark, Tham Piumsomboon, and Huidong Bai. "Hands in space: Gesture interaction with augmented-reality interfaces." IEEE computer graphics and applications 1 (2014): 77-80.

[88] Murakami, Kazuki, et al. "Poster: A wearable augmented reality system with haptic feedback and its performance in virtual assembly tasks." 3D User Interfaces (3DUI), 2013 IEEE Symposium on. IEEE, 2013.

[89] McMahon, Don D., et al. "Effects of Digital Navigation Aids on Adults With Intellectual Disabilities Comparison of Paper Map, Google Maps, and Augmented Reality." Journal of Special Education Technology 30.3 (2015): 157-165.

[90] Möller, Andreas, et al. "A mobile indoor navigation system interface adapted to vision-based localization." Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia. ACM, 2012.

[91] Dietrich, Arne, Hauke Schmidt, and Jean-Pierre Hathout. "Orientation and navigation for a mobile device using inertial sensors." U.S. Patent No. 6,975,959. 13 Dec. 2005.

[92] Leporini, Barbara, Maria Claudia Buzzi, and Marina Buzzi. "Interacting with mobile devices via VoiceOver: usability and accessibility issues." Proceedings of the 24th Australian Computer-Human Interaction Conference. ACM, 2012.

[93] Smailagic, Asim, et al. "Metronaut: A wearable computer with sensing and global communication capabilities." Personal Technologies 1.4 (1997): 260-267.

[94] Höllerer, Tobias, et al. "Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system." Computers & Graphics 23.6 (1999): 779-785.

[95] Papagiannakis, George, Gurminder Singh, and Nadia Magnenat ‐ Thalmann. "A survey of mobile and wireless technologies for augmented reality systems."Computer Animation and Virtual Worlds 19.1 (2008): 3-22.

[96] Fang, Yeqing, et al. "Application of an improved K nearest neighbor algorithm in WiFi indoor positioning." China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III. Springer Berlin Heidelberg, 2015.

[97] Möller, Andreas, et al. "A mobile indoor navigation system interface adapted to vision-based localization." Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia. ACM, 2012.

[98] Deng, Huan, et al. "Magnified augmented reality 3D display based on integral imaging." Optik-International Journal for Light and Electron Optics 127.10 (2016): 4250-4253.

[99] Kim, JongBae, and HeeSung Jun. "Vision-based location positioning using augmented reality for indoor navigation." Consumer Electronics, IEEE Transactions on 54.3 (2008): 954-962.

[100] Oleksy, Tomasz, and Anna Wnuk. "Augmented places: An impact of embodied historical experience on attitudes towards places." Computers in Human Behavior 57 (2016): 11-16.

[101] Walther-Franks, Benjamin, and Rainer Malaka. "Evaluation of an augmented photograph-based pedestrian navigation system." Smart Graphics. Springer Berlin Heidelberg, 2008.

[102] Iwaneczko, Paweł, Karol Jędrasiak, and Aleksander Nawrat. "Augmented Reality in UAVs Applications." Innovative Simulation Systems. Springer International Publishing, 2016. 77-86.

[103] Russell, Spencer, Gershon Dublon, and Joseph A. Paradiso. "HearThere: Networked Sensory Prosthetics through Auditory Augmented Reality" Proceedings of the 7th Augmented Human International Conference 2016. ACM, 2016.

[104] Ali, Hamid M., and Zahraa T. Noori. "Indoor Way Finder Navigation System Using Smartphone." (2016).

[105] Leppäkoski, Helena, Jussi Collin, and Jarmo Takala. "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals." Journal of Signal Processing Systems 71.3 (2013): 287-296.

[106] Schall, Gerhard, Alessandro Mulloni, and Gerhard Reitmayr. "North-centred orientation tracking on mobile phones." Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on. IEEE, 2010.

