Locate, Tell, and Guide: Enabling Public Cameras to Navigate the Public

Guoxuan Chi, Jingao Xu, Jialin Zhang, Qian Zhang, *Student Member, IEEE,* Qiang Ma, *Member, IEEE,* Zheng Yang*, *Senior Member, IEEE,*

Abstract—Indoor navigation is essential to a wide spectrum of applications in the era of mobile computing. Existing vision-based technologies suffer from both start-up costs and the absence of semantic information for navigation. We observe an opportunity to leverage pervasively deployed surveillance cameras to deal with the above drawbacks and revisit the problem of indoor navigation with a fresh perspective. In this paper, we propose *iSAT*, a system that enables public surveillance cameras, as indoor navigating satellites, to locate users on the floorplan, tell users with semantic information about the surrounding environment, and guide users with navigation instructions. However, enabling public cameras to navigate is non-trivial due to 3 factors: absence of real scale, disparity of camera perspective, and lack of semantic information. To overcome these challenges, *iSAT* leverages POI-assisted framework and adopts a novel coordinate transformation algorithm to associate public and mobile cameras, and further attaches semantic information to user location. Extensive experiments in 4 different scenarios show that *iSAT* achieves a localization accuracy of 0.48m and a navigation success rate of 90.5%, outperforming the state-of-the-art systems by $>$ 30%. Benefiting from our solution, all areas with public cameras can upgrade to smart spaces with visual navigation services.

Index Terms—Indoor localization, navigation, map construction, computer vision

1 Introduction

INDOOR location services have established the foundation for smart life and space. Many navigation systems have been proposed over the past decade using various solutions, including Wi-Fi [1], [2], [3], RFID [4], [5], inertial sensors [6], [7], and cameras [8], [9], etc. Among these systems, vision-based navigation has become one of the most attractive solutions in practice. By leveraging the Visual Odometry (VO) [10] and Simultaneous Localization and Mapping (SLAM) [11], vision-based systems can locate users with fine granularity and construct the map of surrounding environments [12], [13], [14]. In addition, vision-based solutions hold the potential to offer user-friendly interaction with visual navigation instructions rendered on real-world objects in the user interface, based on the continuously growing AR/MR technology [15].

According to our deployment experience and customers' demands, however, existing vision-based navigation systems suffer from two drawbacks. First, indoor map construction incurs high bootstrapping overhead. Specifically, all existing solutions based on visual SLAM involve a labor-intensive and time-consuming site survey to gather images (or keyframes) at each location in an environment. What's worse, due to frequent Line-Of-Sight (LOS) blockage by crowds or environmental dynamics, such a cumbersome site survey needs to be repeated over time. Although some recent works adopted a peer-to-peer (P2P) navigation mode by leveraging crowd-sourcing scheme [14], [16], [17], they also need considerable start-up efforts and suffer from limited coverage since the crowd-sourced trajectories cannot cover all destinations. Second, the positioning results from the images are irrelevant to the floorplan, thus lack of the necessary semantic information for navigation, for example, the name of destinations like Room 209 or Starbucks in the floorplan. In detail, the map generated from images is merely a set of keyframes and map points, whose locations are in the camera-coordinate, rather than floorplan-coordinate [1]. To navigate users/robots to the destinations in the floorplan, it's essential to associate the floorplan-coordinate and the camera-coordinate.

Nowadays, surveillance cameras are pervasively deployed in public areas, such as shopping malls, museums, galleries, and so on [18]. On this basis, we find an opportunity to overcome the above limitations and underpin a navigation solution with a novel perspective — Can we enable public cameras to navigate users? The rationale behind this vision is two-fold: on the one hand, surveillance cameras hold the potential to serve as the automatic map constructor and real-time map updater, which will ease, even eliminate, the human efforts for site-survey. On the other hand, with the prior knowledge of the camera's location, surveillance cameras can be leveraged as anchors to associate camera-coordinate and floorplan-coordinate, and hence provide the generated map with absolute locations in the floorplan. However, translating the insight into a navigation system is non-trivial and faces three significant challenges:

1) **Absence of real scale.** In additional to simply locating users in the floorplan, a navigation service should also tell

• *Guoxuan Chi, Jingao Xu, Jialin Zhang, Qian Zhang, Qiang Ma and Zheng Yang are with the School of Software and BNRist, Tsinghua University, Beijing, China, 100084.*
E-mail: {chiguoxuan, xujingao13, zhjialin16, qzhangqz123, tsinghuamq, hmilyyz}@gmail.com

Manuscript received xxx; revised xxx.
(Corresponding author: Zheng Yang.)

