MONITORING-CAMERA-ASSISTED SLAM FOR INDOOR POSITIONING AND NAVIGATION

MONITORING-CAMERA-ASSISTED SLAM FOR INDOOR POSITIONING AND NAVIGATION

BY

HAOYUE ZHENG, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2021)	McMaster University
(Electrical & Computer Engineering)	Hamilton, Ontario, Canada

TITLE:	Monitoring-Camera-Assisted SLAM for Indoor Positioning and Navigation
AUTHOR:	Haoyue Zheng B.Eng. (Telecommunication Engineering), Beijing University of Posts and Telecommunications, Beijing, China
SUPERVISOR:	Dr. Jun Chen
NUMBER OF PAGES:	xiv, 86

Abstract

In the information age, intelligent indoor positioning and navigation services are required in many application scenarios. However, most current visual positioning systems cannot function alone and have to rely on additional information from other modules. Nowadays, public places are usually equipped with monitoring cameras, which can be exploited as anchors for positioning, thus enabling the vision module to work independently.

In this thesis, a high-precision indoor positioning and navigation system is proposed, which integrates monitoring cameras and smartphone cameras. Firstly, based on feature matching and geometric relationships, the system obtains the transformation scale from relative lengths in the cameras' perspective to actual distances in the floor plan. Secondly, by scale transformation, projection, rotation and translation, the user's initial position in the real environment can be determined. Then, as the user moves forward, the system continues to track and provide correct navigation prompts.

The designed system is implemented and tested in different application scenarios. It is proved that our system achieves a positioning accuracy of $0.46m$ and a successful navigation rate of 90.6%, which outperforms the state-of-the-art schemes by 13% and 3% respectively. Moreover, the system latency is only $0.2s$, which meets the real-time demands.

In summary, assisted by widely deployed monitoring cameras, our system can provide users with accurate and reliable indoor positioning and navigation services.

To my dear parents

Acknowledgements

Here I would like to express my sincere appreciation to all the people who have helped me in finishing this thesis.

First of all, I would like to show the deepest gratitude to my supervisor, Dr. Jun Chen. In the initial stage of the research, he provided guidance and suggestions selflessly. And when I was later working on the project, he also gave me continuous support and wise advice. His rigorous attitude towards academic work and full enthusiasm for scientific research greatly influenced me, which was the most precious wealth I gained during my study. I feel heartily honoured and proud to be his student.

In addition, I would like to genuinely thank Dr. Dongmei Zhao and Dr. R. Tharmarasa for kindly being my committee members and taking the time to read my thesis. Their useful comments and valuable suggestions are very helpful to me.

Moreover, I am very grateful to my friends, Yanning Li and Zijun Wu, for their silent companionship and valuable inspiration during my research. Whenever I encounter obstacles, they will promptly lend me a helping hand. Moreover, they can often introduce fresh ideas and novel perspectives to my thesis. Without their support, this work would not have been possible.

Last but not least, I would like to appreciate my parents for their unconditional encouragement and care. Thus, I can have the confidence to overcome all challenges and difficulties. Their love for me has illuminated my life and motivated me to become a better person.

Contents

List of Figures

List of Tables

Abbreviations

AR	Augmented Reality
BA	Bundle Adjustment
BoW	Bag-of-Words
CDF	Cumulative Distribution Function
CPU	Central Processing Unit
EKF	Extended Kalman Filter
EPnP	Efficient-Perspective-n-Point
GP-LVM	Gaussian Process Latent Variable Model
GPS	Global Positioning System
GPU	Graphics Processing Unit
g2o	General Graphic Optimization
IMU	Inertial Measurement Unit
IR	Infrared Ray

LOS	Line-of-Sight
MR	Mixed Reality
NLOS	Non-Line-of-Sight
OpenCV	Open-Source Computer Vision Library
PnP	Perspective-n-Point
POI	Point of Interest
P2P	Peer-to-Peer
RAM	Random Access Memory
RANSAC	Random Sample Consensus
RFID	Radio Frequency Identification
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURF	Speeded Up Robust Features
SVD	Singular Value Decomposition
VO	Visual Odometry
VR	Virtual Reality

Chapter 1

Introduction and Problem Statement

1.1 Indoor Positioning and Navigation

Over the past decade, with the modernization of cities, the number of large buildings increases rapidly. Moreover, with the continuous expansion of the building area, the internal environments of hospitals, art museums, office buildings, subway stations, shopping centres and other buildings which are closely related to people's lives are becoming increasingly complex. For people who are new to a large building, it becomes more and more difficult to quickly and accurately find the destination in the building.

Most existing positioning and navigation techniques are usually designed for outdoor environments, which are mainly based on the GPS module in smartphones or other electronic devices to obtain the current location information. But in indoor environments, due to the block of walls and windows, the satellite signal strength of the GPS tends to be greatly weakened and thus becomes ineffective. Moreover, indoor positioning always requires higher precision, since the deviation of a few meters may cause severe location errors and consequently result in misleading navigation information for users.

Therefore, the research of new high-accuracy indoor positioning and navigation technology (Zafari *et al.*, 2019) has broad prospects of application.

For example, for indoor scenarios with many floors and complex environments, people can use the indoor positioning service to obtain their own location information in real-time, search the places of interest around according to the positioning, and then get personal navigation instructions to reach the destination quickly and accurately. In large shopping malls, businesses can analyze the hot spots of flow through the real-time tracking of customers, and then reasonably arrange the display cases and other facilities to boost sales. Moreover, for users themselves, in some crowded large-scale public places, the locations of the elderly and children can be quickly determined through the positioning function of their mobile devices, so as to prevent them from getting lost.

In addition, intelligent positioning and navigation are also the key technologies for robots to realize automated moving. In recent years, artificial intelligence has become one of the most popular fields of research. Google, Apple and other tech giants have launched their own intelligent robots. Therefore, with the rapid development of smart robots, automated indoor positioning and navigation techniques will also find wide applications.

As discussed above, it is of great significance to develop a low-cost, low-power and high-precision indoor positioning and navigation system.

At present, there are many indoor positioning methods, mainly based on wireless signals, inertial sensors or visual images. The first kind of localization technology based on wireless signals usually utilizes Wi-Fi (Kotaru *et al.*, 2015), RFID (radio frequency identification) (Xiao *et al.*, 2017), Bluetooth (Kriz *et al.*, 2016) and IR (infrared ray) (Chen *et al.*, 2010). Although this method can achieve relatively good accuracy, the layout costs of corresponding dedicated infrastructures are high and the signals are easily affected by interference factors such as walls and glass. Therefore, this method is not amenable for wide adoption. (For example, the current Wi-Fi-based positioning systems usually use the location fingerprinting technique, which needs to collect a large amount of Wi-Fi fingerprint data to build indoor fingerprint maps, which is labour-intensive and time-consuming.) The second type of indoor positioning technique is based on inertial measurement units (IMU) (Höflinger *et al.*, 2013). At present, most mobile devices are equipped with built-in sensors, which makes the research based on inertial sensors more practical in indoor localization. However, the cumulative error of inertial positioning will gradually increase with distance. Up to now, the accuracy of mobile inertial sensors is relatively low, and the obtained data is unstable, so the performance of positioning is also not ideal.

Among the different kinds of solutions that have been proposed, vision-based indoor positioning (Taira *et al.*, 2018) has become one of the most popular schemes in practice because of its more comprehensible information forms (images or videos), user-friendly interactive interfaces and the potential for further integration with AR or MR technology. After the visual SLAM (Simultaneous Localization and Mapping) (Fuentes-Pacheco *et al.*, 2015) was proposed, a lot of research based on this technique came into being. In brief, visual SLAM is to detect and perceive the environment with the help of cameras, estimate the current location and build a map for the surroundings. In the next section, more information about visual SLAM technology will be further detailed.

1.2 Visual SLAM Framework and Mathematical Descriptions

1.2.1 Visual SLAM Framework

As mentioned in the previous section, SLAM (Bailey and Durrant-Whyte, 2006) is the abbreviation of simultaneous localization and mapping. It means that in an unknown environment, a moving robot equipped with specific sensors constantly estimates its real-time position, while simultaneously establishing a model of the surroundings, namely the map (Davison *et al.*, 2007). If the sensor here is a camera, then SLAM is vision-based and is called visual SLAM. In other words, visual SLAM is to infer the camera motion and the surrounding environment according to a series of continuous images, which form a video. Briefly speaking, visual SLAM is to solve the two problems of positioning and mapping with the aid of the camera.

The classic visual SLAM framework is shown in Figure 1.1, which illustrates the constituent modules of visual SLAM.

The whole visual SLAM process includes the following steps:

1. **Sensor Data Reading.** In visual SLAM, this step focuses on obtaining and

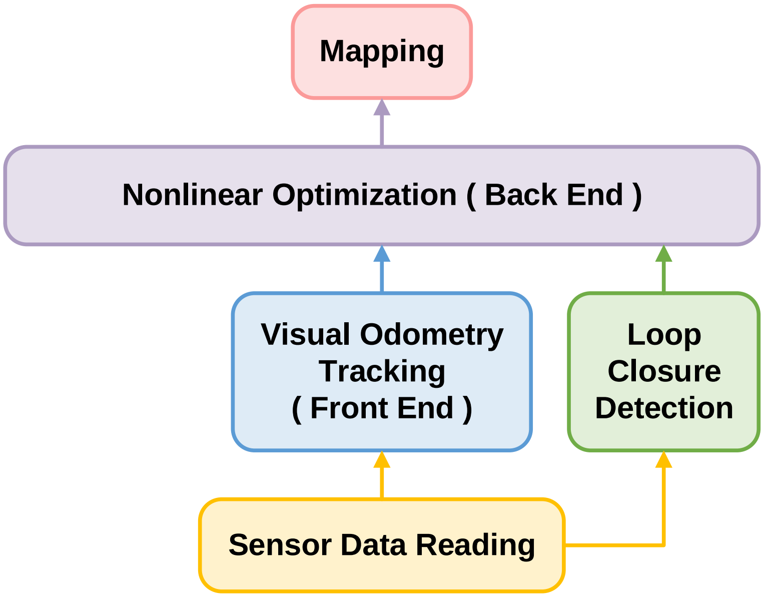

Mapping
Nonlinear Optimization (Back End)
Visual Odometry Tracking (Front End)
Loop Closure Detection
Sensor Data Reading

Figure 1.1: Classic Visual SLAM Framework

preprocessing the images from cameras. For the robots, this part may also include retrieving data from inertial measurement units (IMUs) and other sensors.

2. **Visual Odometry Tracking (Front End).** The task of Visual Odometry (VO) (Aqel *et al.*, 2016) is to estimate the camera motion and the structure of the local map between adjacent images. VO is also called the front end.

3. **Loop Closure Detection.** It is used to determine whether the robot with a camera is back to a previous position. If a loop closure is detected, the information will be provided to the back end for processing.

4. **Nonlinear Optimization (Back End).** The back end receives the poses of the camera measured by VO at different times and the information of loop closure detection, optimizes them, and then constructs a globally consistent trajectory. Because it is connected after VO, it is also called the back end.

5. **Mapping.** Based on the estimated trajectory, a map corresponding to the demand will be established.

In the following, the specific functions of each module are described in detail separately.

- **Visual Odometry Tracking (Front End)**

  Visual Odometry (VO) focuses on the camera motion between adjacent images. But for the computer, this is not an intuitive problem. In visual SLAM, the computer can only get the pixels in images, which are the projections of some spatial points on the imaging plane of the camera. Therefore, to quantitatively estimate the camera motion, the geometric relationship between the camera and the spatial points is needed, which will be introduced later. In brief, visual odometry can track the camera motion and restore the spatial structure of the scene through the adjacent image frames.

  However, if only VO is used to track the camera, the cumulative drift will inevitably occur. This is due to the fact that VO only estimates the motion between two adjacent images. Since each estimation has a certain error, and the previous error will be transferred to the next one, then after a period of time, the trajectory constructed by visual odometry will no longer be accurate. This is called the drift, which leads to the failure of building a consistent map. To solve the drift problem, two steps are needed: nonlinear optimization and loop closure detection. Loop closure detection is responsible for detecting the fact that the current position of the camera has already been reached before, while nonlinear optimization corrects the shape of the whole trajectory according to this information.

- **Nonlinear Optimization (Back End)**

  Generally speaking, nonlinear optimization mainly refers to dealing with the noise in the SLAM process. It estimates the state of the whole system from the sensor data with noise and calculates the uncertainty of this state estimation, which is called maximum a posteriori estimation. The state here includes not only the trajectory of the camera but also the map.

  In contrast, the VO module is usually referred to as the front end. In the SLAM framework, the front end transmits the sensor data with its initial value to the back end, while the back end is responsible for the overall optimization process. In visual SLAM, the front end is more related to the field of computer vision, such as image feature extraction and matching, while the back end focuses on filtering and nonlinear optimization, estimating the mean and variance of the state.

- **Loop Closure Detection**

  Loop Closure Detection mainly solves the problem of position estimation drifting with time. To achieve this, the robot with a camera needs to have the ability to recognize the scene that has been reached before. For example, loop closure detection can be accomplished by judging the similarity between the images captured by a camera. Then, based on the detected loop closure information, like "A and B are the same point", the back end can adjust and optimize accordingly. Therefore, the accumulated errors can be eliminated, and the globally consistent trajectory and map can be established.

- **Mapping**

A map is a description of the environment, but its form is not fixed. It needs to be determined according to the specific application scenarios of SLAM to meet the different needs of users. In general, it can be divided into two types, sparse maps and dense maps. Sparse maps, which have been abstracted to a certain extent, are composed of some representative landmarks. In contrast, dense maps focus on modelling everything in sight. For positioning, sparse maps are sufficient, while for navigation, dense maps are needed.

1.2.2 Mathematical Descriptions of SLAM

As above, the composition of SLAM and the main functions of each module have been shown intuitively. Next, the whole process of SLAM will be described mathematically.

Suppose a robot with sensors is moving in an unknown environment. First of all, because the camera usually collects visual data at certain moments, these discrete moments are noted as $t = 1, \ldots, K$. In the meantime, the corresponding positions of the robot at these moments are $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K$ respectively, which form the motion trajectory of the robot. The map is assumed to be composed of N landmarks, which are $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$. At each moment, the sensor will observe a part of the landmarks and get their measurements.

In this setting, the problem that a robot with sensors moves in an unknown environment is described from the following two aspects:

1. **Motion**: From time $k-1$ to k, the robot position changes from $\boldsymbol{x}_{k-1}$ to $\boldsymbol{x}_k$.
2. **Observation**: At time k, the robot detects a landmark $\boldsymbol{y}_j$ at position $\boldsymbol{x}_k$.

Firstly, for motion, the robot usually carries a sensor to measure its own motion, such as an inertial sensor. (The sensor data may not be the position directly, but some indirect information instead, such as the acceleration and angular velocity.) A general and abstract mathematical model can be used to represent the motion as follows:

$$\boldsymbol{x}_k = f\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}_k\right) \tag{1.2.1}$$

where $\boldsymbol{u}_k$ is the reading of the motion sensor, and $\boldsymbol{w}_k$ is the noise added to this process. Note that a general function f is used to describe this process without specifying how f acts. This allows the whole function to refer to any motion sensor and remain as a general equation. It is called the motion equation.

Secondly, corresponding to the motion equation, there is also an observation equation. The observation equation describes when the robot sees a landmark $\boldsymbol{y}_j$ at the position $\boldsymbol{x}_k$, the observation data $\boldsymbol{z}_{k,j}$ is generated. Similarly, an abstract function h is used to describe this relationship:

$$\boldsymbol{z}_{k,j} = h\left(\boldsymbol{y}_j, \boldsymbol{x}_k, \boldsymbol{v}_{k,j}\right) \tag{1.2.2}$$

where $\boldsymbol{v}_{k,j}$ is the noise in this observation. Since there are many different kinds of sensors that can be used, the observation data $\boldsymbol{z}$ and the observation equation h also have many different forms.

In the context of visual SLAM, the sensor is a camera, and the observation equation represents the process of obtaining pixels in the images after shooting the landmarks. This process is based on the imaging principle of pinhole cameras.

For different sensors, these two equations have different parametric forms. In general, the SLAM process can be summarized into the following two basic equations:

$$\begin{cases} \boldsymbol{x}_k = f\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}_k\right) \\ \boldsymbol{z}_{k,j} = h\left(\boldsymbol{y}_j, \boldsymbol{x}_k, \boldsymbol{v}_{k,j}\right) \end{cases}. \tag{1.2.3}$$

These two equations describe the most basic SLAM problem, that is, how to solve the positioning problem (estimating $\boldsymbol{x}$) and the mapping problem (estimating $\boldsymbol{y}$) when we know the reading $\boldsymbol{u}$ of the motion sensor and the reading $\boldsymbol{z}$ of the observation sensor. In this way, the SLAM problem is modelled as a state estimation problem, that is, how to estimate the internal and hidden state variables from the measured data with noise.

In the following sections, $\boldsymbol{x}$ will be specified as the pose, which is composed of the orientation and position. In other words, $\boldsymbol{x}$ is determined by rotation and translation.

1.3 Scale Ambiguity of Monocular SLAM

As mentioned before, visual SLAM is mainly based on the camera, which can record the surrounding environment at a certain rate to form a continuous video stream. According to different working methods, cameras can be divided into three main categories: monocular cameras, stereo cameras and RGB-D cameras. Intuitively, the monocular camera has only one lens, while the stereo camera has two and the RGB-D camera usually carries multiple lenses.

The principle of stereo cameras is similar to that of human eyes, that is, the distance of an object is determined by comparing the difference between the two images taken by the left and right cameras. However, because of calculating this disparity, stereo cameras require a large number of computations to estimate the depth of each pixel in the images. Moreover, the accuracy of the depth calculated by stereo cameras is limited by the resolution.

RGB-D cameras use the infrared structured light or time-of-flight (TOF) principle. RGB-D cameras can resolve the distance between an object and the camera by actively emitting light to the object and then obtaining information from the reflected light. However, most RGB-D cameras still have many drawbacks, such as narrow measurement range, high noise, small field of view and so on.

The monocular camera uses only one lens. Due to its simple structure and low cost, this type of camera has been widely used in practice. Accordingly, the method of SLAM using the monocular camera is called monocular SLAM. In this thesis, we will focus on monocular SLAM.