[107] Zhang, Yanshun, et al. "Locating method of geomagnetic/inertial integrated navigation system by forecasting the geomagnetic matching initial value."Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese. IEEE, 2014

[108] Saitis, Charalampos, and Kyriaki Kalimeri. "Identifying urban mobility challenges for the visually impaired with mobile monitoring of multimodal biosignals." International Conference on Universal Access in Human-Computer Interaction. Springer International Publishing, 2016.

[109] Crudden, Adele, Jennifer L. Cmar, and Michele C. McDonnall. "Stress Associated with Trans-portation: A Survey of Persons with Visual Impairments." Journal of Visual Impairment & Blindness 111.3 (2017).

[110] Hesch, Joel A., and Stergios I. Roumeliotis. "An indoor localization aid for the visually impaired." Robotics and Automation, 2007 IEEE International Conference on. IEEE, 2007.

[111] Serrão, M., et al. "Computer vision and GIS for the navigation of blind persons in buildings." Universal Access in the Information Society 14.1 (2015): 67-80.

[112] Chen, Hsuan-Eng, et al. "BlindNavi: a navigation app for the visually impaired smartphone user." Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems. ACM, 2015.

[113] Apostolopoulos, Ilias, et al. "Integrated online localization and navigation for people with visual impairments using smart phones." ACM Transactions on Interactive Intelligent Systems (TiiS) 3.4 (2014): 21.

[114] Joseph, Samleo L., et al. "Visual semantic parameterization-To enhance blind user perception for indoor navigation." Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on. IEEE, 2013.

[115] Bigham, Jeffrey P., Richard E. Ladner, and Yevgen Borodin. "The design of human-powered access technology." The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility. ACM, 2011.

[116] Hosseini, Seyedeh Maryam Fakhr, et al. "“Listen2dRoom”: Helping Visually Impaired People Navigate Indoor Environments Using an Ultrasonic Sensor-Based Orientation Aid." Georgia Institute of Technology, 2014.

[117] Vapnik, Vladimir, and Rauf Izmailov. "Knowledge transfer in SVM and neural networks." Annals of Mathematics and Artificial Intelligence (2017): 1-17.

[118] Aguilar, Wilbert G., et al. "Pedestrian detection for UAVs using cascade classifiers with meanshift." Semantic Computing (ICSC), 2017 IEEE 11th International Conference on. IEEE, 2017.

[119] Racoma, J. Angelo. "Indooratlas uses magnetic fields for location-awareness in buildings and large structures." (2013).

[120] Barberis C, Andrea B, Giovanni M, et al. Experiencing indoor navigation on mobile devices. It Professional, 2014, 16(1):50–57.

[121] Jiang S, Alves A, Rodrigues F, et al. Mining point-of-interest data from social networks for urban land use classification and disaggregation. Computers, Environment and Urban Systems, 2015, 53:36–46.

[122] Gonc¸ alves T A G. Location based Social Network. 2014.

[123] Postel M. Point-of-interest recommendation in location based social networks with subject and location awareness. 2013.

[124] Ye M, Yin P, Lee W C, et al. Exploiting geographical influence for collaborative point-of-interest recommendation. Proceedings of Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. ACM, 2011. 325–334.

[125] Liu B, Xiong H, Papadimitriou S, et al. A general geographical probabilistic factor model for point of interest recommendation. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(5):1167–1179.

[126] Liu X, Liu Y, Aberer K, et al. Personalized point-of-interest recommendation by mining users’ preference transition. Proceedings of Proceedings of the 22nd ACM international conference on Information & Knowledge Management. ACM, 2013. 733–738.

[127] Liu B, Fu Y, Yao Z, et al. Learning geographical preferences for point-of-interest recommendation. Proceedings of Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2013. 1043–1051.

[128] Yuan Q, Cong G, Ma Z, et al. Time-aware point-of-interest recommendation. Proceedings of Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2013. 363–372.

[129] Cheng C, Yang H, King I, et al. Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks. Proceedings of Aaai, volume 12, 2012. 17–23.

[130] Bao J, Zheng Y, Wilkie D, et al. Recommendations in location-based social networks: a survey. Geoinformatica, 2015, 19(3):525–565

[131] Retscher G. Indoor Navigation. 2016.