1. In this paper, these two coordinates are illustrated in Figure 4a.

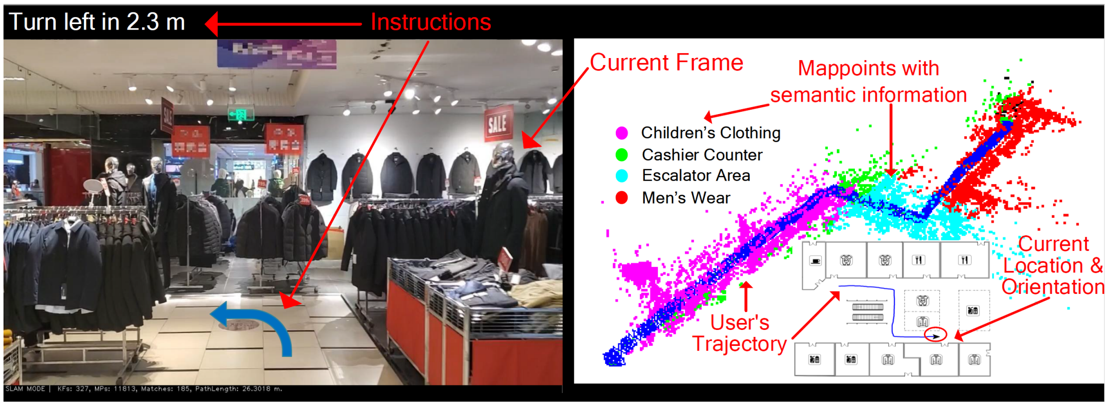


Fig. 1. User interface of our navigation system. *iSAT* provides users with: 1) instant location and orientation in constructed map and floorplan, 2) navigation instruction rendered in user's view and distance to next crossing point, 3) semantic information about surrounding environment.

them the direction and absolute distance to the destination. However, mainstream public cameras and backside imaging cameras on commodity smartphones are monocular cameras, and solutions based on such cameras will hardly acquire the absolute distance in real-world [19]. Hence, it's challenging for leveraging public cameras to meet the demands for user-friendly navigation.

2) **Disparity of camera perspective.** Although both mobile camera and surveillance camera follow pinhole camera model and capture overlapping areas, their perspectives and intrinsic parameters would vary a lot [20]. Specifically, the mobile camera takes shots from a horizon-view, while the surveillance camera takes shots from a top-view at a relatively far distance. Matching frames from two cameras with different perspectives can be difficult. Different intrinsic parameters of two heterogeneous cameras also hinder accurate pose estimation. As a result, the transformation between camera-coordinate and floorplan-coordinate is non-trivial.

3) **Lack of semantic information.** In a typical application scenario of the navigation system, a user will either input the name of stores (*e.g.* McDonald's, Room 209) or select a location on the floorplan as a destination. That is to say, from the user's perspective, the destination is semantic information on the floorplan. However, the map constructed by visual SLAM in previous works is typically a set of keyframes and map points, lacking semantic information for navigation without site survey.

To tackle the above challenges, we design *iSAT*, an indoor navigation system that enables public cameras to serve as **i**ndoor **SAT**ellites, whose missions are similar to the outdoor satellites in GPS systems. As shown in Figure 1, in *iSAT*, surveillance cameras are leveraged to 1) **locate** users with absolute location on indoor floorplan, 2) **tell** users with semantic information about surrounding environment, 3) **guide** users with navigation instructions rendered on the mobile device. To integrate the public cameras and mobile cameras, *iSAT* uses images containing the same Point Of Interest (POI) from both sides and leverages advanced visual features to calculate the relative position between the public camera and the mobile camera. To obtain the real scale, *iSAT* combines geometric information of surveillance camera and POI with calculated relative poses to get a user's location in floorplan. To acquire semantic information about the surrounding environment, *iSAT* transforms the 3D points to their 2D coordinates in floorplan based on the results of previous processes and then attaches semantic information to them.