The data of a monocular camera is the image, which is essentially a projection of a scene on the imaging plane of the camera. The image represents the three-dimensional world in a two-dimensional form. Obviously, this process loses one dimension of the scene, which is called the depth (or the distance). So only through a single image, we cannot calculate the physical distance between the camera and an object in the real environment. Then, the actual size of this object cannot be determined either. It may be a very large but distant object or a smaller but closer object. Due to the principle of perspective, they may look the same in the image.

Since the image captured by a monocular camera is only a two-dimensional projection of the three-dimensional space, in order to restore the three-dimensional structure of the scene through the image, we must change the viewpoint of the camera. Therefore, we have to move the camera to estimate its motion and the distance and size of the objects in the scene, which is called the structure. For the motion of the camera, if it moves to the right, the corresponding object in the captured image will move to the left, which provides information to infer the motion. For the structure of the scene, since the nearby objects move faster and the distant objects move more slowly, the motion of these objects in the image forms a disparity when the camera moves. From this disparity, the distance relationship of objects can be quantified.

However, the obtained distance is only a relative value, and the actual size of these objects in the real environment still cannot be determined. For example, if the motion of the camera and the size of the scene are magnified by the same factor in the mean time, the monocular camera will get the same image. This indicates that the trajectory and map estimated by monocular SLAM differ from the ground truth by one factor, which is called the Scale. Accordingly, the problem that monocular SLAM cannot determine the actual scale from a single image is called the Scale Ambiguity.

The above two problems, namely, the distance can only be calculated by moving the camera and the actual scale in the real environment cannot be determined, cause trouble for the application of monocular SLAM. The core reason is that the depth of the scene is lost in the imaging process of the camera. Thus, in the following chapters, how to solve the scale ambiguity of monocular SLAM will be the focus of discussion.

1.4 Monitoring Cameras in Indoor Scenarios

In order to apply monocular SLAM to indoor positioning and navigation systems, the location, distance and direction information in the real environment is necessary. But at present, most hand-held cameras, such as the rear cameras of smartphones, are typically monocular cameras. Although some new smartphones are equipped with two or more cameras, the parameters and properties of these cameras are not consistent. As described in the previous section, the SLAM based on the monocular camera can only obtain the relative pixel scale rather than the actual scale in the real environment. Therefore, if monocular SLAM is used for indoor positioning and navigation, the transformation relationship from the pixel scale to the actual scale is crucial.

Moreover, even if the physical distances are obtained, in monocular SLAM, the positions of objects are described in the perspective of the smartphone camera, rather than in the floor plan of the real scenario. However, for users, both the current location and the destination correspond to the semantic information on the map. Therefore, it is of great importance for the positioning and navigation system to connect the smartphone camera coordinate system in monocular SLAM with the plane coordinate system on the floor plan.

Currently, public places, such as art museums, educational buildings, subway stations, supermarkets, etc. are usually equipped with monitoring cameras (Zafari *et al.*, 2019) that cover most of the areas. On this basis, an idea is put forward to effectively solve the above problems by leveraging monitoring cameras to assist monocular SLAM for accurate indoor positioning and navigation. There are two distinct advantages of this idea. First of all, the information provided by monitoring cameras is in real-time, which helps to continuously update the position of the smartphone camera during the movement. Secondly, the locations of monitoring cameras are usually fixed for a long time, which can be provided to the system as prior information. Then monitoring cameras can be exploited as anchors to associate the smartphone camera's perspective with the real world, so as to determine the user's physical location in the floor plan.

1.5 Contributions

In this thesis, an indoor positioning and navigation system assisted by monitoring cameras is proposed to tackle the scale ambiguity of monocular SLAM. Firstly, during the movement of the user, the system can leverage the monitoring camera as an anchor to locate the actual position of the smartphone camera by associating the mobile coordinate system with the plane coordinate system. As to the follow-up navigation service, the system will detect whether there are other monitoring cameras in the path that can be used as new anchor points for relocating, so as to continuously track the motion of the smartphone camera. Then, the system can guide the user to the subsequent landmarks on the planned navigation route according to the real-time position information feedback, until the final destination is reached.

In summary, the main innovations and contributions of this work are as follows:

- By exploiting monitoring cameras as anchor points, the proposed system does not require extensive preliminary site surveys, which is labour-saving and time-efficient. With this system, most indoor public places equipped with monitoring cameras can be improved to provide positioning and navigation services for users.

- Innovative algorithms are designed to handle the challenges during the system implementation, which include the actual scale transformation algorithm for solving the scale ambiguity problem in Monocular SLAM, as well as the coordinate conversion algorithm for connecting the smartphone camera's perspective with the floor plan. Based on the designed algorithms, this proposed system is fully implemented, which provides a new approach to vision-based indoor positioning and navigation services.

- Extensive experiments are carried out in different indoor scenarios, and the proposed system is compared with other existing frontier systems. The experimental results verify that the system can achieve a positioning accuracy of $0.46m$ and a successful navigation rate of 90.6%, both of which are higher than those of the existing systems. Besides, the overall latency of the system is only about 0.2s, which meets the real-time demands. These results prove the superior performance and wide applicability of the system for various indoor environments in practice.

1.6 Thesis Structure

The rest of this thesis is divided into the following chapters:

First of all, Chapter 2 introduces the ORB-SLAM scheme and some current techniques related to indoor positioning and navigation.

Chapter 3 makes a brief introduction of the complete framework of the indoor positioning and navigation system designed in this thesis.

Chapter 4 explains in detail, with the help of smartphone cameras and monitoring cameras, how the system calculates the actual scale in the floor plan and then transforms the coordinates from three dimensions to two dimensions based on the improved feature extraction and matching algorithms, so as to locate the user's initial position in the real environment.

Then, Chapter 5 describes concretely, as the user moves, how our system can not only carry out high-precision tracking but also provide correct navigation prompts.

In Chapter 6, the proposed system is fully implemented and tested in various real indoor scenarios. The experimental results demonstrate that, compared with other state-of-the-art schemes, our system can achieve higher accuracy as well as stronger robustness.

Finally, Chapter 7 gives a brief summary of the designed system.

Chapter 2

Related Work

2.1 ORB-SLAM

ORB-SLAM (Mur-Artal and Tardós, 2017) is one of the most efficient and robust modern SLAM systems. For the indoor positioning and navigation system proposed in this thesis, the adopted monocular SLAM scheme is also ORB-SLAM.

ORB-SLAM represents the peak of mainstream feature-point SLAM. Compared with the previous works, ORB-SLAM has the following distinct strengths:

1. ORB-SLAM supports monocular, stereo and RGB-D modes. Therefore, ORB-SLAM has good versatility and can be applied to any kind of cameras.

2. The whole system works around ORB feature points, which is a good compromise between the accuracy and efficiency of current computing platforms. ORB is not as time-consuming as SIFT or SURF, and can be operated in real-time on the CPU. Compared with Harris or other simple corner features, ORB has good rotation and scale invariance. Moreover, ORB provides descriptors that enable

object relocation and loop closure detection, especially for the large-range motion.

3. Loop closure detection is one of the highlights of ORB-SLAM. The excellent loop detection algorithm ensures that ORB-SLAM can quickly correct the trajectory with drifts and effectively avoid the cumulative errors.

4. ORB-SLAM innovatively uses three threads to complete the whole SLAM scheme: the Tracking Thread for locating feature points in real-time, the Co-Visibility Graph Thread for optimizing the local Bundle Adjustment (BA) problem, and the Essential Graph Thread for loop closure detection and optimization of the global pose graph. Firstly, the Tracking Thread is responsible for extracting ORB feature points from each new image and comparing them with those from the nearby key frames, so as to calculate the positions of the feature points and roughly estimate the pose of the camera. Secondly, the Co-Visibility Graph Thread is used for solving a bundle adjustment problem, which includes the locations of feature points and the camera poses, with higher precision in the local space. The third thread, the Essential Graph Thread, detects the loop closure from the global map and all key frames to eliminate the cumulative error. Since there are too many map points in the global map, the optimization of this thread does not include map points but only the camera poses. This three-threaded structure enables ORB-SLAM to achieve very good tracking and mapping performance, as well as ensuring the global consistency of the trajectory and map.

5. ORB-SLAM makes many improvements around feature points, e.g., ensuring

a uniform distribution of feature points based on the feature extraction algorithms in OpenCV, repeating the optimization four times to obtain more correct matches in pose estimation, and a relatively more relaxed selection strategy for key frames. These subtle improvements make ORB-SLAM far more robust than other schemes. Even in unfavorable scenarios, ORB-SLAM can still work well.

These above advantages make ORB-SLAM a peak of feature-point SLAM. Many works take ORB-SLAM as a standard and develop on its basis. The code of ORB-SLAM is known for its legibility and comprehensible annotations. Therefore, the indoor positioning and navigation system in this thesis is also modified and implemented on the basis of ORB-SLAM.

2.2 Indoor Positioning and Navigation Techniques

In this section, the techniques related to indoor positioning and navigation will be summarized in the following aspects and compared with the system proposed in this thesis.

2.2.1 Vision-Based Positioning and Navigation

At present, the indoor positioning and navigation services are mainly based on the wireless signal, inertial measurement unit (IMU), Radio Frequency Identification (RFID) and visual information. Compared with the first three methods, the main advantage of vision-based solutions is using high-definition images and rich features to obtain better performance. Some vision-based systems also need other forms of information to assist them in realizing the desired functions. For example, Overlay (Jain *et al.*, 2015) establishes a geometric representation of the surrounding environment by fusing data collected from smartphone cameras and sensors. RAVEL (Papaioannou *et al.*, 2014) utilizes the fusion of visual information and signal data to achieve high-precision positioning.

After SLAM (Bailey and Durrant-Whyte, 2006) was proposed, a large number of studies based on this technique came into being. In brief, with the help of various sensors (such as cameras, radars and other types of equipment), SLAM is about detecting and perceiving the surrounding environment, estimating the motion and building a map of the current surroundings. ORB-SLAM (Mur-Artal and Tardós, 2017), which combines DBoW2 (Zhang *et al.*, 2010) for scene identification and g2o (Kümmerle *et al.*, 2011) library for nonlinear optimization, is one of the most outstanding visual SLAM frameworks so far.

Furthermore, some studies also introduce other types of information into SLAM, creating a lot of applications related to positioning and navigation. For example, SmartSLAM (Shin *et al.*, 2011) applies SLAM to smartphones and utilizes Wi-Fi signals to help locate the user's current position. However, this technique is only effective for indoor environments with corridor layouts. WiFi-SLAM (Ferris *et al.*, 2007) leverages the Gaussian process model integrating latent variables (GP-LVM) (Lawrence and Hyvärinen, 2005) to construct a wireless signal strength map with correct connectivity. In addition, there are also some applications that exploit other forms of information, such as FootSLAM (Robertson *et al.*, 2009) which utilizes data from the built-in inertial sensor and SemanticSLAM (Abdelnasser *et al.*, 2015) which introduces semantic information to the SLAM scheme.

It can be seen that, in order to deal with the scale ambiguity, most positioning and navigation schemes based on monocular SLAM need the assistance of information from additional modules, such as wireless signals or sensor data. In contrast, the system proposed in this thesis only resorts to the visual information provided by monitoring cameras. Therefore, the actual scale can be calculated and then the user's physical position in the real environment can be obtained without relying on data in other forms. Moreover, compared to other SLAM-integrated techniques, with the help of monitoring cameras, our system can achieve stronger robustness in dynamic application scenarios.

2.2.2 Environmental-Information-Assisted Positioning and Navigation

In real indoor scenarios, in addition to the images, there are many other kinds of data that can provide help for positioning and navigation. Thus, some systems utilize information from the surrounding environment to enhance their performance. EV-Loc (Teng *et al.*, 2013) is a tracking system using visual information to assist wireless positioning, which connects the object's appearance with the electronic signals. Besides, JVWL (Liu *et al.*, 2016) fuses the data from smartphone cameras and Wi-Fi signals, as well as exploiting the deep neural network to optimize results and improve positioning accuracy. In addition, MVG (Liu *et al.*, 2017) proposes a model based on multi-perspective to achieve robust positioning, which leverages visual data and geomagnetic signals from smartphones at the same time.

The works mentioned above all introduce information other than images to improve the performance. However, introducing other kinds of data also means that the system has to deal with the potential problems brought by these forms of information, such as the trajectory drifts caused by the cumulative errors of IMU sensors in smartphones, the trouble of establishing a fingerprint database of wireless signals in advance, etc. By contrast, the input to the system proposed in this thesis is only the synchronized video frames provided by multiple cameras. By exploiting the monitoring camera as an auxiliary information source to improve the robustness, the error source of the system is confined within the form of images, which is beneficial for the subsequent adjustment and optimization.

Chapter 3

System Framework

As is shown in Figure 3.1, this chapter will give a brief introduction to the complete framework of the indoor positioning and navigation system designed in this thesis.

First of all, when starting up, the system enters the initialization stage. In this stage, the system will load the floor plan of the building as well as the abundant semantic information contained in it, including the locations of the POIs (Points of Interest), landmarks and monitoring cameras in the environment, and the division of various functional areas, etc. The above semantic information is of great importance to the positioning and navigation performance of the system.

Secondly, in the positioning and navigation stage, the smartphone camera held by the user and the monitoring cameras deployed in the area start to record the surrounding environment, and continuously transmit the video content to the system for processing and calculation.

After receiving the video frames uploaded from the smartphone camera and monitoring cameras, the system will extract and match the image features in the frames. The well-matched feature point pairs are the basis of the follow-up process and are

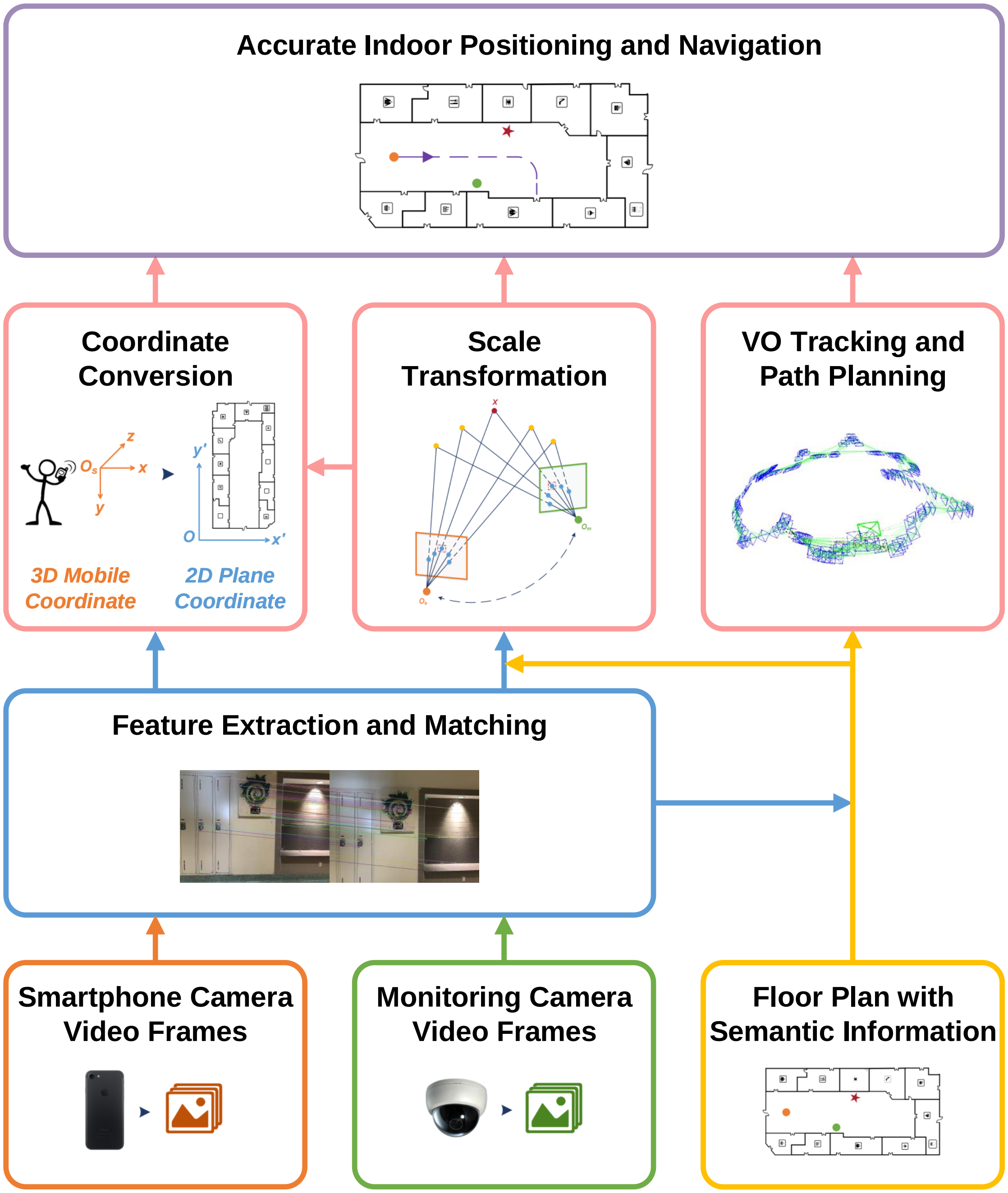

Accurate Indoor Positioning and Navigation
Coordinate Conversion
z
O_s
x
y
y'
O
x'
3D Mobile Coordinate
2D Plane Coordinate
Scale Transformation
x
O_m
O_s
VO Tracking and Path Planning
Feature Extraction and Matching
Smartphone Camera Video Frames
Monitoring Camera Video Frames
Floor Plan with Semantic Information

Figure 3.1: System Framework

crucial for other modules of the system. Although the perspectives of these two types of cameras are quite different and the cameras' parameters are not consistent, with the carefully selected feature points and the improved algorithms, our system can significantly increase the correct matching rate.

Then, the actual scale transformation module will start to detect and recognize the POI (Point of Interest) in the image frames, and further calculate the relative pose relationship of the smartphone camera and monitoring camera. Moreover, the preloaded semantic information, which here refers to the physical distance between the POI and monitoring camera in the real environment, will be leveraged to obtain the actual scale.