[132] Lijding M, Benz H, Meratnia N, et al. Smart signs: Showing the way in smart surroundings. 2006.

[133] Peternier A, Righetti X, Hopmann M, et al. Chloe@ University: an indoor, mobile mixed reality guidance system. Proceedings of Proceedings of the 2007 ACM symposium on Virtual reality software and technology. ACM, 2007. 227–228.

[134] Ferreira A P G, Silva T H, Loureiro A A F. Beyond Sights: Large Scale Study of Tourists' Behavior Using Foursquare Data. Proceedings of 2015 IEEE International Conference on Data Mining Workshop (ICDMW). IEEE, 2015. 1117–1124.

[135] Baral R, Li T. MAPS: A Multi Aspect Personalized POI Recommender System. Proceedings of Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016. 281– 284.

[136] Scissors L, Burke M, Wengrovitz S. What's in a Like?: Attitudes and behaviors around receiving Likes on Facebook. Proceedings of Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing. ACM, 2016. 1501–1510.

[137] Gao H. Personalized POI Recommendation on Location-Based Social Networks[Doctor Thesis]. ARIZONA STATE UNIVERSITY, 2014.

[138] Gao H, Tang J, Hu X, et al. Content-Aware Point of Interest Recommendation on LocationBased Social Networks. Proceedings of AAAI. Citeseer, 2015. 1721–1727.

[139] Yang D, Zhang D, Yu Z, et al. A sentiment-enhanced personalized location recommendation system. Proceedings of Proceedings of the 24th ACM Conference on Hypertext and Social Media. ACM, 2013. 119–128.

[140] Berjani B, Strufe T. A recommendation system for spots in location-based online social networks. Proceedings of Proceedings of the 4th Workshop on Social Network Systems. ACM, 2011. 4.

[141] Ye M, Yin P, Lee W C. Location recommendation for location-based social networks. Proceedings of Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems. ACM, 2010. 458–461.

[142] Gao H, Tang J, Hu X, et al. Exploring temporal effects for location recommendation on location-based social networks. Proceedings of Proceedings of the 7th ACM conference on Recommender systems. ACM, 2013. 93–100.

[143] Rizzo, Giuseppe, et al. "The 3cixty Knowledge Base for Expo Milano 2015: Enabling Visitors to Explore the City." Proceedings of the 8th International Conference on Knowledge Capture. ACM, 2015.

[144] Luo, Junhai, and Huanbin Gao. "Deep belief networks for fingerprinting indoor localization using ultrawideband technology." International Journal of Distributed Sensor Networks 12.1 (2016): 5840916.

[145] Gu, Yang, et al. "Semi-supervised deep extreme learning machine for Wi-Fi based localization." Neurocomputing 166 (2015): 282-293.

[146] Shu, Yuanchao, et al. "Magicol: Indoor localization using pervasive magnetic field and opportunistic WiFi sensing." IEEE Journal on Selected Areas in Communications 33.7 (2015): 1443-1457.

[147] Angermann, M.; Frassl, M.; Doniec, M.; Julian, B.J.; Robertson, P. Characterization of the indoor magnetic field for applications in localization and mapping. In Proceedings of 2012 International Conference on the Indoor Positioning and Indoor Navigation (IPIN), Sydney, Australia, 13–15 November 2012; pp. 1–9.

[148] Karimi, H. Indoor Wayfinding and Navigation; Taylor Francis: Boca Raton, FL, USA, 2015.

[149] Yang X, Steck H, Guo Y, et al. On top-k recommendation using social networks. Proceedings of ACM Conference on Recommender Systems, 2012. 67–74.

[150] Hu B, Ester M. Social Topic Modeling for Point-of-Interest Recommendation in LocationBased Social Networks. Proceedings of IEEE International Conference on Data Mining, 2014. 845–850.