We implement *iSAT* on the server and four different types of smartphones. We conduct extensive experiments under four common scenarios of indoor navigation systems including a shopping mall, office building, library, and teaching building. The total size of the experimental areas is more than $4000m^2$. We locate and navigate users for more than 20 hours, collecting 115.1k video frames. Evaluation results demonstrate that *iSAT* achieves a localization accuracy of $0.48m$, outperforming existing state-of-the-art systems by more than 30%. The navigation success rate of *iSAT* is 90.5%, which can compete with most other existing systems.

Our key contributions are summarized as follows:

• To the best of our knowledge, this is the first work that combines mobile cameras with surveillance cameras and further enables them to **locate**, **tell** and **guide** mobile users/robots with little human start-up effort. Benefiting from this technology, all areas with public cameras can upgrade to smart spaces with visual navigation services.

• We design novel algorithms to solve three challenges, including the absence of real scale, disparity of camera perspective, and lack of semantic information to enable public cameras to navigate. Our scheme of interaction and communication between ambient and mobile cameras provides a new perspective to underpin indoor localization and navigation with fine granularity.

• We fully prototype *iSAT* and conduct extensive experiments under 4 different scenarios with 5 state-of-the-art approaches. The evaluation results show that *iSAT* achieves delightful results (localization accuracy of $0.48m$ and navigation success rate of 90.5%) and outperforms previous works in all scenarios, shedding light on pervasive indoor navigation for mobile users/robots.

The rest of the paper is organized as follows. We first present the overview of *iSAT* in Section 2, followed by a detailed presentation of *Real Scale Acquisition* in Section 3 and *3D-2D Coordinate Transformation* in Section 4. Methodology of *Localization, Navigation, and Semantic Map Construction* is proved in Section 5. Implementation and evaluation are described in Section 6. Then we review related works in Section 7 and conclude this paper in Section 8.

2 System Overview

Figure 2 sketches the system architecture of *iSAT*.