Furthermore, by synthesizing the previously acquired information, including the correctly matched image features in video frames and the obtained actual scale, the coordinate conversion module will calculate the transformation relationship from the three-dimensional mobile coordinates in the smartphone camera's perspective to the two-dimensional plane coordinates in the floor plan. Thus, based on the outputs of the actual scale transformation module and coordinate conversion module, our system can determine the actual initial position of the user in the floor plan.

In addition, when the smartphone camera moves into the environment where the line of sight (LOS) is blocked by the barriers (that is, the POI cannot be shot by the monitoring camera and the smartphone camera at the same time), the system will utilize the visual odometry (VO) with semantic information for continuous tracking and navigation. VO can calculate the pose changes of the smartphone camera between the adjacent video frames, and then estimate the current position of the user according to the acquired 3D-2D coordinate conversion relationship. This process will continue until the next time the system detects that the video frames uploaded by the smartphone camera and a new monitoring camera can capture the same POI. At that time, the relocation function of the system will be activated to correct the constructed motion trajectory of the user. This design can significantly reduce the accumulated error and drift in the previous process.

In order to navigate the user, our system will first build a map showing the connectivity between the landmarks based on the semantic information contained in the floor plan. With the user's input of the target location, the optimal navigation path will be planned. Then, according to the real-time tracking results, the system can navigate the user to each landmark on the planned route in order until the final destination is reached.

In summary, on the one hand, for positioning, the system achieves the high-precision positioning of the user in the floor plan by fusing the visual information from the smartphone camera and monitoring cameras. On the other hand, for navigation, the system can always track the user's current position and provide reliable navigation prompts in real-time until the user reaches the destination.

Based on these designed algorithms and modules, our proposed system can provide users with accurate and efficient indoor positioning and navigation services.

Chapter 4

Actual Initial Position Acquisition

4.1 Feature Extraction and Matching

In order to obtain the actual initial position of the user, extracting and matching feature points from the image frames of the smartphone camera and monitoring cameras plays a vital role, which is the basis of the follow-up process. Feature extraction is to select some representative points from the image, which is usually divided into two steps: key-points detection and descriptors calculation. Feature matching is to get feature point pairs with high similarity by comparing the calculated descriptors.

As mentioned before, it is not easy to directly match the image features extracted from the video frames of these two types of cameras because of the large difference in their perspectives. In addition, the parameters of the cameras are not consistent. In order to solve the above problems, our system mainly innovates from the following two aspects:

First of all, four kinds of image features are tested, namely SIFT (Lowe, 2004), SURF (Bay *et al.*, 2006), ORB (Rublee *et al.*, 2011) and A-KAZE (Alcantarilla and

Solutions, 2011), in diverse experimental environments with various angles formed by the two cameras and POI. This will be analyzed concretely later in the experimental part. By comparing the performance of the above four image feature points, A-KAZE finally stands out and is adopted by our system. A-KAZE, or Accelerated-KAZE, is an improvement on the basis of the KAZE method. Compared with SIFT and SURF, the speed of feature extraction and matching of the A-KAZE algorithm is faster. In the meanwhile, compared to ORB, the repeatability and robustness of the A-KAZE algorithm are significantly enhanced.

Secondly, the algorithm of feature extraction and matching in the pose calculation module provided by OpenCV is modified and improved, so that it can be applied to two images from cameras with different parameters.

4.2 Actual Scale Transformation

At present, image matching and relative pose calculation methods based on feature points have been utilized in various visual positioning and navigation systems (Niu *et al.*, 2019; Liu *et al.*, 2017). However, since the actual scale of the real environment cannot be obtained directly through the vision module, some other image-based indoor positioning systems cannot operate alone. Only with the assistance of information from other sources, such as sensor data from the IMU module or wireless signals from the Wi-Fi module, can these systems provide location-based services.

In contrast, because of the designed actual scale calculation algorithm, our indoor positioning and navigation system can acquire the user's actual position without additional information, only through the smartphone camera and the monitoring camera. In this section, the method of actual scale calculation will be introduced in detail.

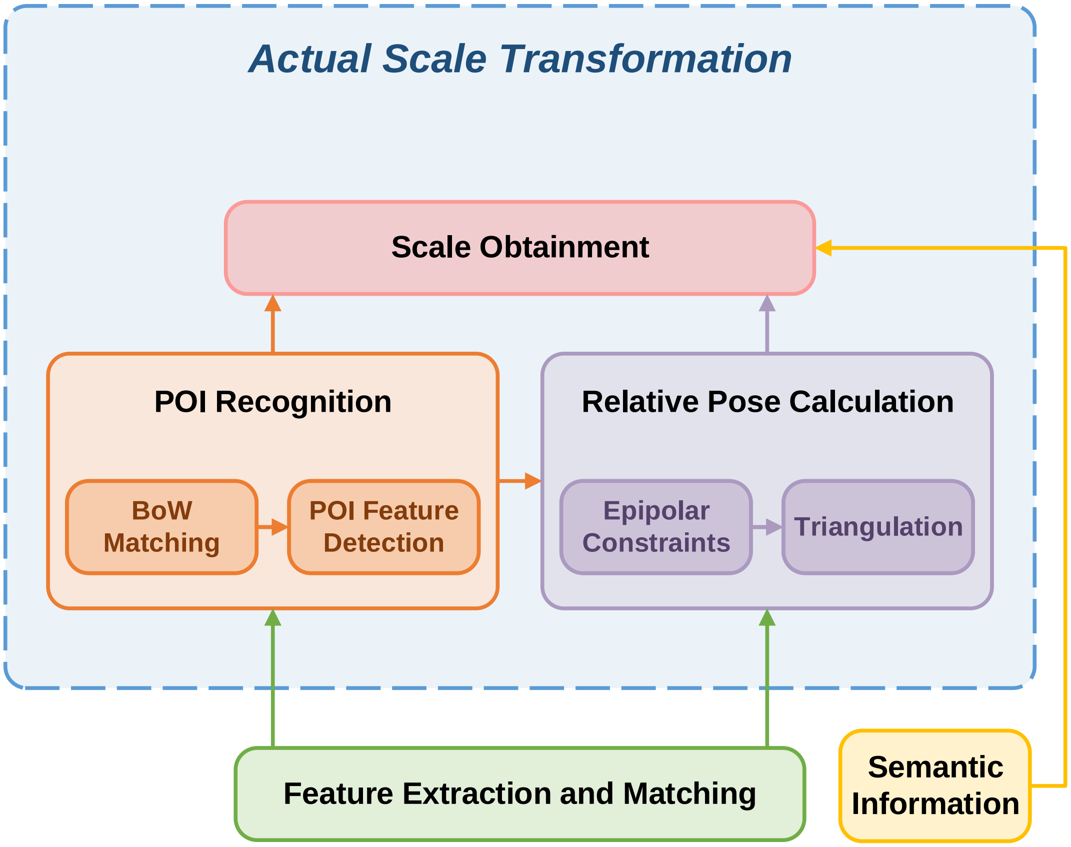


Figure 4.1: Framework of the Actual Scale Transformation Algorithm

Figure 4.1 describes the flow of the actual scale transformation method. This algorithm can be further divided into three steps, which are POI recognition, relative pose calculation and scale obtainment.

As described in Section 4.1, our system first extracts and matches A-KAZE features from the image frames of the smartphone camera and monitoring cameras. The feature point pairs with the high correct matching rate provide a solid foundation for the actual scale transformation algorithm.

1. **POI Recognition.** First of all, for POI recognition, the system utilizes the Bag-of-Words (BoW) model (Zhang *et al.*, 2010) to calculate the corresponding word vector for each image frame. By comparing the word vectors of different frames, the system will select the most suitable monitoring camera, which generally has a viewing angle with the highest similarity to the smartphone camera. Then, our system can recognize and match POI in the image frames from these two cameras.

2. **Relative Pose Calculation.** Secondly, the system solves the relative pose relationship of the selected monitoring camera and the user's smartphone camera according to the geometric constraints. Then, with the obtained pose relationship, the system calculates the relative distance of the POI and monitoring camera through the principle of triangulation.

3. **Scale Obtainment.** Finally, according to the semantic information contained in the preloaded floor plan, the ratio of the scale transformation can be obtained, so that the relative length in the smartphone camera coordinate system can be connected to the actual length in the plane coordinate system.

Next, the algorithm of actual scale transformation in our system will be divided into three subsections and described in detail respectively.

4.2.1 POI Recognition

In most public places, there are usually some objects whose positions remain unchanged over time, such as sculptures in buildings, exit signs along corridors, logo boards of stores in shopping malls, wayfinding signages in lobbies and so on. By using these fixed and eye-catching objects, important reference information can be introduced into our system. These objects, selected by the system in the real scenarios, are called POIs (Points of Interest).

The process of POI Recognition can be separated into the following two steps:

First of all, the system selects the most suitable video source from multiple monitoring camera candidates to calculate the pose relationship. Because matching image features of video frames from all monitoring cameras and the smartphone camera can take a large amount of time, which will cause obvious system latency and affect the user's experience. Therefore, our system adopts the algorithm of DBoW (Zhang *et al.*, 2010), which is based on the Bag-of-Words model to compare the differences between image frames from different monitoring cameras and the smartphone camera. Firstly, the A-KAZE feature points in all video frames are extracted, then the word vector of each frame is calculated. So, the system can compare and select the monitoring camera whose word vectors of image frames have the highest similarity to those of the smartphone camera, and then utilize this chosen monitoring camera as the anchor point for locating users. Therefore, the system saves a lot of resources and time for exhaustive image feature matching of all monitoring cameras' and the smartphone camera's video frames.

In the second step, the system identifies and locates the position of POI in the images through feature matching. Because the orientation and position of the monitoring camera in public places are always constant, it can be considered that POI is usually acquired in a fixed area in the video frame of the monitoring camera, that is, the POI recognition area of the system, as shown in Figure 4.2. Our system will extract the image feature of this specific area in the frame of the monitoring camera and match it to the frame of the smartphone camera to obtain the feature point with the highest similarity. In this way, the system can locate the position where POI is projected in both of the video frames from two cameras.

4.2.2 Relative Pose Calculation

This section mainly introduces the calculation method of the relative pose. Utilizing this algorithm, the system can obtain the relative position of the POI and monitoring camera (such as points X and O_m in Figure 4.2) in the smartphone camera coordinate system, which hereinafter is referred to as the mobile coordinate system.

As shown in Figure 4.2, after feature extraction of video frames collected by the monitoring camera and smartphone camera, the system leverages the method of image matching to calculate the pose relationship between these two cameras. The Epipolar Constraints (Zhang, 1998) are often utilized to obtain the pose difference of cameras between two image frames. Through feature matching of images from the monitoring camera and smartphone camera, the system can acquire multiple well-matched two-dimensional feature point pairs. Suppose that $\boldsymbol{x}_1$ represents a two-dimensional feature point in the video frame of the smartphone camera, $\boldsymbol{x}_2$ represents the matched one in the frame of the monitoring camera, and $\boldsymbol{X}$ represents the corresponding three-dimensional point of them in the mobile coordinate system. Besides, the above two-dimensional coordinates of feature points are converted into homogeneous forms. From the imaging principle of pinhole cameras, we can obtain the

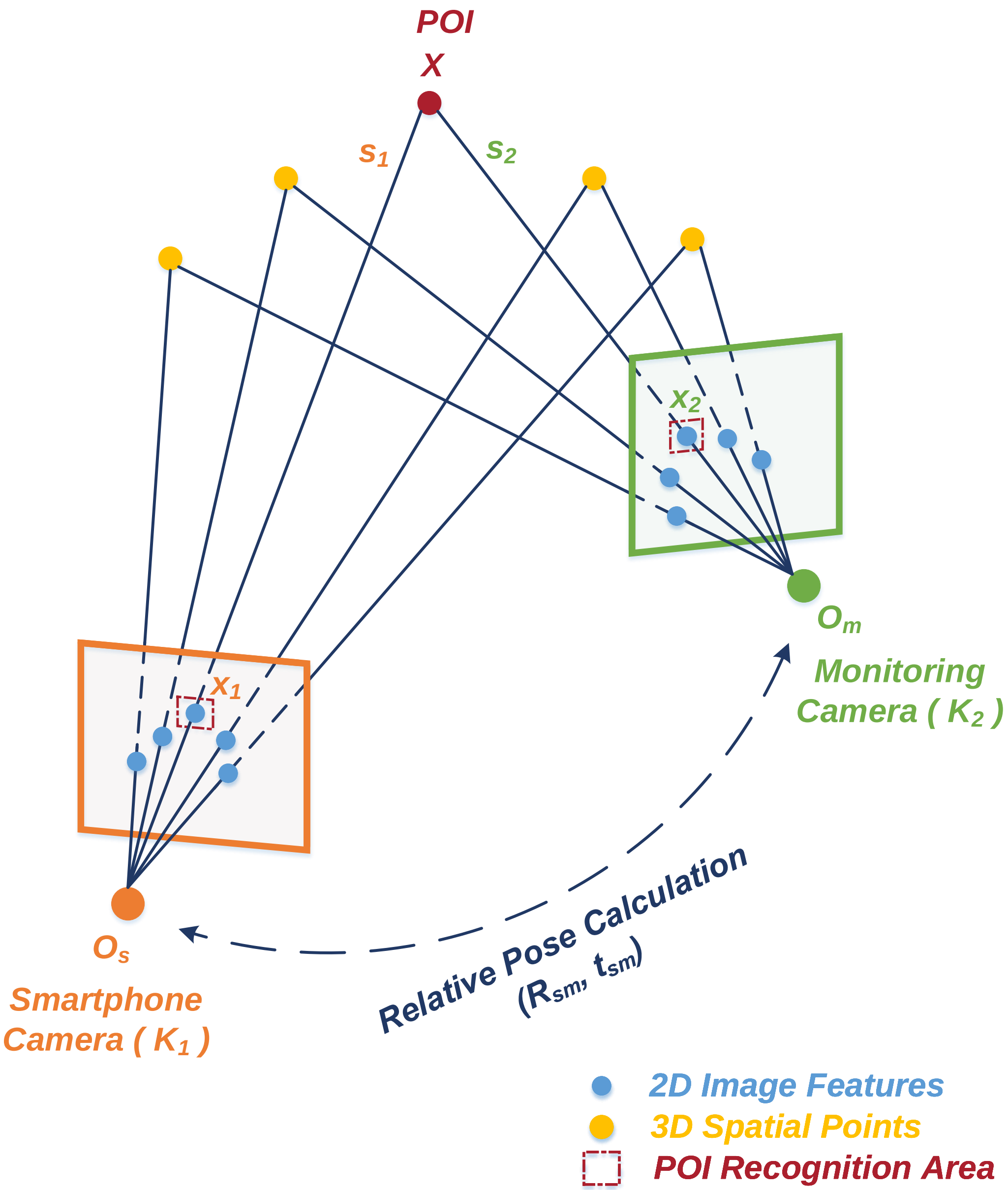


Figure 4.2: POI Recognition and Relative Pose Calculation

relationships of these points as the following equations:

$$\begin{aligned} s_1\boldsymbol{x}_1 &= \boldsymbol{K}_1\boldsymbol{X} \\ s_2\boldsymbol{x}_2 &= \boldsymbol{K}_2\left(\boldsymbol{R}_{sm}\boldsymbol{X}+\boldsymbol{t}_{sm}\right) \end{aligned} \tag{4.2.1}$$

where $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ represent the parameter matrices of the smartphone camera and monitoring camera correspondingly, s_1 and s_2 represent the pixel depths of the two-dimensional image features in the video frames acquired by the smartphone camera and the monitoring camera respectively, $\boldsymbol{t}_{sm}$ and $\boldsymbol{R}_{sm}$ represent the corresponding translation vector and rotation matrix between the mobile coordinate system and the monitoring camera coordinate system. After eliminating $\boldsymbol{X}$, Equation 4.2.1 can be expressed as follows:

$$\boldsymbol{X} = s_1\left(\boldsymbol{K}_1^{-1}\boldsymbol{x}_1\right) \tag{4.2.2}$$

$$s_2\left(\boldsymbol{K}_2^{-1}\boldsymbol{x}_2\right) = s_1\boldsymbol{R}_{sm}\left(\boldsymbol{K}_1^{-1}\boldsymbol{x}_1\right)+\boldsymbol{t}_{sm}. \tag{4.2.3}$$

Let $\boldsymbol{t}_{sm}^{\wedge}$ denote the skew-symmetric matrix corresponding to the translation vector $\boldsymbol{t}_{sm}$. Firstly, multiply this skew-symmetric matrix $\boldsymbol{t}_{sm}^{\wedge}$ on the left and right sides of Equation 4.2.3, which is equivalent to making the outer product of both sides with the translation vector $\boldsymbol{t}_{sm}$. This method is also applicable to other equations later in this thesis. Then, multiply both sides of the equation by the term $\left(\boldsymbol{K}_2^{-1}\boldsymbol{x}_2\right)^T$ at the same time. After elimination, we can have the following equations as the epipolar constraints:

$$\begin{gathered}\left(\boldsymbol{x}_2^T \boldsymbol{K}_2^{-T}\right) \boldsymbol{E} \left(\boldsymbol{K}_1^{-1} \boldsymbol{x}_1\right) = 0 \\ \boldsymbol{E} = \boldsymbol{t}_{sm}^{\wedge} \boldsymbol{R}_{sm}\end{gathered} \tag{4.2.4}$$

where $\boldsymbol{E}$ represents the Essential Matrix.

As described before, the image feature matching algorithm can normally acquire hundreds to one thousand two-dimensional feature point pairs, which can be provided to calculate Equation 4.2.4. In the implementation, our system firstly leverages the RANSAC method (Random Sample Consensus) (Derpanis, 2010) to solve the essential matrix $\boldsymbol{E}$, and further utilizes the SVD algorithm (Singular Value Decomposition) (Abdi, 2007) to decompose $\boldsymbol{E}$ and obtain the translation vector $\boldsymbol{t}_{sm}$ and rotation matrix $\boldsymbol{R}_{sm}$.

However, in the process of calculating $\boldsymbol{t}_{sm}$ and $\boldsymbol{R}_{sm}$, the pixel depths s_1 and s_2 of POI in the video frames from the cameras are eliminated. Therefore, our system needs to utilize the principle of triangulation (Taketomi *et al.*, 2017), so that we could recover the corresponding three-dimensional coordinates of POI in the mobile coordinate system on the basis of its two-dimensional projection on the camera imaging plane.