[151] Kaixu Liu, Gianmario Motta, Tianyi Ma: Navigation Services for Indoor and Outdoor User Mobility: an overview. The 9th International Conference on Service Science (ICSS 2016)

[152] Liu, Kaixu, et al. "A 2D and 3D Indoor Mapping Approach for Virtual Navigation Services." Service-Oriented System Engineering (SOSE), 2017 IEEE Symposium on. IEEE, 2017.

[153] Liu, Kaixu, Gianmario Motta, and Juncheng Dong. "Wi-Fi-Aided Magnetic Field Positioning with Floor Estimation in Indoor Multi-Floor Navigation Services." Internet of Things (ICIOT), 2017 IEEE International Congress on. IEEE, 2017.

[154] Liu, Kaixu, Gianmario Motta, and Tianyi Ma. "XYZ Indoor Navigation through Augmented Reality: A Research in Progress." Services Computing (SCC), 2016 IEEE International Conference on. IEEE, 2016.

[155] LeCun, Yann, Koray Kavukcuoglu, and Clément Farabet. "Convolutional networks and applications in vision." Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on. IEEE, 2010.

[156] Kim, Sung-Eun, et al. "Effects of tourism information quality in social media on destination image formation: The case of Sina Weibo." Information & Management (2017).

[157] Zhang, Chi, et al. "GROPING: Geomagnetism and cROwdsensing Powered Indoor NaviGation." Mobile Computing, IEEE Transactions on 14.2 (2015): 387-400.

[158] Tsai, TeSheng, et al. "Indoor user navigation for enhanced quality of experience in wireless local area networks." Mobile Computing and Ubiquitous Networking (ICMU), 2015 Eighth International Conference on. IEEE, 2015.

[159] Ahmetovic, Dragan, et al. "Achieving practical and accurate indoor navigation for people with visual impairments." Proceedings of the 14th Web for All Conference on The Future of Accessible Work. ACM, 2017.

[160] Sprunk, Christoph, et al. "An experimental protocol for benchmarking robotic indoor navigation." Experimental Robotics. Springer International Publishing, 2016.

List of Publications

Level A：

[1] Linlin You, Gianmario Motta, Kaixu Liu, Tianyi Ma: CITY FEED: A Pilot System of Citizen-sourcing for City Issue Management. ACM Transactions on Intelligent Systems and Technology (TIST) 7.4 (2016): 53.

[2] Kaixu Liu, Gianmario Motta, Tianyi Ma: XYZ Indoor Navigation Through Augmented Reality: a research in progress. 2016 IEEE International Conference on Services Computing (SCC). IEEE, 2016.

[3] Tianyi Ma, Gianmario Motta, Kaixu Liu: MOBANA: a Distributed Stream-based Information System for Public Transit. 2016 IEEE International Conference on Services Computing (SCC). IEEE, 2016.

[4] Tianyi Ma, Gianmario Motta, Kaixu Liu: Delivering Real-time Information Services on Public Transit: a Framework. IEEE Transactions on Intelligent Transportation Systems 2017

[5] Liu, Kaixu, Gianmario Motta, and Juncheng Dong. "Wi-Fi-Aided Magnetic Field Positioning with Floor Estimation in Indoor Multi-Floor Navigation Services." Internet of Things (ICIOT), 2017 IEEE International Congress on. IEEE, 2017.

[6] Muhammad Hassan Arif, Jianxin Li, Muhammad Iqbal, Kaixu Liu: Sentiment Analysis and Spam Detection in Short Informal Text using Learning Classifier Systems. Soft Computing (2017): 1-11, 10.1007/s00500-017-2729-x.

[7] Gianmario Motta, Tianyi Ma, Kaixu Liu, et al.: Overview of Smart White Canes: Connected Smart Cane from Front End to Back End. In book: Mobility of Visually Impaired People: pp.469-535. DOI: 10.1007/978-3-319-54446-5_16.