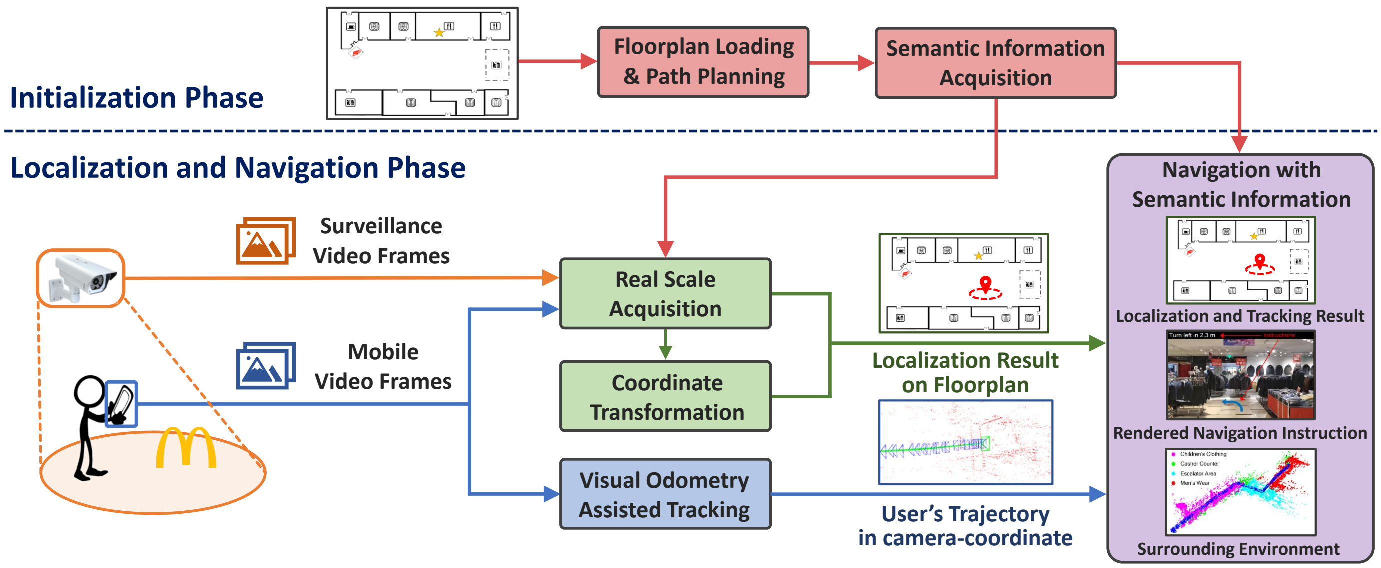


Fig. 2. *iSAT* architecture

2.1 Workflow from User Perspective

When a user activates *iSAT*, the mobile camera is automatically turned on. Then, it records videos of the surrounding environment and sends the videos to the server. After some lightweight processing which takes no more than 0.5s, the server will send the user's location back to the mobile device. Then, the client side of *iSAT* will display the instant location with floorplan on the screen. If the user wants to be navigated to a certain location in this building, he/she can either select a location on the floorplan or input a semantic location (*e.g.* Room 211) as destination. Afterwards, the server side of *iSAT* will choose the optimal path and navigate the user to the destination. During the navigation process, a 3D semantic map which shows different functional areas of the current building will be constructed in real-time.

2.2 Workflow from Server Perspective

Once the system is deployed, *iSAT* enters the initialization phase. During the initialization phase, *iSAT* loads the floorplan with semantic information (*e.g.* the coordinates of POI, landmarks and surveillance cameras, *etc.*), which is essential for localization and navigation. By analyzing connectivity among landmarks, *iSAT* generates a connected graph and runs a path planning algorithm. The path planning result can be used multiple times and does not need to be updated unless the indoor environment changes (*e.g.* new surveillance cameras are installed or functions of certain rooms are changed).

During the localization and navigation phase, multiple surveillance cameras monitor different public areas in the building and stream recorded videos to the server. Once the videos recorded and uploaded by client are also received by the server, the *Real Scale Acquisition Module* of *iSAT* starts kicking off, which calculates the relative pose (*i.e.* orientation and location) between the mobile camera and surveillance camera, and further solves the real scale using prior provided semantic information (*e.g.* the distance from surveillance camera to POI in real world). Leveraging the *Coordinate Transformation Module*, *iSAT* gets a transformation matrix, which transforms 3D points in camera-coordinate to the corresponding 2D points in floorplan-coordinate. Combining the results of both two modules mentioned above, *iSAT* obtains the user's initial location on the floorplan.

In order to track and navigate users in non-line-of-sight (NLOS) environment, *iSAT* integrates *Visual Odometry Module* and fully utilizes the semantic information. After getting the user's initial location, the VO module takes over the system and continuously estimates the user's motion. Furthermore, *iSAT* generates user's trajectory and provides real-time navigation instructions based on user's current location. It is worth mentioning that once a POI is captured by the mobile camera during the navigation process, the relocalization function of *iSAT* activates automatically, which effectively reduces the accumulative errors and avoids severe deviation. During the navigation process, VO also generates the point cloud map of the surroundings, which will be modified into a 3D semantic map based on the division of functional areas in the building. The constructed map can not only provide room-level semantic information for robots to complete certain tasks (*e.g.* automatic package delivery), but also have great potential to be combined with AR/MR technology to provide user-friendly interaction.

On the basis of the above procedures, 1) instant localization result on floorplan, 2) navigating instructions rendered in user's view, and 3) semantic information about the surrounding environment are sent back to user for navigation.

3 Real Scale Acquisition

In recent years, feature-based image registration and pose estimation are widely used in localization and navigation systems [14], [21]. However, due to the lack of real-world scale, previous vision-based indoor localization and navigation systems need to be integrated with other modules (*e.g.* IMU sensor and wireless module) and cannot work alone. In this section, we design an approach of real scale acquisition, which enables surveillance cameras and mobile cameras to obtain the absolute location without extra modules.