Suppose $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are the two-dimensional feature points of POI in the image frames of two cameras respectively, and $\boldsymbol{X}$ is the corresponding three-dimensional position of POI in the mobile coordinate system. Firstly, we can calculate s_1 by multiplying the skew-symmetric matrix $\left(\boldsymbol{K}_2^{-1} \boldsymbol{x}_2\right)^{\wedge}$, which is corresponding to the term $\boldsymbol{K}_2^{-1} \boldsymbol{x}_2$, on the left and right sides of Equation 4.2.3:

$$s_1 \left(\boldsymbol{K}_2^{-1} \boldsymbol{x}_2\right)^{\wedge} \boldsymbol{R}_{sm} \left(\boldsymbol{K}_1^{-1} \boldsymbol{x}_1\right) + \left(\boldsymbol{K}_2^{-1} \boldsymbol{x}_2\right)^{\wedge} \boldsymbol{t}_{sm} = 0. \tag{4.2.5}$$

With the calculated translation vector and rotation matrix, $\boldsymbol{t}_{sm}$ and $\boldsymbol{R}_{sm}$ respectively, Equation 4.2.5 is a linear equation with the only variable s_1, so that it can be easily solved.

Secondly, the method of calculating s_2 is similar. We can calculate s_2 by multiplying the skew-symmetric matrix $\left(\boldsymbol{K}_1^{-1}\boldsymbol{x}_1\right)^\wedge$, which is corresponding to the term $\boldsymbol{K}_1^{-1}\boldsymbol{x}_1$, on the left and right sides of Equation 4.2.3:

$$s_2\left(\boldsymbol{K}_1^{-1}\boldsymbol{x}_1\right)^\wedge\left(\boldsymbol{K}_2^{-1}\boldsymbol{x}_2\right) = \left(\boldsymbol{K}_1^{-1}\boldsymbol{x}_1\right)^\wedge\boldsymbol{t}_{sm}. \tag{4.2.6}$$

With $\boldsymbol{t}_{sm}$ and $\boldsymbol{R}_{sm}$, Equation 4.2.6 is also a linear equation with the only variable s_2, so it can be solved similarly.

By combining the pixel depths, s_1 and s_2, of POI in the video frames, with the translation vector $\boldsymbol{t}_{sm}$ between these two camera coordinate systems, the relative position relationships of the smartphone camera, monitoring camera and POI in the mobile coordinate system, that is, the geometry of $\triangle O_sO_mX$ in Figure 4.2, can be obtained.

4.2.3 Scale Obtainment

In Subsection 4.2.2, we have obtained the relative positions of the two cameras and POI in the mobile coordinate system, namely the geometry of $\triangle O_sO_mX$. However, as explained in Section 1.3, due to the scale ambiguity in monocular SLAM (Strasdat *et al.*, 2010), we still cannot determine the actual position of the smartphone camera. In other words, the three sides of $\triangle O_sO_mX$, s_1, s_2, and $\|\boldsymbol{t}_{sm}\|_2$ respectively, which we have solved in the previous section, are only the normalized lengths of $\left\|\overrightarrow{O_sX}\right\|_2$, $\left\|\overrightarrow{O_mX}\right\|_2$, and $\left\|\overrightarrow{O_sO_m}\right\|_2$ in the mobile coordinate system, rather than the actual lengths in the real scenarios. Thus, the physical distances between the two cameras and POI in the environment still remain unknown to us. Here, $\|\boldsymbol{t}_{sm}\|_2$ represents the $\boldsymbol{L}^2$ norm of the vector $\boldsymbol{t}_{sm}$, namely the Euclidean length of the vector $\boldsymbol{t}_{sm}$, which is also applicable to other vectors mentioned in this thesis.

But, according to the semantic information contained in the preloaded floor plan of the building, our system can attach the actual lengths in the real environment to $\triangle O_s O_m X$, that is, using meters as the physical units.

Suppose that in the real scenario, the physical distance between the POI and monitoring camera is l meters, namely the actual length of $\left\|\overrightarrow{O_m X}\right\|_2$ in the environment. Then we only need to make the ratio of the lengths of the same vector $\overrightarrow{O_m X}$ in two different measurement methods, l and s_2, to obtain the scale transformation ratio r, which is:

$$r = \frac{l}{s_2}. \tag{4.2.7}$$

This scale transformation ratio r states that the normalized length of a unit in the mobile coordinate system represents the actual length of r meters in the real environment.

4.3 Coordinate Conversion from 3D to 2D

All the locations and distances involved in Section 4.2 are discussed in the mobile coordinate system. To determine the actual position of the user holding the smartphone in the floor plan, it is necessary to obtain the conversion relationship from the mobile coordinate system (corresponding to the perspective of the smartphone camera) to

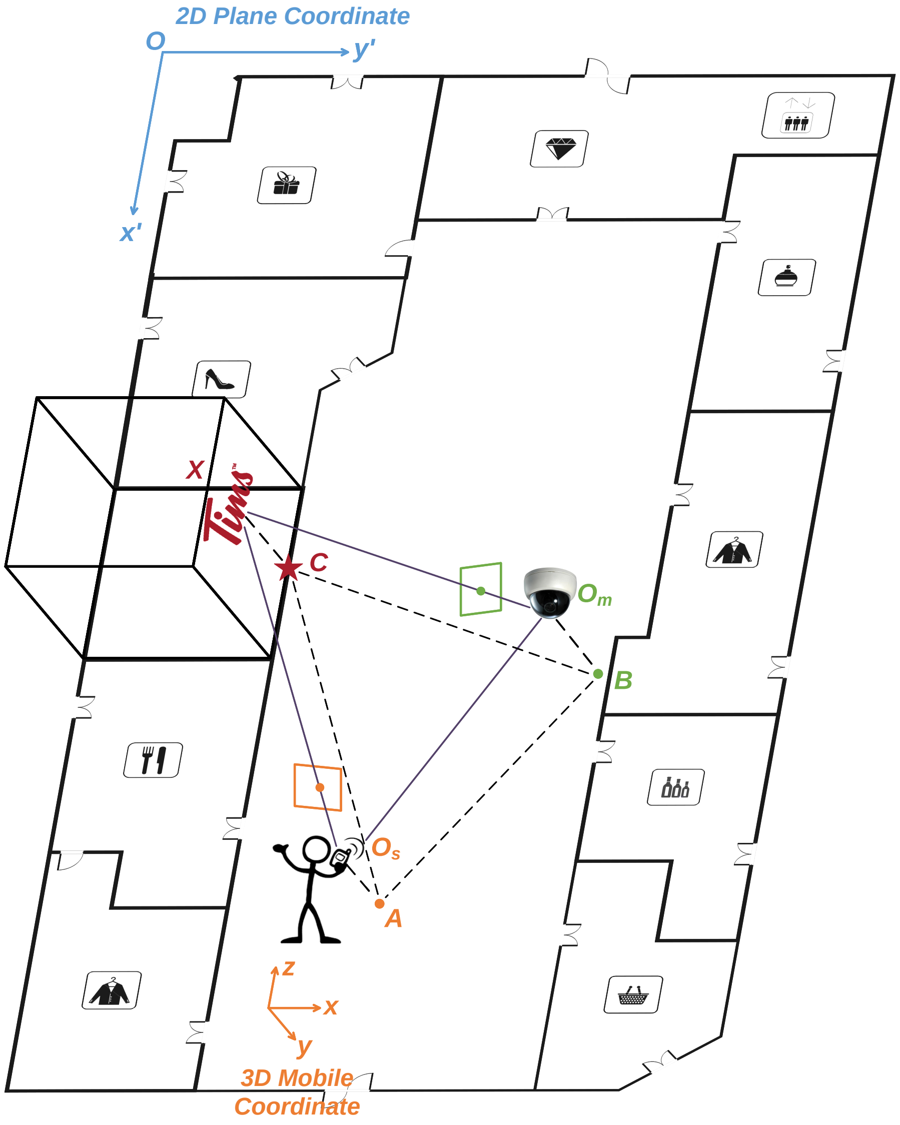


Figure 4.3: Coordinate Conversion from 3D to 2D in Application Scenarios

the plane coordinate system (corresponding to the floor plan of the building).

First of all, as is shown in Figure 4.3, the two coordinate systems that need to be converted have the following differences:

1. **Scales:** The mobile coordinate system utilizes the normalized scale, and this scale will change dramatically with diverse application scenarios. However, the plane coordinate system uses the actual scale in the real environment, such as meters.

2. **Dimensions:** The mobile coordinate system is a three-dimensional coordinate system. But the plane coordinate system is two-dimensional.

3. **Directions:** As depicted in Figure 4.3, the directions of the axes of the mobile coordinate system depend on the posture of the user holding the smartphone when the system is first started, while the directions of the axes of the plane coordinate system have been defined in advance according to the floor plan.

4. **Origins:** Similar to the directions, the origin of the mobile coordinate system is determined by the user's initial position when the system is just turned on, such as point O_s shown in Figure 4.3. In comparison, the origin of the plane coordinate system is normally predefined as a corner of the edge in the floor plan, such as point O shown in Figure 4.3.

The difference in scales between the plane coordinate system and the mobile coordinate system has been settled in Section 4.2. Our system still needs to resolve the differences in the other three aspects.

As shown in Figure 4.3, for the sake of introduction, point A represents the user's initial position in the floor plan when the system is first started. According to the semantic information preloaded by the system, B and C represent the corresponding two-dimensional positions of the monitoring camera and POI in the floor plan.

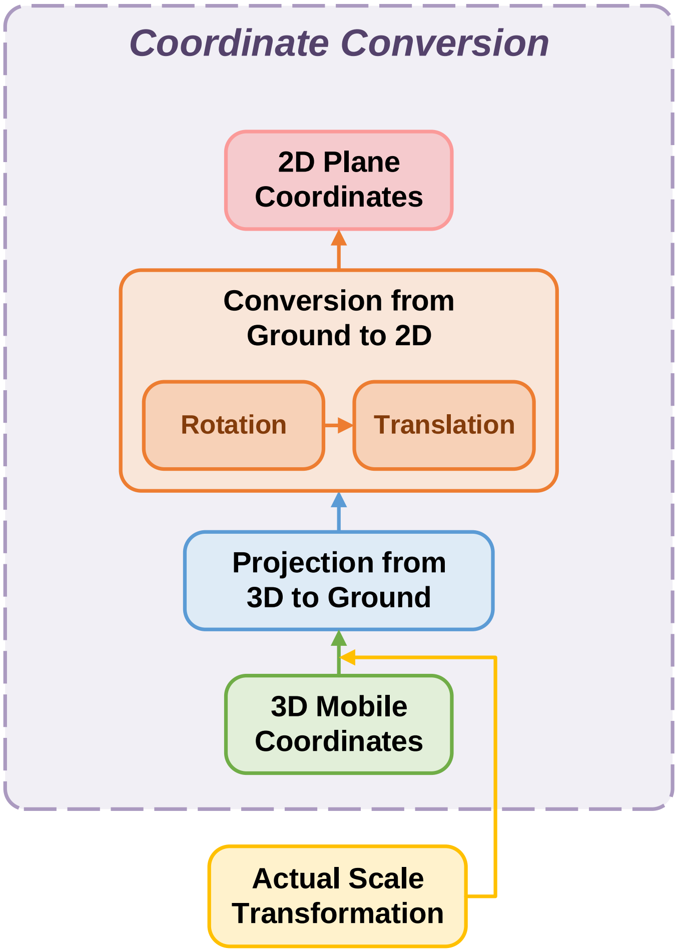

Coordinate Conversion
2D Plane Coordinates
Conversion from Ground to 2D
Rotation
Translation
Projection from 3D to Ground
3D Mobile Coordinates
Actual Scale Transformation

Figure 4.4: Framework of the Coordinate Conversion Algorithm

Figure 4.4 depicts the overall flow of the coordinate conversion algorithm. In order to realize the transformation from the mobile coordinate system in the smartphone camera's perspective to the plane coordinate system in the floor plan, in addition to the scale transformation introduced in Section 4.2, the system also needs the following two more steps:

Firstly, the system projects the three-dimensional mobile coordinates onto the horizontal ground. Secondly, after appropriate rotation and translation, the corresponding two-dimensional plane coordinates can be calculated.

The specific transformation methods will be explained in detail in this section.

4.3.1 Projection from 3D to Ground

In order to locate the user's initial position in the floor plan and continuously track the user's motion, the system needs to project the three-dimensional mobile coordinates onto the two-dimensional horizontal ground. As shown in Figure 4.3, to obtain such a projection relationship, when the system is started for the first time, the user is recommended to hold the smartphone in such a way that the y-axis of the mobile coordinate system is perpendicular to the horizontal ground. Therefore, the projection matrix from the mobile coordinate system to the horizontal ground and from three-dimensional to two-dimensional can be expressed as follows:

$$\boldsymbol{M}_p = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.3.1}$$

The projection matrix $\boldsymbol{M}_p$ eliminates the coordinate components of O_s, O_m and X which are perpendicular to the horizontal ground, the y-components, so as to project three dimensions into two dimensions.

In detail, based on the projection matrix, in order to project $\triangle O_s O_m X$ in the mobile coordinate system to the corresponding triangle $\triangle O'_s O'_m X'$ on the horizontal ground, our system obtains the side $\overrightarrow{O'_m X'}$ by the following formula:

$$\overrightarrow{O'_m X'} = \boldsymbol{M}_p \left(\overrightarrow{O_m X} \right). \tag{4.3.2}$$

In addition, one of the strengths of our system is that only during the coordinate conversion in the initialization stage, the user needs to take the smartphone in a posture where the y-axis of the mobile coordinate system is perpendicular to the horizontal ground. After connecting the three-dimensional and two-dimensional coordinate systems, that is, after the system locates the user's initial position, the user can take his smartphone casually, as long as the contents of the video frames acquired by the smartphone camera do not change dramatically.

Furthermore, even when the system performs the coordinate conversion, the user is allowed to take the smartphone in a slightly inclined posture. If in the initial positioning phase, the user turns the smartphone by 5°, then in the subsequent tracking phase, from $\cos(5^\circ) \approx 0.9962$, it can be obtained that every time the user advances $100m$, the system will only make a positioning error of $0.38m$, which is obviously tolerable.

Moreover, the relocation method of our system can also significantly reduce the projection error and correct the user's trajectory. Therefore, when POI can be captured by both the user's smartphone camera and the monitoring camera in the surrounding environment at the same time, the accumulated deviation caused by the imprecise projection can be directly corrected by the system.

4.3.2 Conversion from Ground to 2D

However, after scale transformation and projection, $r\boldsymbol{M}_p\left(\overrightarrow{O_mX}\right)$ and $\overrightarrow{BC}$ are still different. They are just two vectors with the same length, but they are in the different two-dimensional coordinate systems. Thus, the system needs to further calculate the 2×2 rotation matrix $\boldsymbol{R_f}$ and the 2×1 translation vector $\boldsymbol{t}_{OA}$ from the two-dimensional projection of the mobile coordinate system to the plane coordinate system. This process can be divided into the following two steps.

First of all, in order to calculate the rotation matrix $\boldsymbol{R}_f$, make the inner product of the vectors $r\boldsymbol{M}_p\left(\overrightarrow{O_mX}\right)$ and $\overrightarrow{BC}$, and denote the angle between these two vectors as θ, then we can get:

$$r\boldsymbol{M}_p\overrightarrow{O_mX}\cdot\overrightarrow{BC}=\left\|r\boldsymbol{M}_p\overrightarrow{O_mX}\right\|_2\|\overrightarrow{BC}\|_2\cos\theta=\|\overrightarrow{BC}\|_2^2\cos\theta \tag{4.3.3}$$

So:

$$\cos\theta=\frac{r}{\|\overrightarrow{BC}\|_2^2}\left(\boldsymbol{M}_p\overrightarrow{O_mX}\cdot\overrightarrow{BC}\right) \tag{4.3.4}$$

From $\cos^2\theta+\sin^2\theta=1$, we can also get:

$$\sin\theta=\sqrt{1-\cos^2\theta}=\sqrt{1-\frac{r^2}{\|\overrightarrow{BC}\|_2^4}\left(\boldsymbol{M}_p\overrightarrow{O_mX}\cdot\overrightarrow{BC}\right)^2} \tag{4.3.5}$$

According to the expression of the rotation matrix on the two-dimensional plane as follows:

$$R(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix} \tag{4.3.6}$$

the system can obtain the rotation matrix $\boldsymbol{R}_f$ of the conversion between the above different two-dimensional coordinate systems:

$$\boldsymbol{R}_f = \begin{bmatrix} \frac{r}{\|\overrightarrow{BC}\|_2^2}\left(\boldsymbol{M}_p\overrightarrow{O_mX}\cdot\overrightarrow{BC}\right) & -\sqrt{1-\frac{r^2}{\|\overrightarrow{BC}\|_2^4}\left(\boldsymbol{M}_p\overrightarrow{O_mX}\cdot\overrightarrow{BC}\right)^2} \\ \sqrt{1-\frac{r^2}{\|\overrightarrow{BC}\|_2^4}\left(\boldsymbol{M}_p\overrightarrow{O_mX}\cdot\overrightarrow{BC}\right)^2} & \frac{r}{\|\overrightarrow{BC}\|_2^2}\left(\boldsymbol{M}_p\overrightarrow{O_mX}\cdot\overrightarrow{BC}\right) \end{bmatrix} \tag{4.3.7}$$

Secondly, after the scale transformation, projection and rotation of the translation vector $\boldsymbol{t}_{sm}$, the system can get $\overrightarrow{AB} = \boldsymbol{t}_{AB}$ in the plane coordinate system as the following equation:

$$\overrightarrow{AB} = \boldsymbol{t}_{AB} = r\boldsymbol{R}_f\boldsymbol{M}_p\boldsymbol{t}_{sm}. \tag{4.3.8}$$

Since the monitoring camera's location in the floor plan, $\overrightarrow{OB} = \boldsymbol{t}_{OB}$, is known, and with $\overrightarrow{OA} = \overrightarrow{OB} - \overrightarrow{AB}$, we can obtain that when the system is started, the corresponding position of the origin of the three-dimensional mobile coordinate system in the two-dimensional floor plan, namely $\overrightarrow{OA} = \boldsymbol{t}_{OA}$, can be expressed as follows:

$$\overrightarrow{OA} = \boldsymbol{t}_{OA} = \boldsymbol{t}_{OB} - r\boldsymbol{R}_f\boldsymbol{M}_p\boldsymbol{t}_{sm} \tag{4.3.9}$$

where $\overrightarrow{OA} = \boldsymbol{t}_{OA}$ is just the starting point of the user in the floor plan when the system is turned on, such as position A in Figure 4.3. Therefore, our system has been able to successfully locate the user's initial position in the real environment.