Level B：

[8] Linlin You, Gianmario Motta, Kaixu Liu, Tianyi Ma: A Pilot Crowdsourced City Governance System: CITY FEED. CSE 2014: 1514-1519

[9] Gianmario Motta, Daniele Sacco, Tianyi Ma, Linlin You, Kaixu Liu: Personal Mobility Service System in Urban Areas: The IRMA Project. SOSE 2015: 88-97

[10] Kaixu Liu, Gianmario Motta, Tianyi Ma, Tao Guo: Multi-floor Indoor Navigation with Geomagnetic Field Positioning and Ant Colony Optimization Algorithm. DOI: 10.1109/SOSE.2016.18, 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)

[11] Kaixu Liu, Gianmario Motta, Bige Tunçer, Iman ABUHASHISH: A 2D and 3D Indoor Mapping Approach for Virtual Navigation Services. Service-Oriented System Engineering (SOSE), 2017 IEEE Symposium on. IEEE, 2017.

[12] Iman ABUHASHISH, Gianmario MOTTA, Tianyi Ma, Kaixu LIU: An Analysis of Social Data Credibility for Services Systems in Smart Cities – Credibility Assessment and

Classification of Tweets. 2nd EAI International Conference on ICT Infrastructures and Services for Smart Cities. IISSC 2017.

Level C：

[13] Kaixu Liu, Gianmario Motta, Linlin You, Tianyi Ma: A Threefold Similarity Analysis of Crowdsourcing Feeds. 2015 International Conference on Service Science (ICSS). IEEE, 2015. ICSS 2015: 93-98

[14] Gianmario Motta, Kaixu Liu: The Lab-Internship Alternative. Software Engineering Education Going Agile. Springer International Publishing, 2016. 81-90.

[15] Gianmario Motta, Tianyi Ma, Kaixu Liu: Software / Services Engineering education in a China-Europe collaboration program. 11th China – Europe International Symposium on Software Industry Orientated Education. CEISEE 2015.

[16] Kaixu Liu, Gianmario Motta, Tianyi Ma, Ke Fan: A threefold similarity analysis of crowdsourcing feeds. International Journal of Information Technology and Management (IJITM) 2016

[17] Kaixu Liu, Gianmario Motta, Tianyi Ma: Navigation Services for Indoor and Outdoor User Mobility: an overview. ICSS 2016

[18] Gianmario Motta, Tianyi Ma, Kaixu Liu: Software / Services Engineering education in a China-Europe collaboration program. 12th China – Europe International Symposium on Software Industry Orientated Education. CEISEE 2016.

[19] Gianmario MOTTA, Antonella LONGO, Kaixu LIU, Iman ABUHASHISH: Service Systems for Digital Services: A Framework. International Conference on Business and Information. BAI 2017.

[20] Gianmario MOTTA, Kaixu LIU, Iman ABUHASHISH, Michela Meazza: Integration of Services Systems: An Education Roadmap. 13th China – Europe International Symposium on Software Industry Orientated Education. CEISEE 2017.

Acknowledgement

The fruitful three years' Ph.D. study has brought me valuable knowledge and experience, which are foundations of my future academic career.

I would like to express my gratitude to a number of people without whose support this work would have been impossible. First of all, I would like to express my deep gratitude to my supervisor Prof. Gianmario Motta, who continuously supports me during the whole time of my study in University of Pavia, Italy. In the same time, I also would like to express my gratitude to other professors who assisted my research, Prof. Bige Tuncer, Prof. Feng Jiang, etc.

The developments of this project introduced in this dissertation could not have been possible without the contributions of a number of master students who graduated from service engineering laboratory. Moreover, my thanks go to my colleagues: Linlin You, Tianyi Ma and Iman Abuhashishi, who support the projects and help me during my work.

I also thank my parents, Chunmin Yuan and Jiechun Liu, who care much about me in my daily life. They brighten my life, widen my thought, and strengthen my confidence. Last but not least, I thank my boyfriend, Yutian Yuan who give me infinite love. He comforts me with his extreme gentleness and his technical guidance. Definitely, this work would not be possible to be accomplished without their continuously support and encouragement.