Figure 3 shows the workflow of the real scale acquisition module, which can be divided into three parts: POI detec-

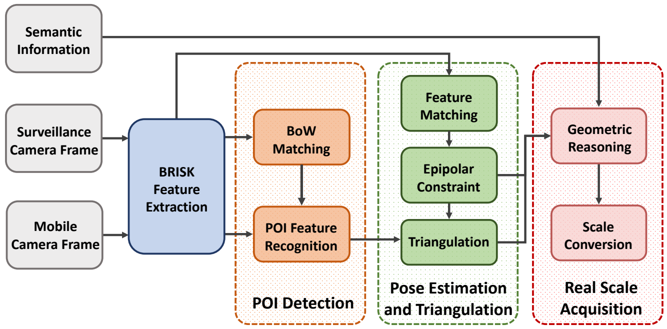


Fig. 3. Workflow of real scale acquisition module

tion, relative pose estimation, and scale conversion, which will be further introduced in detail. *iSAT* first extracts feature points in the video frames from all surveillance cameras and user's mobile camera and chooses the best surveillance camera whose captured scene is the most similar to that of the mobile camera, and then detects POI on video frames. Afterwards, *iSAT* estimates the relative pose between the mobile camera and the surveillance camera by geometry constraints, and calculates the relative location between the mobile camera and POI by triangulation. Then, with known semantic information, *iSAT* solves the scale conversion ratio, which converts the length in camera-coordinate system to the length in the real world.

As aforementioned, due to the large perspective disparity between the mobile camera and surveillance camera, it is non-trivial to match feature points extracted from different images straightforwardly. To solve the challenge, the key innovations of *iSAT* are two-fold: first, we compare different feature points and evaluate their performance with diverse view disparity (*i.e.*, the angle at POI) in different experimental scenarios. Based on extensive evaluations, we finally choose the BRISK [22], which is the most compelling feature. The experiment will be described in Section 6.2.1; And second, we modify the code of the feature matching algorithm, which is provided in the pose estimation function in OpenCV library, to support two cameras with diverse intrinsic parameters (*i.e.*, K_1 and K_2 in the following content).

3.1 POI Detection

The whole process of POI detection can be divided into two stages. First, *iSAT* automatically selects the best one from all surveillance cameras. Due to the large number of surveillance cameras in large-scale public places (*e.g.* shopping mall), feature matching is usually time-consuming and incurs high system latency. Therefore, we use DBoW [23] to calculate the similarity between the frames captured by the mobile camera and surveillance cameras. By computing the similarity of word vector corresponding to each video frame, *iSAT* gets the best surveillance camera whose captured scene is the most similar to that of the mobile camera.

Afterwards, *iSAT* recognizes POI by matching the feature points in the mobile camera frame and surveillance camera frame. Since the location and orientation of the surveillance camera are fixed, we assume that the POI is always projected on a specific region (*i.e.* POI region) of the surveillance camera frame as shown in Figure 4b. By extracting the features in that region and selecting the most similar one in mobile camera frame, *iSAT* detects POI on both mobile camera frame and surveillance camera frame.

It's worth mentioning that setting the size of the POI region is a trade-off problem. If the POI region is set too small, few feature points can be detected in this region, thus the POI detection module will be more likely to fail. If the region is set too large, then the system error will increase. The guidelines for choosing POI regions are summarized as follows. First, the POI region must be captured by at least one surveillance camera and not easily obscured by other objects. Second, the texture of the POI region should be rich enough to extract several feature points.

3.2 Relative Pose Estimation

In this subsection, we mainly focus on the pose estimation technique by which *iSAT* gets the location of both surveillance camera and POI in camera-coordinate (*i.e.* O_s and P in Figure 4b).

As shown in Figure 4b, *iSAT* extracts features on both mobile camera frame and surveillance camera frame, and then estimates their relative pose by image registration. Epipolar geometry constraints [24], [25] is a widely used method to estimate the relative pose between two images. By matching features on mobile camera frame and surveillance camera frame, *iSAT* obtains several pairs of 2D feature points. Denote the 2D feature point on mobile camera frame as $(u_1, v_1)^{\mathrm{T}}$, and the matched feature point on surveillance camera frame as $(u_2, v_2)^{\mathrm{T}}$. Denote $\mathbf{P} = (X, Y, Z)^T$ as the 3D point in camera-coordinate corresponding with them. For ease of notion, we transform feature points into homogeneous coordinates as $\mathbf{p_1} = (u_1, v_1, 1)^{\mathrm{T}}$ and $\mathbf{p_2} = (u_2, v_2, 1)^{\mathrm{T}}$ respectively. According to the pinhole camera model, we get their relationships as follows:

$$\begin{aligned} s_1\mathbf{p_1} &= \mathbf{K_1}\mathbf{P}, \\ s_2\mathbf{p_2} &= \mathbf{K_2}(\mathbf{R_{cs}}\mathbf{P} + \mathbf{t_{cs}}), \end{aligned} \tag{1}$$

where $\mathbf{K_1}$ and $\mathbf{K_2}$ are known intrinsic matrices of mobile camera and surveillance camera respectively, s_1 and s_2 are the depths of feature point in mobile frame and surveillance camera frame, while $\mathbf{R_{cs}}$ and $\mathbf{t_{cs}}$ are the rotation matrix and translation vector from camera coordinate to surveillance camera coordinate respectively. Then we eliminate $\mathbf{P}$ and transform Equation 1 into the following form:

$$\begin{aligned} \mathbf{P} &= s_1\mathbf{K_1^{-1}}\mathbf{p_1}, \\ s_2\mathbf{K_2^{-1}}\mathbf{p_2} &= s_1\mathbf{R_{cs}}\mathbf{K_1^{-1}}\mathbf{p_1} + \mathbf{t_{cs}}. \end{aligned} \tag{2}$$

Make cross product of Equation 2 and the translation vector t, and then multiply $(\mathbf{K_1^{-1}}\mathbf{p_1})^{\mathrm{T}}$ on both sides. After deduction, we get the epipolar geometry constraints as follows:

$$\begin{aligned} \mathbf{p_2}^{-\mathrm{T}}\mathbf{K_2}^{-\mathrm{T}}\mathbf{E}\mathbf{K_1}\mathbf{p_1} &= 0, \\ \mathbf{E} &= \mathbf{t_{cs}} \times \mathbf{R_{cs}}, \end{aligned} \tag{3}$$

where $\mathbf{E}$ is the Essential Martix.

As mentioned above, feature matching usually provides hundreds of well-matched feature points, which can be used to solve Equation 2. *iSAT* first uses RANSAC (Random Sample Consensus) [26] algorithm to calculate $\mathbf{E}$, and then gets $\mathbf{t_{cs}}$ and $\mathbf{R_{cs}}$ through SVD (Singular Value Decomposition).

Although *iSAT* solves $\mathbf{t_{cs}}$ and $\mathbf{R_{cs}}$ successfully, the depths of POI (s_1 and s_2) are lost during the solving process.

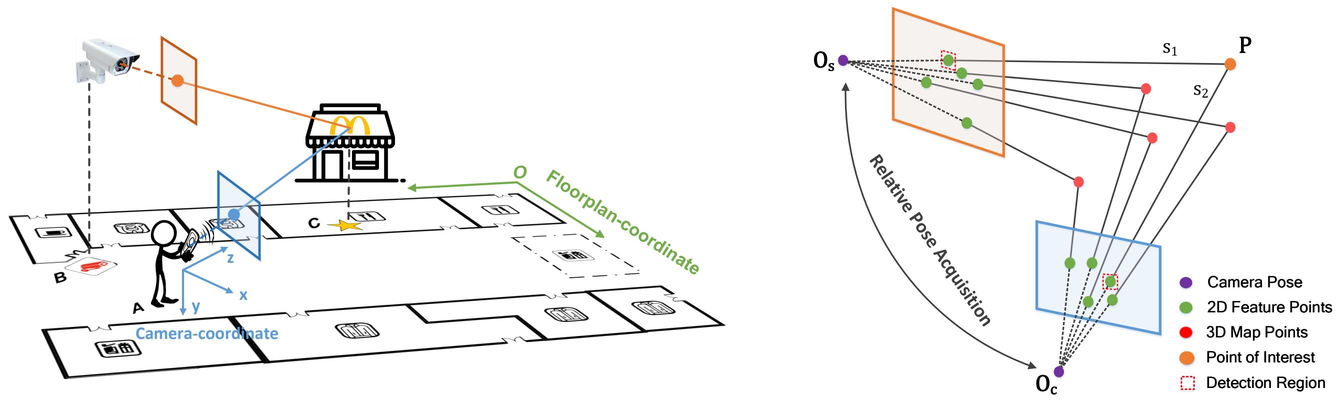


Fig. 4. Application scenarios: (a) from a real-world perspective, mobile camera and surveillance camera both capture POI, then *iSAT* locates the user. (b) from the camera-coordinate perspective, *iSAT* matches feature points, detects POI and estimates relative pose between mobile camera and surveillance camera.