In summary, according to the scale transformation ratio r, rotation matrix $\boldsymbol{R}_f$, projection matrix $\boldsymbol{M}_p$ and translation vector $\boldsymbol{t}_{OA}$, our system could convert any three-dimensional point in the mobile coordinate system to the corresponding two-dimensional point in the floor plan. In other words, the system is able to connect the three-dimensional mobile coordinate system with the two-dimensional plane coordinate system.

So far, the initialization stage of the system is completed.

Then, as the user moves forward, our system will also track and update the user's latest position, which will be further explained in the next chapter.

Chapter 5

Continuous Tracking and Navigation

According to the calculated results in Chapter 4, our system can locate the user's initial position in the two-dimensional floor plan. Based on this, the system will provide an appropriate navigation route for the user. Next, while the user is moving, our system will continue to track the user's real-time position, as well as give accurate and reliable navigation prompts.

This chapter will introduce in detail how our system tracks and navigates the user.

5.1 Real-Time VO Tracking

The positioning function can be divided into two steps: initial positioning and follow-up VO tracking.

For initial positioning, the system will obtain the user's initial position in the plane coordinate system based on Equation 4.3.9, such as position A in Figure 4.3.

For follow-up tracking in real-time, the relative pose calculation and triangulation method introduced in Subsection 4.2.2 can only be used in LOS (line-of-sight) scenes, where POI can appear in the view of the monitoring camera and smartphone camera at the same time. However, in NLOS (non-line-of-sight) scenes, our system cannot locate the user's current position only with the method mentioned above. Therefore, our system exploits the technique of VO (visual odometry) (Aqel *et al.*, 2016) to calculate the motion of the smartphone camera between adjacent video frames. The VO module of our system leverages the idea of PnP (perspective-n-point) problem (Hesch and Roumeliotis, 2011), which estimates the user's motion by matching the image features in different frames. Hence, our system can always track the user's current position in the mobile coordinate system.

Specifically, as introduced in Subsection 4.2.2, the system can obtain many point pairs by image matching and further triangulation. Each pair contains a two-dimensional pixel point $\boldsymbol{x}_i$ in the k_{th} video frame of the smartphone camera and its corresponding three-dimensional spatial point $\boldsymbol{X}_i$ in the mobile coordinate system.

Suppose that the conversion factors corresponding to the k_{th} video frame acquired by the smartphone camera are the rotation matrix $\boldsymbol{R}_k$ and the translation vector $\boldsymbol{t}_k$. Similar to Equation 4.2.1, by the imaging principle of pinhole cameras, we can get the following relationship:

$$s_i\boldsymbol{x}_i = \boldsymbol{K}_1\left(\boldsymbol{R}_k\boldsymbol{X}_i + \boldsymbol{t}_k\right) \tag{5.1.1}$$

where $\boldsymbol{K}_1$ still represents the parameter matrix of the smartphone camera and s_i represents the pixel depth of the two-dimensional image feature $\boldsymbol{x}_i$ in the k_{th} video frame acquired by the smartphone camera.

Then, the system calculates the rotation matrix $\boldsymbol{R}_k$ and the translation vector $\boldsymbol{t}_k$ by settling the following optimization problem:

$$\boldsymbol{R}_k, \boldsymbol{t}_k = \underset{\boldsymbol{R}_k, \boldsymbol{t}_k}{\arg\min} e = \underset{\boldsymbol{R}_k, \boldsymbol{t}_k}{\arg\min} \frac{1}{2} \sum_{i=1}^{n} \left\| \boldsymbol{x}_i - \frac{1}{s_i} \boldsymbol{K}_1 \left(\boldsymbol{R}_k \boldsymbol{X}_i + \boldsymbol{t}_k \right) \right\|_2^2 . \tag{5.1.2}$$

Here, our system utilizes the EPnP (Efficient-PnP) (Lepetit *et al.*, 2009) method, which is an algorithm with a complexity of $O(n)$, to solve this PnP problem. By minimizing the sum e of the error terms, the system can obtain the optimal solution of the rotation matrix $\boldsymbol{R}_k$ and the translation vector $\boldsymbol{t}_k$ corresponding to the k_{th} video frame, that is, the optimal estimation of the pose of the smartphone camera when shooting this video frame. Therefore, our system can obtain the three-dimensional position of the user corresponding to this frame in the mobile coordinate system, and then continuously record the user's motion trajectory.

After our system determines the user's initial position, the visual odometry function will be enabled, and with the user's movement, the corresponding translation vectors $\{\boldsymbol{t}_0, \boldsymbol{t}_1, \boldsymbol{t}_2, \ldots\}$ of the user in the three-dimensional mobile coordinate system can be gained in real-time.

Suppose that A_k is the corresponding two-dimensional position of the user in the floor plan when shooting the k_{th} image frame. To acquire the user's current position, with $\overrightarrow{OA_k} = \overrightarrow{OA} + \overrightarrow{AA_k}$, the system converts the corresponding translation vector $\boldsymbol{t}_k$ in the three-dimensional mobile coordinate system to the two-dimensional plane coordinate system:

$$\boldsymbol{t}_{OA_k} = \boldsymbol{t}_{OA} + \boldsymbol{t}_{AA_k} = \boldsymbol{t}_{OA} + r\boldsymbol{R}_f \boldsymbol{M}_p \boldsymbol{t}_k . \tag{5.1.3}$$

In this way, our system can track the user's position in the plane coordinate system in real-time.

In summary, through the relative pose calculation method in the initialization stage and the visual odometry function in the follow-up tracking stage, our system can always perform high-precision positioning of the user in the two-dimensional floor plan.

5.2 Path Planning and Navigation Strategy

The accurate and continuous tracking of the user has been explained in Section 5.1. In this section, we will mainly introduce how the system plans the optimal navigation path, and further provides the user with correct navigation prompts in real-time based on the current position information.

In the initialization stage, our system will first load the floor plan of the building and obtain the rich semantic information contained in it, such as a series of landmarks in the environment with their respective names and whether they are accessible to each other. Thus, the system can draw a map $G =< V, E >$ which shows the connectivity of the landmarks. Furthermore, the position of the i_{th} landmark is noted as the node $v_i \in V$, and the actual distance between the two landmarks, v_i and v_j, is noted as the length of the edge $e_{ij} \in E$. Then, the system will utilize the Dijkstra method (Broumi *et al.*, 2016) to calculate and select the shortest path for each node in the graph, and also record the obtained results.

First of all, when the user activates the navigation function, by determining the initial position, the system will note the landmark with the shortest distance as the user's starting point. Secondly, according to the starting position and target position, the system will plan the optimal route and provide the initial navigation prompts for the user. Then, while the user is continuously moving forward, our system will always track the user's current position, and on this basis guide the user to the subsequent landmark on the navigation path. Therefore, the user can finally arrive at the right destination efficiently.

One of the highlights of our system is the use of the relocation technique. Especially, when the navigation path is longer, this technique is more conducive to obtaining successful navigation services. When the user moves forward, as described in Subsection 4.2.1, our system will always detect and recognize the POI in the environment. When the POI in the scene can be captured by the user's smartphone camera, the system will determine the user's real-time position with the assistance of the surrounding monitoring camera again to eliminate the accumulated deviation and correct the previous trajectory drift. Therefore, the success rate of navigation can be significantly improved through the relocation technique.

In summary, with the techniques introduced in Chapter 5, as the user continues moving on, our system can always not only accurately locate and track the real-time position of the user but also provide correct and reliable navigation services for the user.

Chapter 6

System Implementation and Experimental Evaluation

6.1 Experimental Validation

In this thesis, the indoor positioning and navigation application is implemented on the Ubuntu operating system. As the input of the system, the videos are recorded in different scenarios with different types of cameras. This section will first introduce the experimental setup and methods.

6.1.1 Experimental Settings

In order to test the effect of the video quality on indoor positioning and navigation, the frame rates of the recorded videos include 30FPS and 60FPS, and the resolutions include 1920 × 1080 pixels (1080p) and 1280 × 720 pixels (720p). The computer used for calculation and processing has an i7-9750H CPU with 16G RAM, and the operating system is Ubuntu 16.04. The designed positioning and navigation system is modified and implemented on the basis of ORB-SLAM (Mur-Artal and Tardós, 2017), which is exploited as the visual odometry module.

6.1.2 Experimental Environments

Extensive experiments have been conducted in three representative public places, including an educational building, an apartment building and a supermarket. These areas have different interior layouts, and the changes in people flow in these scenarios are not the same. For example, the number of residents in the apartment building at night is much more than that in the daytime. In contrast, the educational building is occupied by more people during the daytime than at night. Generally speaking, among these three scenarios, the supermarket has the highest foot traffic, while there are much fewer people in the educational building. In view of these characteristics, the arrangements of data collection in each scenario are planned accordingly, such as the time, duration, coverage and so on.

In addition, in the process of gathering experimental data, various types of smartphones were used, including the iPhone 6, iPhone 7 Plus and iPhone XR. The camera parameters (such as the focal lengths, distortion coefficients, optical center positions, etc.) of these smartphones are different.

6.1.3 Evaluation Metrics

During the experiments, to collect data, the recorders used different postures to take the smartphones and shot at different heights. The performance of the indoor positioning and navigation system will be evaluated from the following two aspects:

positioning accuracy and successful navigation rate.

- **Positioning Accuracy:** The positioning performance of the system is mainly evaluated in the initialization stage. The user can utilize the smartphone camera to take pictures containing any POIs within sight of the monitoring cameras. Then, the system will combine the image data from the monitoring camera and smartphone camera to locate the user's current position.

  In order to obtain the ground truth of locations for the comparison of positioning results, all the queried locations are later manually measured and recorded in all experimental scenarios. Thus, the value of positioning accuracy is the distance error between the positioning result estimated by the system and the user's real location.

- **Successful Navigation Rate:** The semantic information needed for navigation has been defined and included in the preloaded floor plan. For each navigation simulation, the starting position and destination are randomly chosen on the floor plan. Then the system will plan the navigation path intelligently for the user. On each planned path, some specific landmarks are selected as measurement points, such as turns, elevators, etc.

  Successful navigation means that the system can guide the user to each measurement point in order correctly until the destination is reached. Otherwise, this navigation is failed. In addition, the result of each navigation simulation, which is success or failure, will be recorded accordingly. Therefore, the successful navigation rate is the percentage of successful navigations in all simulated navigation experiments.

6.1.4 Comparative Systems

In order to further evaluate the performance of the designed system, it is compared with four state-of-the-art indoor positioning and navigation systems in the experimental part. These systems are also mainly realized on the basis of smartphone cameras. In terms of positioning, this thesis selects HAIL, MVG and JVWL for comparison. With regard to navigation, this thesis compares the system with Travi-Navi.

1. **HAIL** (Niu *et al.*, 2019): HAIL is an automated indoor positioning algorithm mainly based on images. By memorizing only representative appearance features of landmarks, the system reduces the resource consumption of calculation and storage. Furthermore, the k-d tree method is leveraged to filter out the images that are matched incorrectly, enabling users to get rid of the trouble of extra operations.

2. **MVG** (Liu *et al.*, 2017): MVG proposes a model based on multi-perspective to obtain robust indoor positioning performance, utilizing visual data and geomagnetic signals from smartphones at the same time. Besides, local features and global information are combined to distinguish different locations.

3. **JVWL** (Liu *et al.*, 2016): Combining visual positioning and wireless positioning, JVWL is an advanced indoor tracking scheme. This system integrates the data from mobile cameras and Wi-Fi signals to perform positioning and exploits deep neural networks to further optimize the results, achieving much better performance.

4. **Travi-Navi** (Zheng *et al.*, 2017): Travi-Navi is a P2P indoor navigation system that can provide visual guidance. The image information and sensor data

generated during the guide's movement will be collected. Therefore, the system can offer image prompts and necessary alerts to followers on the navigation path without deploying indoor positioning in advance.

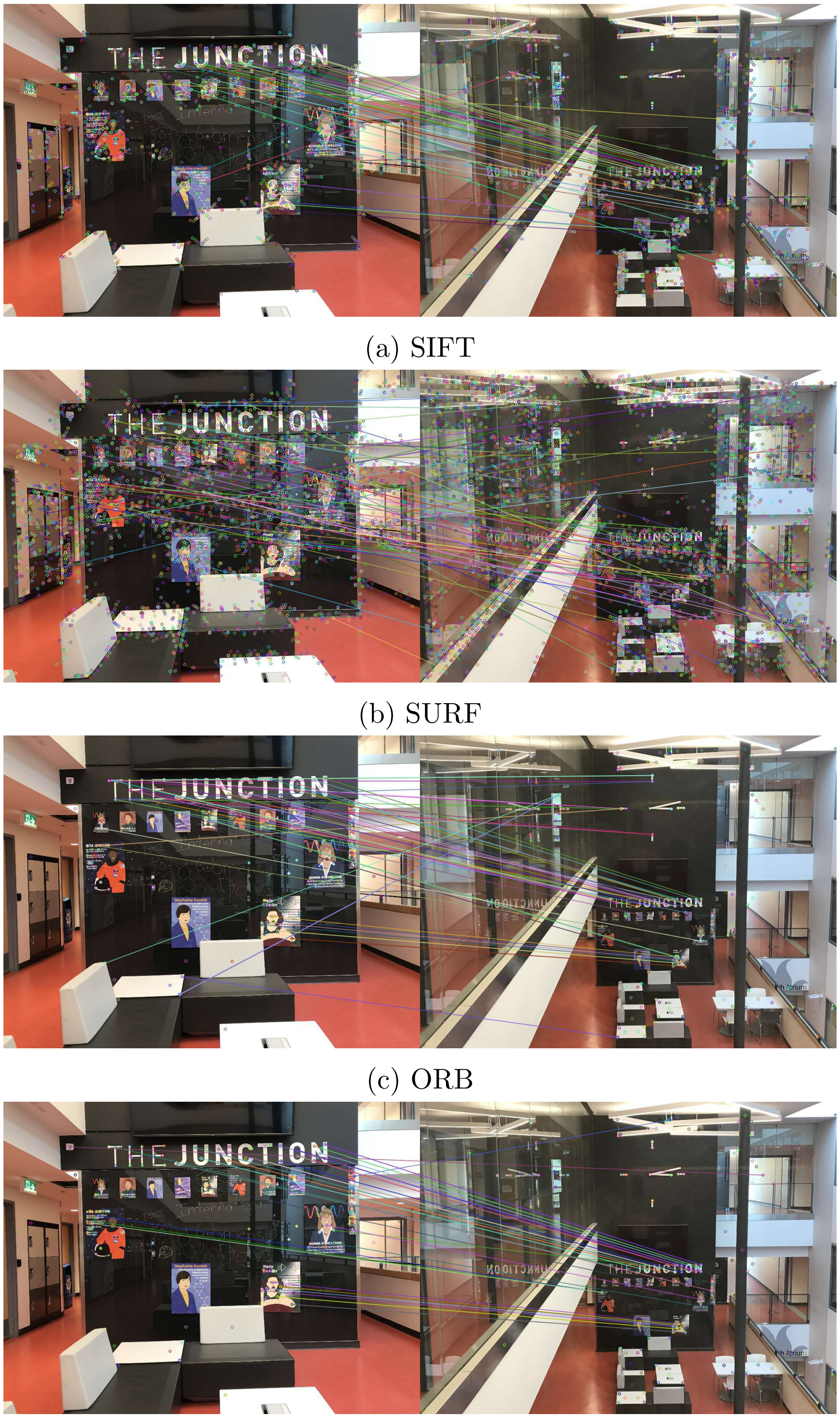
6.2 Experimental Results and Analysis

6.2.1 Feature Points Comparison

As introduced in Section 4.1, extracting and matching feature points of the image frames from the smartphone camera and monitoring camera plays a vital role in determining the user's actual initial position, which has a significant impact on the follow-up process of the system.

In order to compare the effect of four types of image feature points, SIFT, SURF, ORB and A-KAZE, extensive experiments have been carried out. As shown in Figure 6.1, Figure 6.2 and Figure 6.3, here we take these three groups of images as examples of the experimental results of feature extraction and matching. The three pairs of raw video frames were captured in three buildings on campus, from the perspectives of the smartphone camera and monitoring camera respectively. Among them, the left image frame of each group is obtained by the smartphone camera, while the right one is taken from the perspective of the monitoring camera. Then, SIFT, SURF, ORB and A-KAZE feature points are extracted and matched in each group of images.

It can be seen that, compared with SIFT, SURF and ORB, in all these three scenarios, the A-KAZE feature points from the image frames acquire the best matching performance. In particular, for the first group of frames, the angles of view of



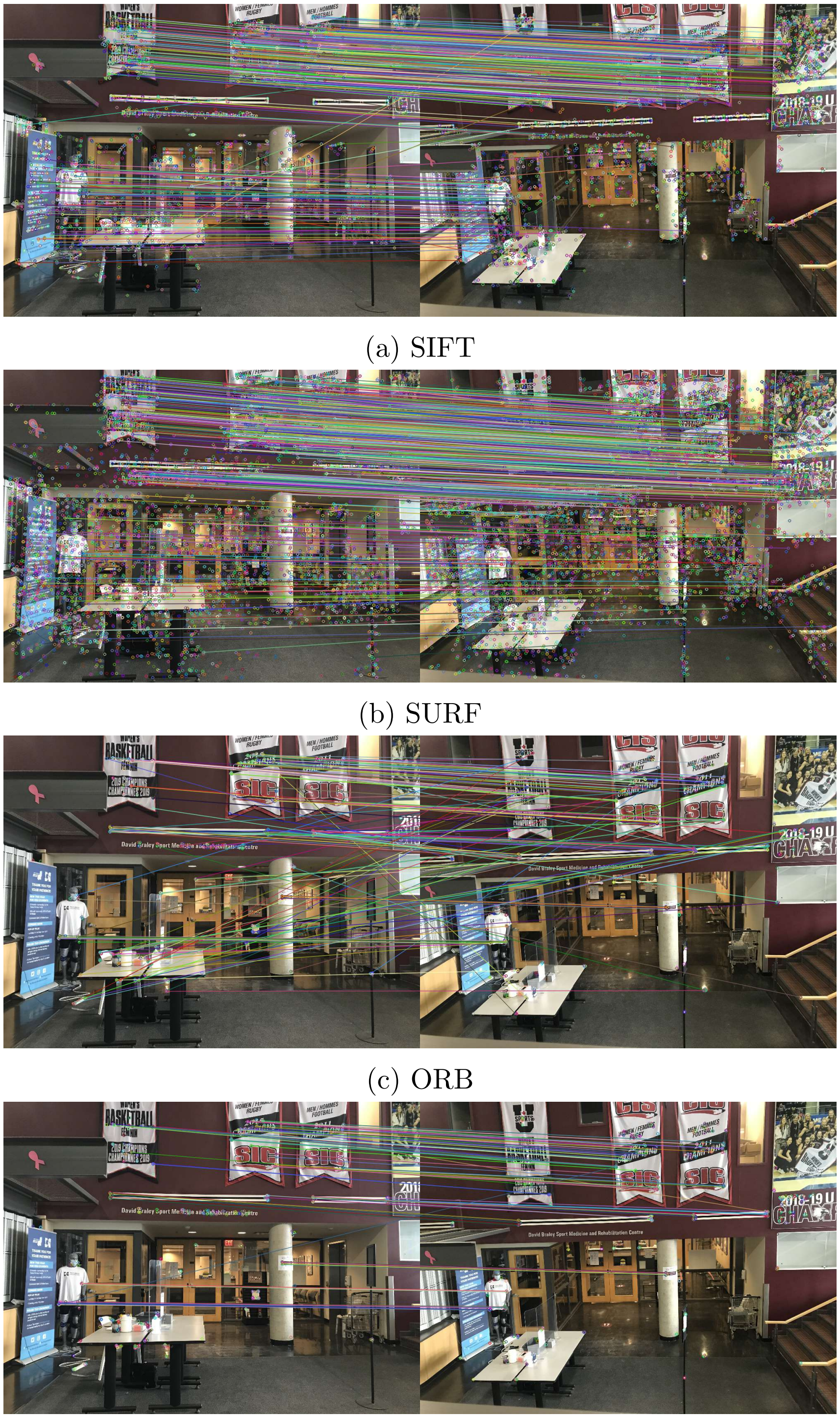
(a) SIFT

(b) SURF

(c) ORB

(d) A-KAZE

Figure 6.1: Feature Matching Comparison in Building 1



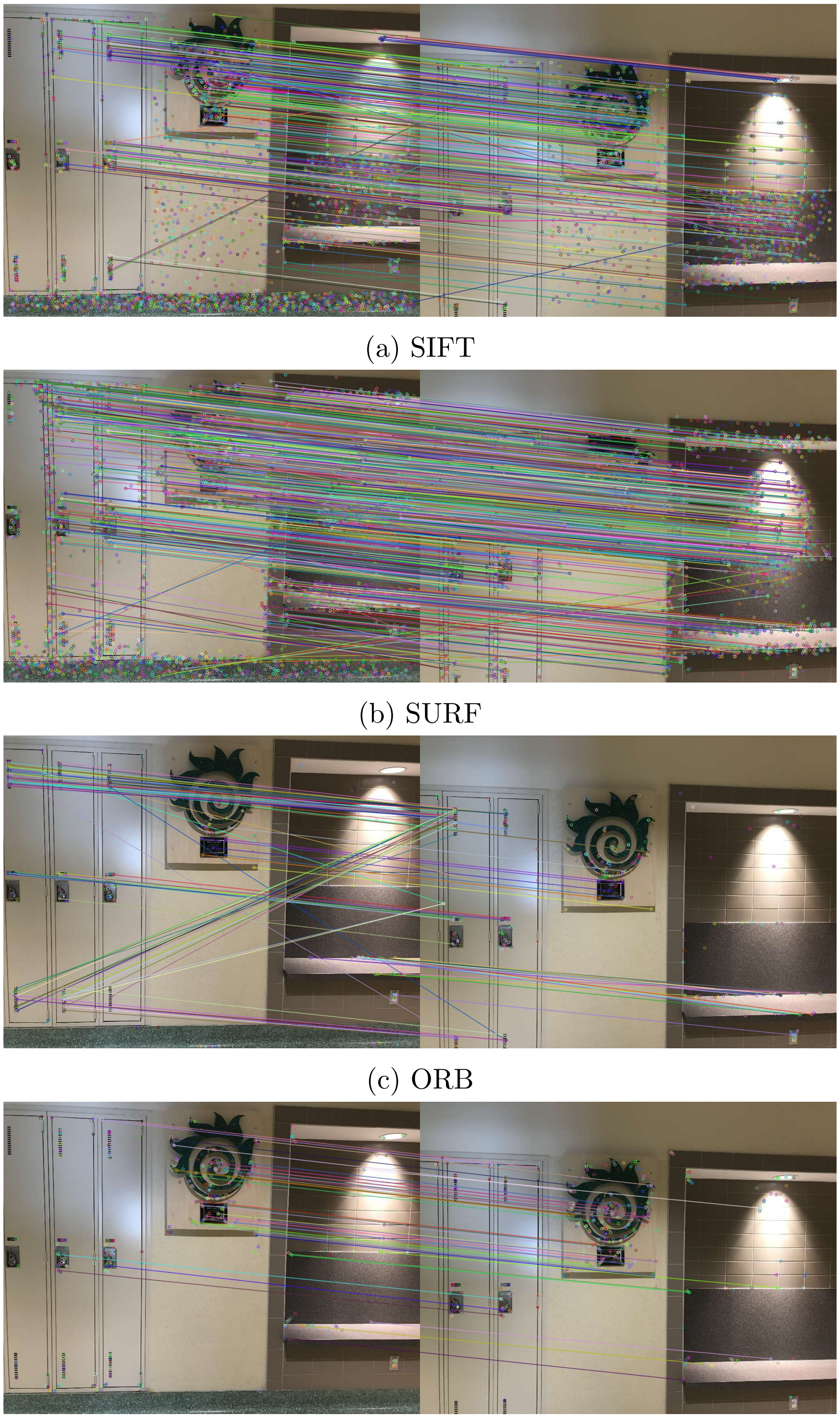
(a) SIFT

(b) SURF

(c) ORB

(d) A-KAZE

Figure 6.2: Feature Matching Comparison in Building 2



(a) SIFT

(b) SURF

(c) ORB

(d) A-KAZE

Figure 6.3: Feature Matching Comparison in Building 3

the two cameras are quite different, and the third group has many similar feature points which are hard to distinguish, but the A-KAZE algorithm still gains satisfactory matching results. It can be concluded that, in comparison with the other three methods, A-KAZE can achieve more stable and effective performance when processing images with different viewing angles and similar features, which is very suitable for our system.

Moreover, the experiments can also verify that, through the modification of the feature extraction and matching algorithm in OpenCV, our system can successfully handle the frames captured by cameras with different parameters.

Scenarios Features	Building 1	Building 2	Building 3
SIFT	86.9%	82.1%	86.5%
SURF	79.8%	83.3%	79.4%
ORB	76.5%	71.6%	74.3%
A-KAZE	**94.1%**	**93.7%**	**95.2%**

Table 6.1: Comparison of Correct Matching Rates of Four Feature Points

Furthermore, as shown in Table 6.1, the average correct matching rates of SIFT, SURF, ORB and A-KAZE in these three buildings are tested. Here, the correct matching rate refers to the percentage of correct matches among all matched feature point pairs between two video frames.

It can be concluded that, in the above three buildings, the average correct matching rates of image features are 94.3% of A-KAZE, 85.2% of SIFT, 80.8% of SURF and only 74.1% of ORB correspondingly. Compared with A-KAZE, the feature matching accuracy of SIFT, SURF and ORB are reduced by 9.2%, 13.5% and 20.2% respectively. The experimental data prove that A-KAZE obtains the best performance among all four feature points, with high accuracy and strong robustness. Therefore, our system also leverages the image feature extraction and matching algorithm of A-KAZE.

6.2.2 Influence of the Relative Pose of POIs and Two Cameras

As is introduced in Subsection 4.2.2, the accuracy of relative pose calculation and triangulation is essential for the whole positioning and navigation system. Therefore, in this subsection, we mainly focus on the influence of the relative pose of the POI, smartphone camera and monitoring camera on the positioning performance. In the experiment, it is assumed that the orientation and position of the POI and monitoring camera are fixed. By moving the smartphone camera, the relative distance between the POI and smartphone camera and the angle formed by all these three objects are changed. So that we can analyze the influence of different relative poses on the subsequent positioning results.

First of all, as is shown in Figure 6.4, when the distance between the POI and smartphone camera is fixed at $5m$, the influence of different angles on the positioning results is tested. When the angles formed by the smartphone camera, POI and monitoring camera are set at different values, the means of positioning accuracies of our system are $0.68m$ at 30°, $1.05m$ at 45° and $1.86m$ at 60° respectively. The result shows that the positioning error becomes larger with the increase of the angle formed by the three. The reason is that when the angle is larger, the difference between

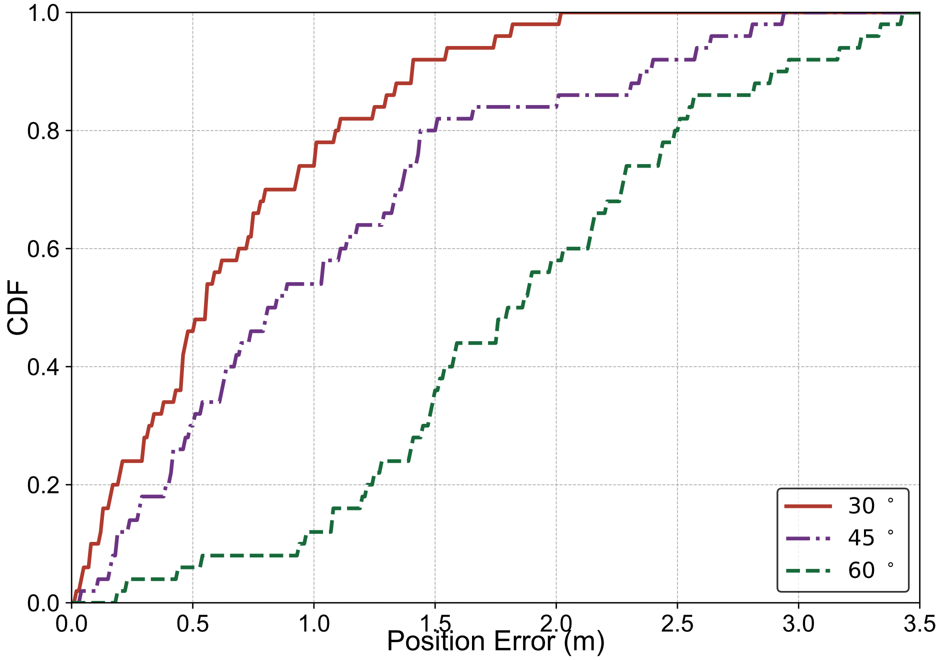


Figure 6.4: Influence of Different Angles on Positioning Accuracy

the video frames captured by the smartphone camera and monitoring camera will also increase significantly. This can bring more difficulties for the subsequent image registration, leading to the decline of the feature matching accuracy, and then affect the positioning performance.

In addition, with the above three angles, the corresponding variances of positioning errors are 0.26 at 30°, 0.59 at 45° and 0.61 at 60°. It can be seen that the increase of the angle will also lead to a significant increase in the variance of the positioning error. That is to say, a too large angle will add to the uncertainty of the positioning results and make the positioning function more unstable. Therefore, when using the system, it is suggested that the angle between the smartphone camera, POI and monitoring camera should not exceed 45°.

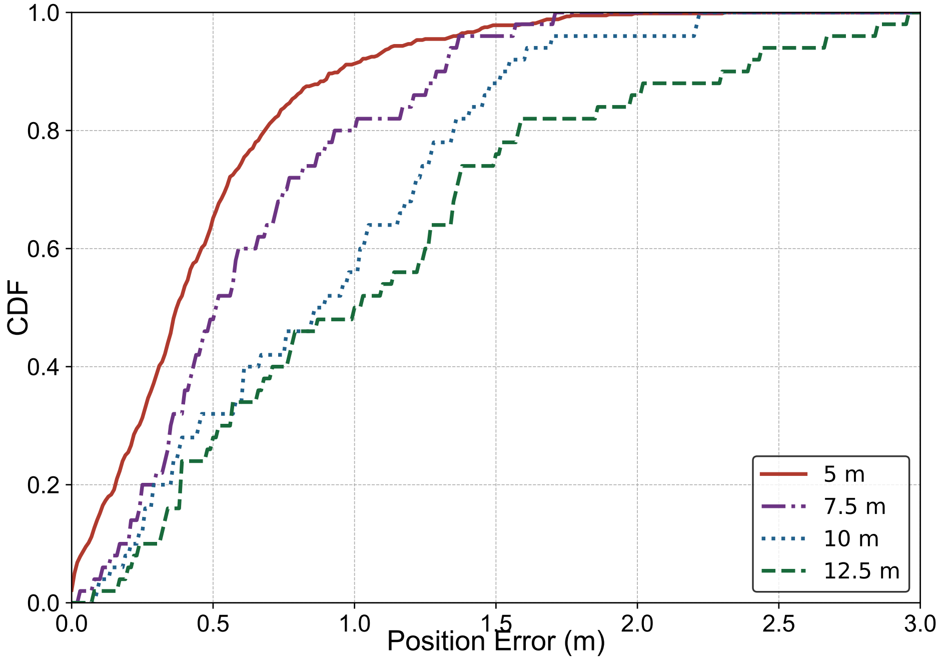


Figure 6.5: Influence of Different Distances on Positioning Accuracy

Secondly, as is shown in Figure 6.5, with the angle of the smartphone camera, POI and monitoring camera fixed to 30°, the influence of different distances on the positioning results is tested. When the distances between the POI and smartphone camera are $5m$, $7.5m$, $10m$ and $12.5m$, the means of positioning errors of our system are $0.46m$, $0.64m$, $0.89m$ and $1.11m$ respectively. Moreover, the $90th$ percentile positioning errors of the above four distances are $0.95m$, $1.30m$, $1.53m$ and $2.32m$ correspondingly. The experimental results are in accordance with the expectation. This is because, with the increase of the distance between the POI and smartphone camera, the depth estimation of POI will become more difficult, which will affect the calculation of the relative distance in triangulation, and ultimately reduce the positioning accuracy.

6.2.3 Influence of Video Quality

Because our indoor positioning and navigation system is mainly based on vision, the quality of videos as data inputs is of vital importance to the whole system. The experiments in this subsection are designed to explore the impacts of two essential factors, namely the frame rate and resolution, which determine the video quality. Four different settings of the frame rate and resolution are adopted to observe the experimental results, which are 1080p 60FPS, 1080p 30FPS, 720p 60FPS and 720p 30FPS.

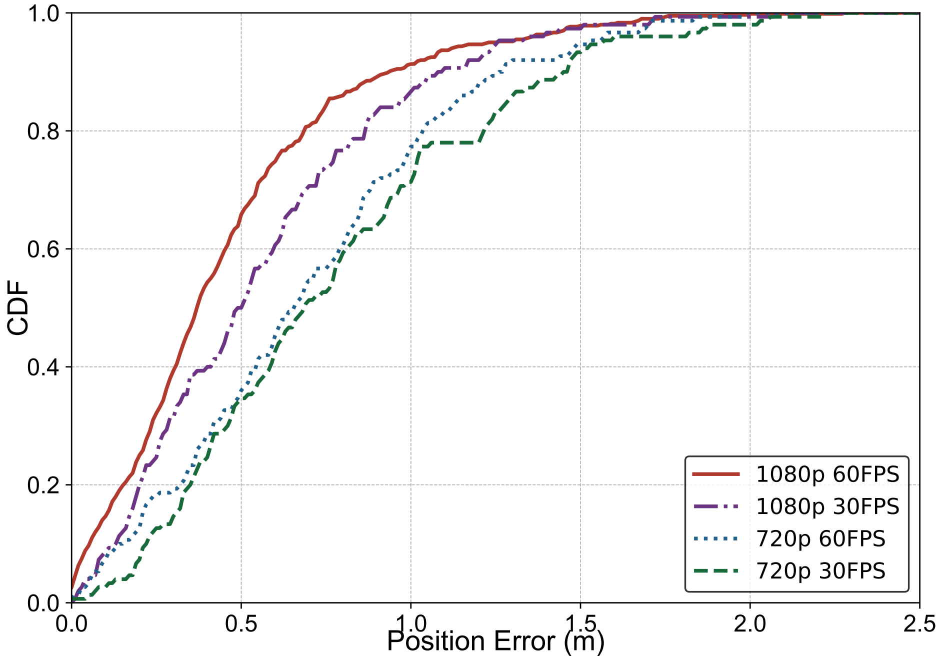


Figure 6.6: Influence of Different Video Quality on Positioning Accuracy

The results of the positioning experiment are shown in Figure 6.6. With the four types of video quality, 1080p 60FPS, 1080p 30FPS, 720p 60FPS and 720p 30FPS, the corresponding means of positioning errors of our system are 0.46m, 0.56m, 0.71m and 0.78m respectively. When the smartphone camera adopts the same video frame rate, the mean of positioning errors of 720p resolution is 0.74m, while the mean of 1080p resolution is reduced by 29.1% to 0.53m. Moreover, with the same resolution, the system error of 60FPS frame rate is 11.2% less than that of 30FPS, which are 0.60m and 0.67m respectively.