Fortunately, with the help of triangulation [27], *iSAT* can reconstruct 3D coordinates of POI based on its projection on 2D frames. Suppose $\mathbf{p_1}$ and $\mathbf{p_2}$ are feature points corresponding to POI, we can solve s_1 by making cross product of vector $\mathbf{K_2^{-1}p_2}$ on both sides of Equation 2:

$$s_1\mathbf{R_{cs}}(\mathbf{K_1^{-1}p_1}) \times (\mathbf{K_2^{-1}p_2}) + \mathbf{t_{cs}} \times (\mathbf{K_2^{-1}p_2}) = 0. \quad (4)$$

Since *iSAT* has solved the rotation matrix $\mathbf{R_{cs}}$ and the translation vector $\mathbf{t_{cs}}$, Equation 4 can be treated as a linear equation of s_1. *iSAT* can also get s_2 by simply making cross product of $\mathbf{K_1^{-1}p_1}$ and Equation 2, thus constructing a linear equation of s_2 in the same way. Based on the depth s_1, s_2 and the translation vector $\mathbf{t_{cs}}$, the shape of $\triangle O_cO_sP$ in Figure 4b is completely determined.

3.3 Scale Conversion

So far, we have determined the geometric relationship (*i.e.* the shape of $\triangle O_cO_sP$) in camera-coordinate. However, due to the scale ambiguity of monocular vision [19], the user's location in the real world remains unknown. More specifically, s_1, s_2 and $\|\mathbf{t_{cs}}\|_2$ in Section 3.2 only represent the relative length of $\overrightarrow{O_cP}$, $\overrightarrow{O_sP}$ and $\overrightarrow{O_cO_s}$ instead of the physical length in real world.

Fortunately, with the help of semantic information, *iSAT* can assign the physical length (in meters) to $\triangle O_cO_sP$. Suppose the real-world distance from surveillance camera to POI is l meters. Obviously, there is only a constant conversion ratio between l and $s_2 = \|\overrightarrow{O_sP}\|_2$. *iSAT* calculates the conversion ratio as follows:

$$r = \frac{l}{\|\overrightarrow{O_sP}\|_2} = \frac{l}{s_2}. \quad (5)$$

In other words, one unit length in camera-coordinate corresponds to r meters in the real world.

4 3D-2D Coordinate Transformation

All these poses and locations mentioned in Section 3 are in camera-coordinate. To get the user's location on the floorplan, a transformation from camera-coordinate to floorplan-coordinate is necessary. Therefore, we analyze the differences between the two coordinates in the following aspects:

1) **Different Dimentions.** The floorplan-coordinate is 2D, while the camera-coordinate is 3D.

2) **Different Scales.** The scale of floorplan-coordinate is the unit length in real world(*e.g.* meter), while standardized scale, which varies a lot in different application scenarios, is used in camera-coordinate.

3) **Different Origins.** The origin of floorplan-coordinate is usually preset at the corner of the building(*e.g.* O in Figure 4a). However, the origin of camera-coordinate depends on the location where the user first activates *iSAT* on the smartphone (*e.g.* O_c in Figure 4b).

4) **Different Orientations.** As shown in Figure 4a, the orientation of floorplan-coordinate is preset, while the the orientation of camera-coordinate is determined by user's orientation when *iSAT* is activated.

Except the different scales problem which has been solved in Section 3, *iSAT* needs to eliminate three other differences, which will be described in the following subsections. For ease of notion, we denote the user's initial location on floorplan as A, while the known 3D location of surveillance camera and POI as B and C respectively, as shown in Figure 4a.

4.1 3D-2D Projection

To locate and track users on 2D floorplan, a projection from 3D camera-coordinate to the ground is necessary. Our system suggests users to keep the y-axis (shown in Figure 4a) of their device perpendicular to the ground when starting service. So the 3D-2D projection matrix can be expressed into the following form:

$$\mathbf{M_p} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (6)$$

which means *iSAT* only needs to delete the y-axis component of O_c, O_s and P to complete the 3D-2D projection.