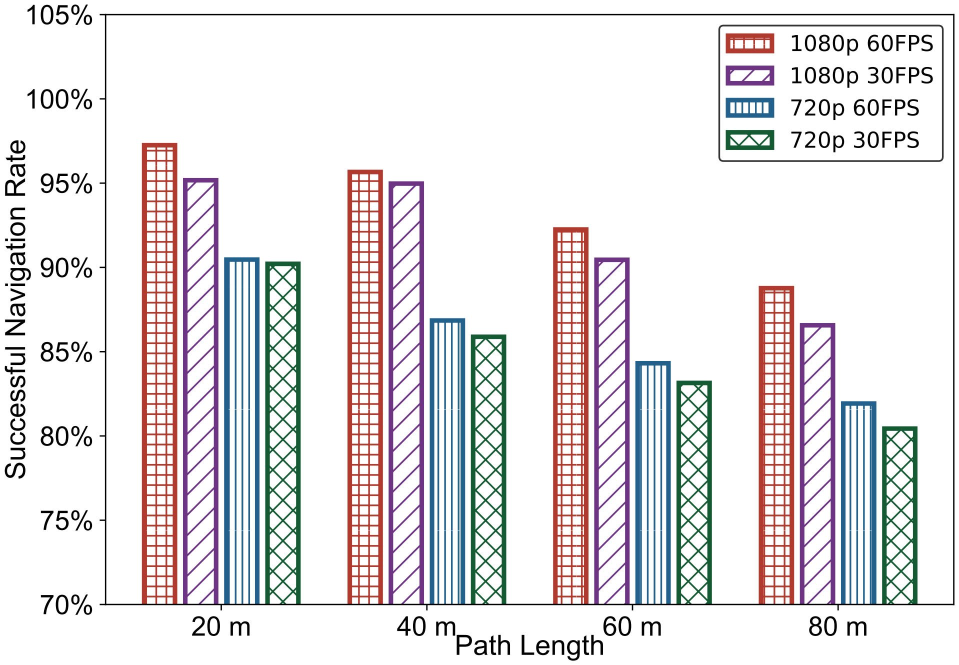


Figure 6.7: Influence of Different Video Quality on Successful Navigation Rate

Then, in the navigation experiment, the above four kinds of video quality of the smartphone camera are tested on different-length paths of 20m, 40m, 60m and 80m respectively. The navigation results are shown in Figure 6.7.

As introduced in Subsection 6.1.3, successful navigation refers to that, according to the starting point and target location, the system can guide the user to each landmark, which is selected and detected as the experimental settings, on the planned route sequentially until the destination is reached. Otherwise, this is recorded as a failed navigation. The successful navigation rate, which is used to measure the system performance, is the percentage of successful navigations in all simulated navigation experiments.

Overall, using different video quality of 1080p 60FPS, 1080p 30FPS, 720p 60FPS and 720p 30FPS on all above paths, the means of successful navigation rates are 93.3%, 90.8%, 87.5% and 84.4% respectively. An obvious contrast is that, on the $20m$, $40m$, $60m$ and $80m$ paths, the means of successful navigation rates of 1080p resolution outperform 720p resolution by 5.9%, 9.0%, 7.6%, and 6.5% respectively.

The experimental results indicate that the frame rate and resolution of videos recorded by the smartphone camera have a significant impact on the performance of our indoor positioning and navigation system. The frame rate of the smartphone camera will affect the blur degree of the video, so the high frame rate video can reduce the error of positioning and tracking. From the aspect of resolution, more feature points can be extracted from the higher resolution image, which helps to improve the accuracy of image matching between the smartphone camera and monitoring camera. In addition, during the user's movement, with higher resolution, the system can estimate the pose changes of the smartphone camera between adjacent image frames more accurately in real-time. Moreover, from the test results, we can obtain that the resolution of videos from the smartphone camera has a more remarkable impact on the system performance than the frame rate, and the high-definition video is conducive to providing more stable indoor positioning and navigation services.

6.2.4 Influence of Relocation

As mentioned in Chapter 3, in the process of positioning and navigation, the system can leverage the relocation function to reduce the possibility of drifts, so as to reduce or even eliminate the cumulative error. In order to verify the function of the relocation module, experiments are designed to compare the navigation performance of the system when the relocation function is turned on and off.

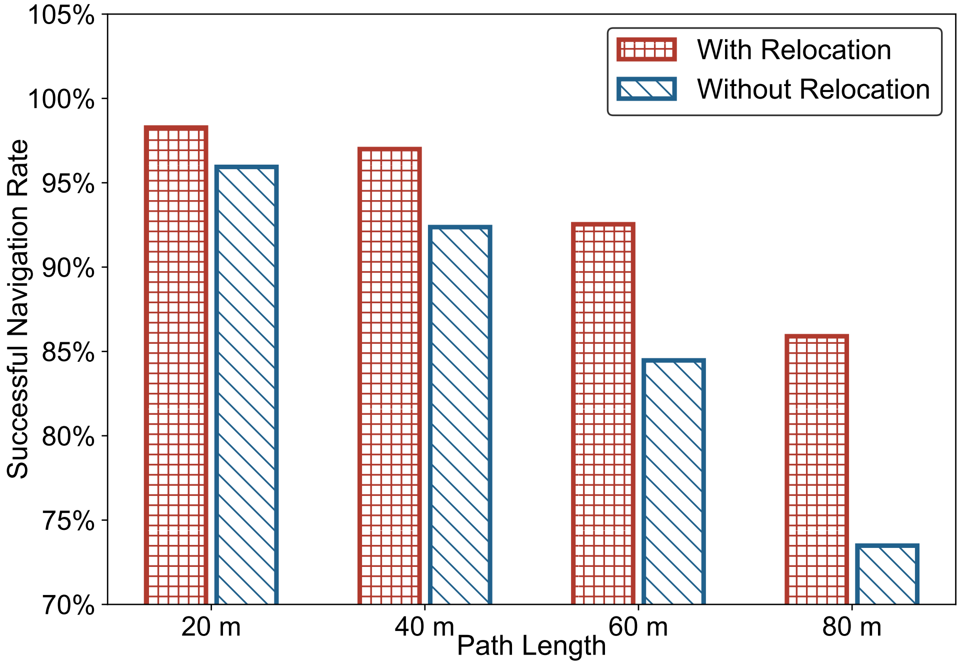


Figure 6.8: Validation of Relocation Function

In the experiment, four different-length paths in the educational building are chosen to evaluate the effect of relocation. The navigation results are shown in Figure 6.8. Turning off the relocation function means that, only in the initialization stage, the system uses the monitoring camera and smartphone camera in the mean time to capture the POI. While in the subsequent navigation process, other POIs and monitoring cameras are not used, only the smartphone camera is leveraged to continuously locate and track the user.

It can be seen from the histogram that the navigation performance of the system when using the relocation function is obviously superior to that when the relocation function is turned off. On the whole, on all paths, the use of relocation remarkably improves the average successful navigation rate by 7.9% to 93.4%. Specifically, the successful navigation rate of the system is increased by 2.4%, 5.1%, 9.6% and 16.9% respectively on the four navigation paths of $20m$, $40m$, $60m$ and $80m$. This reveals that as the path becomes longer, the effect of relocation on the improvement of navigation service is more outstanding. The experimental results are in accordance with the theoretical principle introduced before. Longer navigation paths usually result in greater drifts. But the relocation function can eliminate the previous accumulated error effectively and relocate the smartphone camera to the correct position and trajectory, so as to optimize the navigation performance.

Moreover, the experimental results demonstrate that, even without using the relocation module, our system can still achieve relatively high successful navigation rates of 96.1% and 92.4% on the $20m$ and $40m$ navigation paths correspondingly, which both have exceeded 90%. This verifies the robustness of our indoor positioning and navigation system. Even if merely in the initial stage, the monitoring camera and smartphone camera can capture the POI at the same time, but in the subsequent navigation process, only the smartphone camera is used for tracking the user due to the occlusion of obstacles or the blurred video frames, our system can still provide the relatively stable and satisfying navigation service.

6.2.5 Influence of Different Scenarios

As described in Subsection 6.1.2, the positioning and navigation experiments are conducted in an apartment building, an educational building and a supermarket, and the performances of the system in different indoor scenarios are compared.

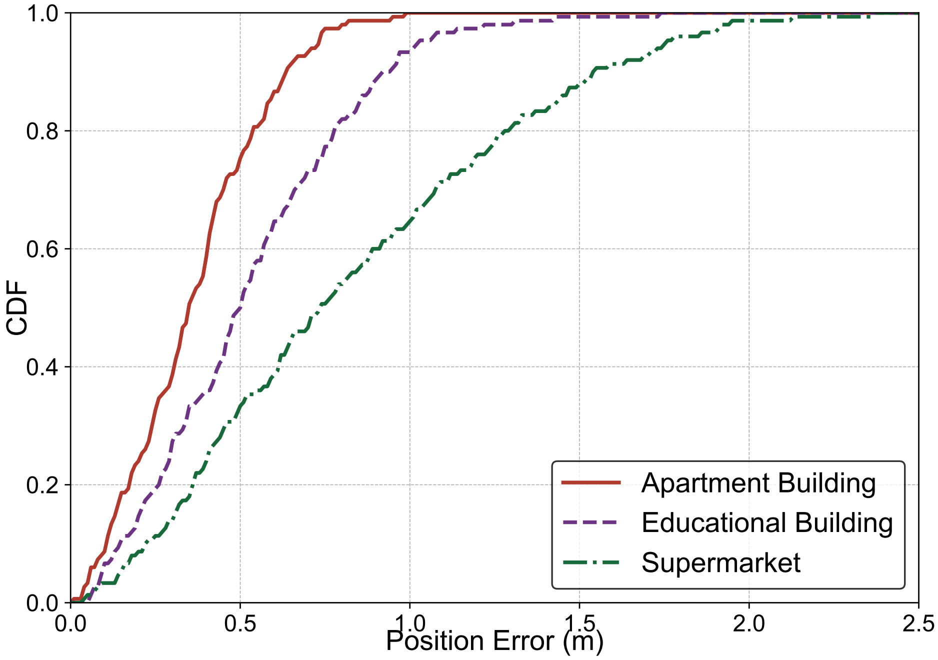


Figure 6.9: Comparison of Positioning Accuracy in Different Scenarios

The result of positioning is shown in Figure 6.9. It can be obtained that the means of positioning errors are 0.37m in the apartment building, 0.54m in the educational building and 0.84m in the supermarket. Moreover, correspondingly, the 90th percentile positioning errors in these three environments are 0.64m, 0.93m and 1.55m.

This reveals that our system can achieve the positioning accuracy of less than one meter or around one meter in all three experimental areas, regardless of the differences in the indoor environments. In addition, in the above three different buildings, the variances of positioning errors are 0.04, 0.09 and 0.26 respectively, which are all less than 0.3. This clarifies that our system can provide users with a relatively stable positioning experience in all three indoor scenarios.

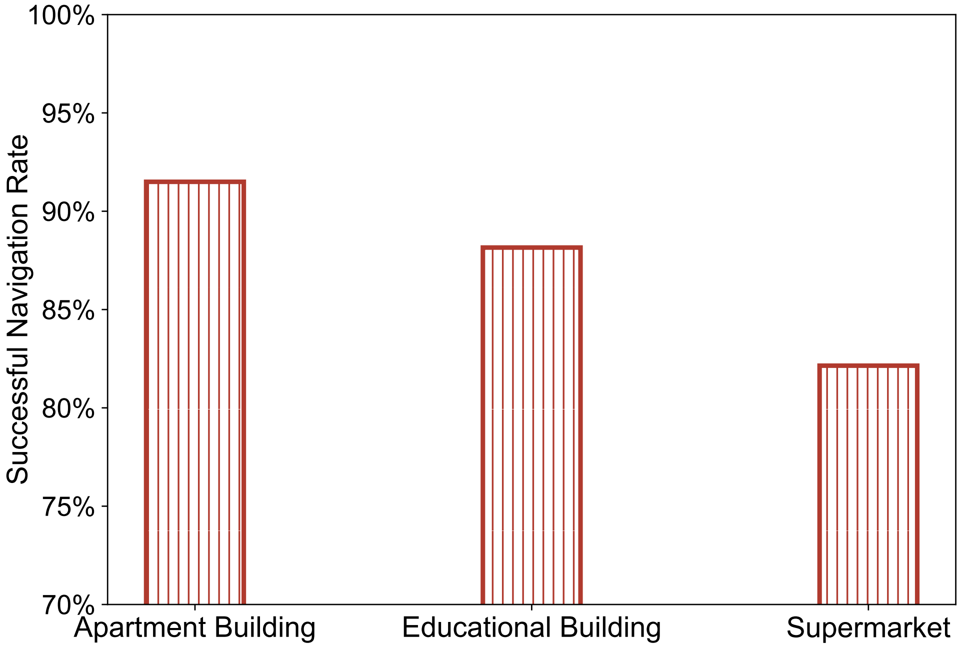


Figure 6.10: Comparison of Successful Navigation Rate in Different Scenarios

The comparison results of navigation simulation experiments of our system in the three different indoor scenarios are shown in Figure 6.10. The successful navigation rate of the system can reach 91.5% and 88.2% respectively in the apartment building and educational building. However, the successful navigation rate in the supermarket is only 82.1%, reduced by about 10%.

In the supermarket scenario, the main reason for the obvious decline of positioning accuracy and successful navigation rate is that there are many interferences caused by the complex indoor environment and the crowded customer flow. For example, the moving pedestrians or shopping carts may block the POI, which leads to great changes in the content of video frames, and thus the system could not make good image matching.

But generally speaking, our positioning and navigation system in these three types of common indoor scenarios shows superior performance, regardless of the specific differences and dynamic variations in the environments.

6.2.6 Influence of Lighting Conditions

In the three scenarios (the apartment building, educational building and supermarket) described before, video data is collected in multiple time periods of the day. Thus, we can compare the performance of our positioning and navigation system under different lighting conditions. In the experiment, videos are recorded at about $9:00$, $14:00$ and $19:00$.

The results of influence on positioning accuracy are shown in Figure 6.11. Considering all the experimental scenarios, the means of positioning errors of our system in the three time periods are $0.51m$, $0.52m$ and $0.56m$ respectively. This reveals that the positioning accuracies in the morning and in the afternoon are slightly different. While in the nighttime, the positioning accuracy decreases rapidly, and the positioning error increases by 7.6% compared with that in the daytime.

In addition, among all the three experimental scenarios, the apartment building experiences the largest variation of the lighting condition during one day, and the

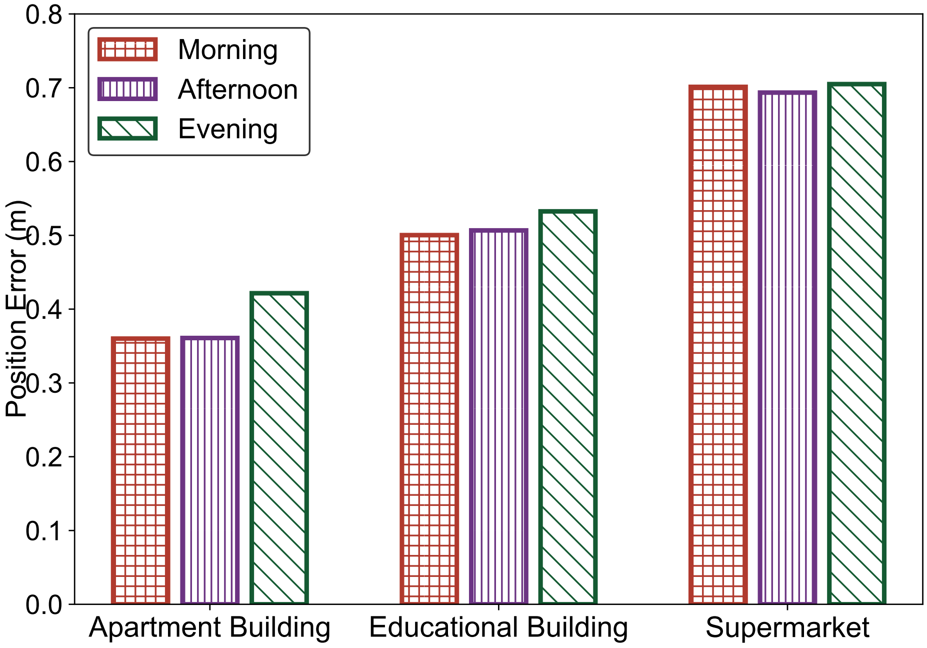


Figure 6.11: Comparison of Positioning Accuracy in Different Time Periods

positioning error of the system also increases sharply by 16.8% from 0.36m in the daytime to 0.42m in the nighttime. This change of positioning error can reflect the influence of different lighting conditions on the system performance. The main reason is that in the scenarios (such as apartment buildings) where the lighting conditions at night may be significantly worse, the number of feature points that can be extracted from the captured video frames will be dramatically reduced. This will increase the image matching error between the smartphone camera and monitoring camera, thus affecting the positioning accuracy. For similar reasons, the successful navigation rate of our system in the apartment building is only 86.7% in the nighttime, which is around 3.1% lower than that in the daytime.

6.2.7 System Latency Comparison

The latency of the positioning and navigation service is mainly caused by the system taking a certain amount of time to process and calculate, which is strongly associated with the designed localization and tracking algorithms. The overall latency of the system is composed of four major parts: feature detection, POI recognition, VO tracking and pose estimation. Because the image resolution of the captured video frame can significantly affect the running time of all these modules, the resolution is included in the experimental factors influencing the system latency. In addition, the indoor positioning system, JVWL, is compared with our system to prove the efficiency of the proposed algorithms.

The latencies of four modules in our system are tested separately. First of all, after the system starts up, for the module of extracting and matching the A-KAZE feature points in image frames, the average latencies of our system are $21ms$ and $31ms$ respectively with the image inputs of 720p and 1080p resolution. Secondly, POIs are detected and recognized. In the meantime, the visual odometry will run synchronously to estimate the motion of the smartphone camera between adjacent image frames. The above two parallel modules spend $66ms$ at 720p resolution and $113ms$ at 1080p resolution. Finally, with regard to the module for pose estimation and location tracking of the smartphone camera, the average time spent by the system is $19ms$.

The overall comparison of experimental results is shown in Figure 6.12. The top edge of each bar represents the average latency of the system. The distance between the top or bottom cap and the edge shows the value of the standard deviation. With regard to the comparative experimental systems, MVG does not provide the relevant

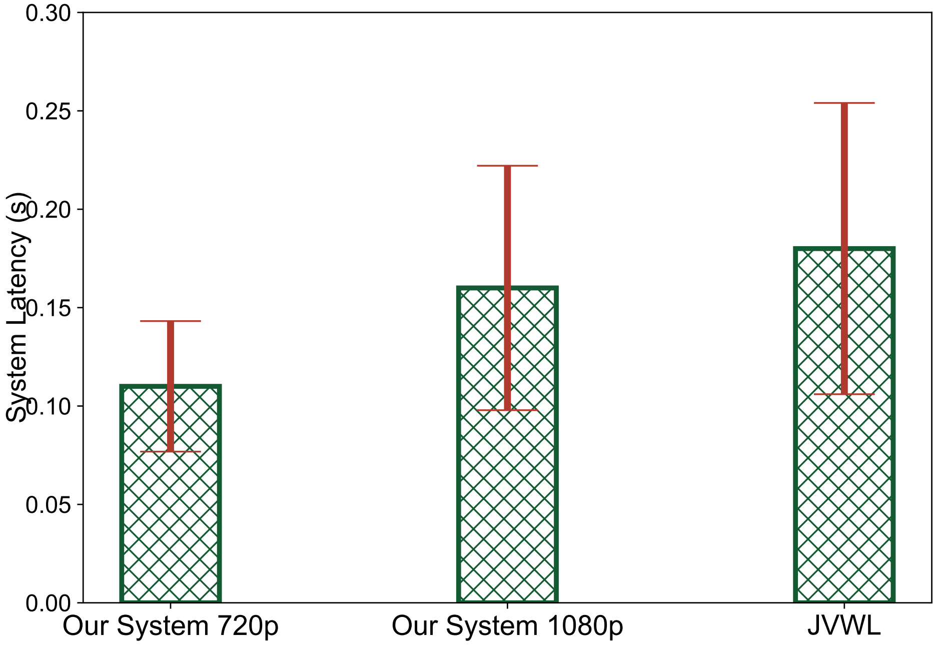


Figure 6.12: Overall System Latency Comparison

data of system running time. In addition, due to spending a large amount of time on landmark detection and image matching, the total time required for localization of HAIL is 3.7s, closely to 4s, which is far more than that of our system. Therefore, in this experiment, the latency of our system is only compared with JVWL.

It can be seen that the means of total latencies of our system correspond to 0.11s and 0.16s with the video inputs of 720p and 1080p. Compared with JVWL, the latencies of the designed system are remarkably reduced by 38.9% and 11.2% adopting the above two resolutions. Moreover, the standard deviations of system latencies are 0.03 for 720p and 0.06 for 1080p, which are 55.2% and 16.1% less than that of JVWL respectively. It proves that the positioning time of our system is not only shorter but also relatively more stable. Thus, the designed system can provide users with a smoother and more reliable positioning experience.

It can be summarized that our proposed system can achieve the positioning and tracking accuracy within one meter, and in the meanwhile, the overall system latency can be less than 0.2*s*. This excellent performance can satisfy the real-time demand of the indoor positioning and navigation system.

6.2.8 Overall System Performance Comparison

As explained in Subsection 6.1.4, the overall positioning performance of the proposed system is compared with three cutting-edge vision-based indoor positioning systems, which are MVG, JVWL and HAIL respectively.

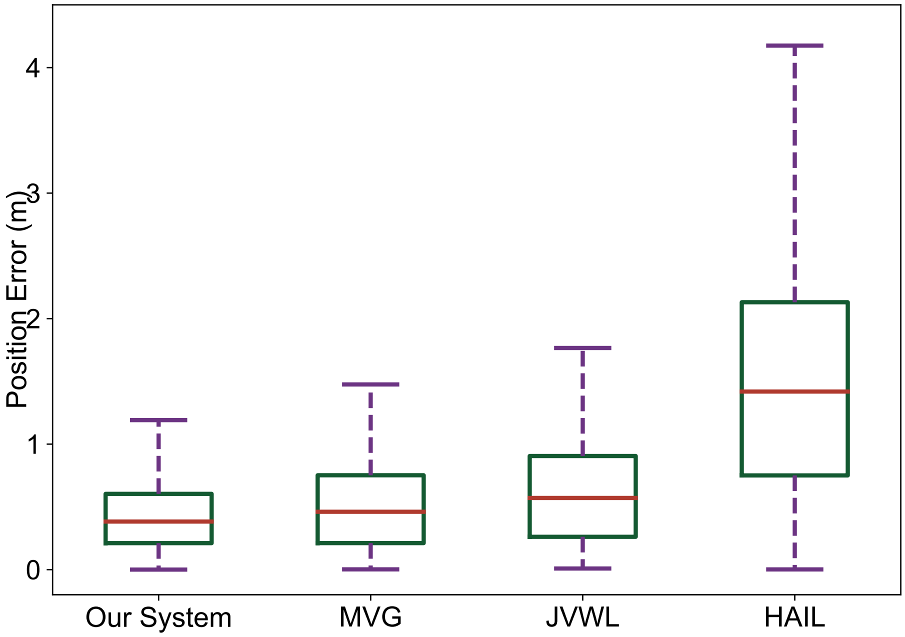


Figure 6.13: Overall Positioning Accuracy Comparison of Different Systems

For the box plot, the line in each box indicates the median of positioning errors. The lower and upper edges represent the $25th$ and $75th$ percentile positioning errors respectively. The lower and upper whiskers extend to the corresponding minimum and maximum values.

From Figure 6.13, it can be obtained that the mean of positioning errors of our system is $0.46m$, which is 13.2% less than that of MVG, 25.8% less than that of JVWL and 69.3% less than that of HAIL correspondingly. Therefore, it can be concluded that our system can provide the best positioning accuracy among these four systems.

In addition, with regard to the $90th$ percentile positioning error, the $0.93m$ of our system is 17.9% lower than MVG, 24.5% lower than JVWL and 67.2% lower than HAIL. Meanwhile, the variance of positioning errors of the proposed system is 0.13, which is reduced by 23.4%, 27.1% and 87.1% compared with MVG, JVWL and HAIL respectively. The above results prove that, as compared to the other three systems, our indoor positioning system not only can achieve the sub-meter level accuracy but also ensure the stability and reliability of the performance, so as to provide satisfying positioning services.

With regard to the navigation performance, experiments are conducted to compare the navigation simulation results of our system with Travi-Navi. As described in Subsection 6.1.4, Travi-Navi is one of the most advanced indoor navigation systems. Figure 6.14 depicts that the average success rate of indoor navigation achieved by our system is 90.6%, which is 3.4% higher than that of Travi-Navi. It can be obtained that our system can provide the navigation service with an overall success rate of about 90%, which can meet the needs of users in practical navigation applications.

The above experimental data clarifies that the navigation performance of our

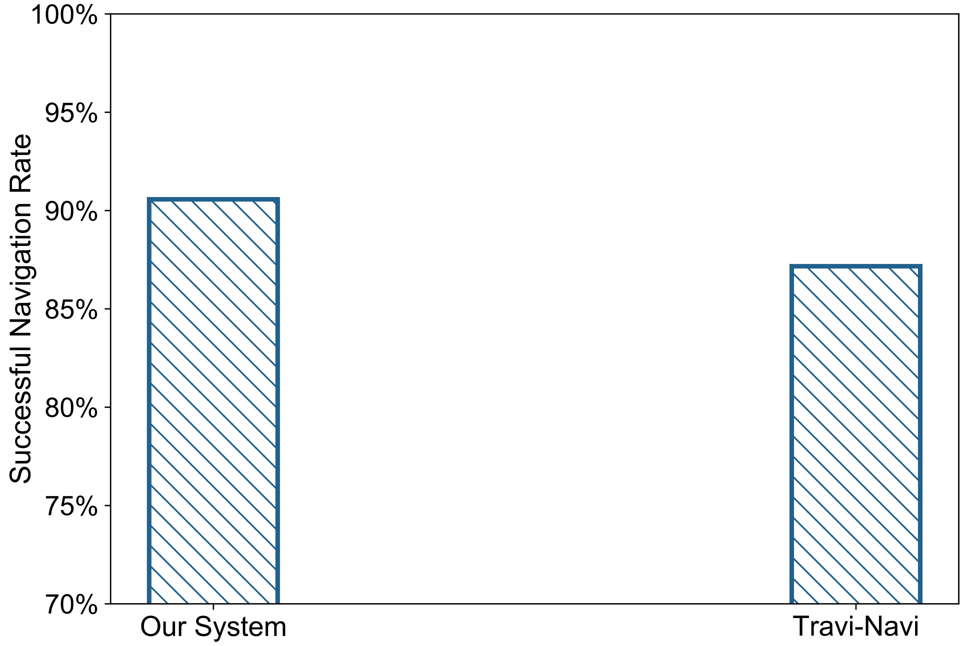


Figure 6.14: Overall Successful Navigation Rate Comparison of Different Systems

system can outperform some most effective existing indoor navigation systems. Furthermore, unlike other P2P navigation systems, such as Travi-Navi, our system has the potential of combining positioning with navigation. It is more labour-saving to leverage the image information which can be directly loaded than the trajectory map which needs to be manually constructed for assisting navigation.

Based on the results of all comparison experiments, it can be summarized that, compared with other advanced systems, our system has achieved better performance in indoor positioning accuracy and successful navigation rate. The main reasons are as follows.

1. **The inherent advantages of image-based positioning.** As mentioned before, our system is mainly based on vision, combining the image information captured by the monitoring camera and smartphone camera. The positioning accuracies of other indoor positioning systems based on inertial sensors or Wi-Fi signals are generally about $5m$. This is due to the inherent defects of these systems. For example, the acceleration and angular velocity measured by inertial sensors can have obvious drifts, which makes the calculated camera pose very unreliable. In addition, in a complex indoor environment, due to the occlusion of walls or other obstacles, Wi-Fi signals can fluctuate significantly, resulting in sharp attenuation or even disappearance of signal strength.

   In contrast, our vision-assisted positioning system has higher accuracy and stronger robustness. On the one hand, the image frames captured by cameras contain more abundant and fine information, such as the feature points, which are beneficial to image registration with high accuracy. On the other hand, the application of the visual odometry module enables the system to continuously locate and track the position of the user in motion. Even if only using the smartphone camera, without using the other monitoring cameras and POIs, the performance of our system is still relatively superior and stable.

2. **The optimization of data integration.** Though other leading-edge indoor positioning and navigation systems also exploit multiple sources of information as inputs, their data integration algorithms are loosely coupled. In these systems, each submodule generates its own positioning result independently. Therefore, the positioning errors from different submodules will be accumulated, which will further affect the final positioning accuracy of the whole system. By

contrast, the integration of information from different sources in our system is tightly coupled. In the proposed system, the positionings of the smartphone camera and monitoring camera are not two independent parallel submodules. Image feature matching and epipolar constraints both need to use and fuse the information from the video frames captured by the two cameras at the same time. This will significantly reduce the cumulative error caused by integrating the positioning results of each independent submodule.

Chapter 7

Conclusion

In this thesis, an indoor positioning and navigation system integrating the monitoring camera and smartphone camera has been proposed, which can be further divided into the following three parts.

First of all, the system extracts and matches A-KAZE feature points from the video frames of two cameras and identifies the POI in the shot scenes. From epipolar constraints and triangulation, the relative pose relationship between cameras and POI can be calculated. Thus, combined with the preloaded semantic information, the system can acquire the transformation scale from relative lengths in the perspective of the smartphone camera to real distances in the floor plan.

Secondly, according to the obtained scale relationship, by projecting, rotating and translating, any three-dimensional point in the mobile coordinate system can be converted to the corresponding two-dimensional point in the plane coordinate system. Therefore, when starting the system, the user's initial position in the actual environment can be determined successfully.

Then, when the user continues moving forward, our system will always track the user's real-time position, as well as provide appropriately planned paths and correct navigation tips.

Based on the designed algorithms, the proposed indoor positioning and navigation system has been implemented completely and its performance has been tested in various application scenarios, such as an educational building, an apartment building and a supermarket. Experimental results prove that our system can achieve a positioning accuracy of $0.46m$ and a successful navigation rate of 90.6%, which outperform the state-of-the-art schemes by more than 13% and 3% respectively. Furthermore, it is worth mentioning that the overall system latency is only about $0.2s$, which can meet the real-time demands. It can be concluded that our system can provide users with sub-meter level high-precision positioning and smooth navigation experiences.

In summary, assisted by already widely deployed monitoring cameras, the proposed indoor positioning and navigation system not only is easy to be implemented in real environments but also can provide accurate, efficient and reliable services for users. It is believed that in the future, the public places equipped with our system will be able to bring more convenience to people's lives.

Bibliography

Abdelnasser, H., Mohamed, R., Elgohary, A., Alzantot, M. F., Wang, H., Sen, S., Choudhury, R. R., and Youssef, M. (2015). Semanticslam: Using environment landmarks for unsupervised indoor localization. *IEEE Transactions on Mobile Computing*, **15**(7), 1770–1782.

Abdi, H. (2007). Singular value decomposition (svd) and generalized singular value decomposition. *Encyclopedia of measurement and statistics*, pages 907–912.

Alcantarilla, P. F. and Solutions, T. (2011). Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, **34**(7), 1281–1298.

Aqel, M. O., Marhaban, M. H., Saripan, M. I., and Ismail, N. B. (2016). Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, **5**(1), 1–26.

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, **13**(3), 108–117.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer.

Broumi, S., Bakal, A., Talea, M., Smarandache, F., and Vladareanu, L. (2016). Applying dijkstra algorithm for solving neutrosophic shortest path problem. In *2016 International conference on advanced mechatronic systems (ICAMechS)*, pages 412–416. IEEE.

Chen, P.-W., Ou, K.-S., and Chen, K.-S. (2010). Ir indoor localization and wireless transmission for motion control in smart building applications based on wiimote technology. In *Proceedings of SICE Annual Conference 2010*, pages 1781–1785. IEEE.

Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. (2007). Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, **29**(6), 1052–1067.

Derpanis, K. G. (2010). Overview of the ransac algorithm. *Image Rochester NY*, **4**(1), 2–3.

Ferris, B., Fox, D., and Lawrence, N. D. (2007). Wifi-slam using gaussian process latent variable models. In *IJCAI*, volume 7, pages 2480–2485.

Fuentes-Pacheco, J., Ruiz-Ascencio, J., and Rendón-Mancha, J. M. (2015). Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, **43**(1), 55–81.

Hesch, J. A. and Roumeliotis, S. I. (2011). A direct least-squares (dls) method for pnp. In *2011 International Conference on Computer Vision*, pages 383–390. IEEE.

Höflinger, F., Müller, J., Zhang, R., Reindl, L. M., and Burgard, W. (2013). A wireless micro inertial measurement unit (imu). *IEEE Transactions on instrumentation and measurement*, **62**(9), 2583–2595.

Jain, P., Manweiler, J., and Roy Choudhury, R. (2015). Overlay: Practical mobile augmented reality. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 331–344.

Kotaru, M., Joshi, K., Bharadia, D., and Katti, S. (2015). Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 269–282.

Kriz, P., Maly, F., and Kozel, T. (2016). Improving indoor localization using bluetooth low energy beacons. *Mobile Information Systems*, **2016**.

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g 2 o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613. IEEE.

Lawrence, N. and Hyvärinen, A. (2005). Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of machine learning research*, **6**(11).

Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, **81**(2), 155.

Liu, Z., Zhang, L., Liu, Q., Yin, Y., Cheng, L., and Zimmermann, R. (2016). Fusion of magnetic and visual sensors for indoor localization: Infrastructure-free and more effective. *IEEE Transactions on Multimedia*, **19**(4), 874–888.

Liu, Z., Cheng, L., Liu, A., Zhang, L., He, X., and Zimmermann, R. (2017). Multiview and multimodal pervasive indoor localization. In *Proceedings of the 25th ACM international Conference on Multimedia*, pages 109–117.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, **60**(2), 91–110.

Mur-Artal, R. and Tardós, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, **33**(5), 1255–1262.

Niu, Q., Li, M., He, S., Gao, C., Gary Chan, S.-H., and Luo, X. (2019). Resource-efficient and automated image-based indoor localization. *ACM Transactions on Sensor Networks (TOSN)*, **15**(2), 1–31.

Papaioannou, S., Wen, H., Markham, A., and Trigoni, N. (2014). Fusion of radio and camera sensor data for accurate indoor positioning. In *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, pages 109–117. IEEE.

Robertson, P., Angermann, M., and Krach, B. (2009). Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 93–96.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee.

Shin, H., Chon, Y., and Cha, H. (2011). Unsupervised construction of an indoor floor plan using a smartphone. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **42**(6), 889–898.

Strasdat, H., Montiel, J., and Davison, A. J. (2010). Scale drift-aware large scale monocular slam. *Robotics: Science and Systems VI*, **2**(3), 7.

Taira, H., Okutomi, M., Sattler, T., Cimpoi, M., Pollefeys, M., Sivic, J., Pajdla, T., and Torii, A. (2018). Inloc: Indoor visual localization with dense matching and view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7199–7209.

Taketomi, T., Uchiyama, H., and Ikeda, S. (2017). Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, **9**(1), 1–11.

Teng, J., Zhang, B., Zhu, J., Li, X., Xuan, D., and Zheng, Y. F. (2013). Ev-loc: integrating electronic and visual signals for accurate localization. *IEEE/ACM Transactions on Networking*, **22**(4), 1285–1296.

Xiao, F., Wang, Z., Ye, N., Wang, R., and Li, X.-Y. (2017). One more tag enables fine-grained rfid localization and tracking. *IEEE/ACM Transactions on Networking*, **26**(1), 161–174.

Zafari, F., Gkelias, A., and Leung, K. K. (2019). A survey of indoor localization systems and technologies. *IEEE Communications Surveys & Tutorials*, **21**(3), 2568–2599.

Zhang, Y., Jin, R., and Zhou, Z.-H. (2010). Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, **1**(1-4), 43–52.

Zhang, Z. (1998). Determining the epipolar geometry and its uncertainty: A review. *International journal of computer vision*, **27**(2), 161–195.

Zheng, Y., Shen, G., Li, L., Zhao, C., Li, M., and Zhao, F. (2017). Travi-navi: Self-deployable indoor navigation system. *IEEE/ACM transactions on networking*, **25**(5), 2655–2669.