More specifically, suppose *iSAT* projects $\triangle O_cO_sP$ from camera-coordinate to the horizontal plane as $\triangle O_c'O_s'P'$, we can get $\overrightarrow{O_s'P'}$ by the following equation:

$$\overrightarrow{O_s'P'} = \mathbf{M_p}(\overrightarrow{O_sP}). \quad (7)$$

It's worth mentioning that *iSAT* only requires the users to hold the smartphones with its y-axis perpendicular to the ground during the initial localization stage. Once associate the 3D and 2D coordinates, which indicates the user's initial location is acquired, users can hold their smartphones freely.

4.2 2D-2D transformation

However, after projection and scaling, $r\overrightarrow{O'_sP'}$ and $\overrightarrow{BC}$ are still not equal. Even though they have the same length, they are not in the same 2D coordinate. Therefore, *iSAT* calculates the 2×2 rotation matrix $\mathbf{R_f}$ and 2×1 translation vector $\mathbf{t_{OA}}$ in two steps.

First, *iSAT* gets $\mathbf{R_f}$ by solving an optimization problem in the following form:

$$\mathbf{R_f} = \arg\min_{\mathbf{R_f}} e = \arg\min_{\mathbf{R_f}} \|\overrightarrow{BC} - r\mathbf{R_f}(\overrightarrow{O'_sP'})\|_2^2. \quad (8)$$

Then, by scaling and rotating the translation vector $\mathbf{t_{cs}}$, *iSAT* gets $\mathbf{t_{AB}} = \overrightarrow{AB}$ in the floorplan coordinate as follows:

$$\mathbf{t_{AB}} = \overrightarrow{AB} = r\mathbf{R_f}\mathbf{M_p}\mathbf{t_{cs}}. \quad (9)$$

Therefore, the origin of the world coordinate on 2D floorplan (*i.e.* $\mathbf{t_{OA}} = \overrightarrow{OA}$) can be calculated by *iSAT*:

$$\mathbf{t_{OA}} = \mathbf{t_{OB}} - r\mathbf{R_f}\mathbf{M_p}\mathbf{t_{cs}}. \quad (10)$$

Based on the scale conversion ratio r, projection matrix $\mathbf{M_p}$, rotation matrix $\mathbf{R_f}$ and translation vector $\mathbf{t_{OA}}$, *iSAT* can tranform any point from camera-coordinate to floorplan-coordinate.

5 Localization, Navigation and Semantic Map Construction

With all information provided in Section 3 and Section 4, *iSAT* can successfully gets user's location on floorplan, and then gives navigation instructions. During the navigation process, *iSAT* constructs a 3D semantic map based on the known semantic information and point cloud. In this section, we explain the technical details of *iSAT* to achieve localization, navigation and semantic map construction.

5.1 High-accuracy Localization

The process of high-accuracy localization can be divided into two parts: initial location acquisition and real-time tracking.

Once the system is activated, *iSAT* first gets the user's initial location on the floorplan (*i.e.* location A in Figure 4a), which can be solved easily according to Equation 10.

However, the pose estimation function in Section 3.2 can only work in line-of-sight (LOS) environments, where POI can be captured by both mobile camera and surveillance camera. That means *iSAT* cannot track user's real-time location in non-line-of-sight (NLOS) environment using techniques mentioned above. Therefore, Visual Odometry (VO) [10] is introduced into *iSAT*, which estimates the motion of a camera in real-time using sequential frames. The VO module in *iSAT* exploits the idea of Perspective-n-Point (PnP) [28] to estimate continuous motion of the mobile camera. By matching feature points on two adjacent frames, *iSAT* can get the user's real-time pose in camera-coordinate.

Concretely, by the image registration and triangulation in Section 3.2, *iSAT* acquires a set of points correspondences, each composed of a 3D reference point $\mathbf{P_i} = (X_i, Y_i, Z_i)^{\mathrm{T}}$ in camera-coordinate and its 2D projection $\mathbf{p_i} = (u_i, v_i, 1)^{\mathrm{T}}$ on the k_{th} mobile camera frame.

Denote $\mathbf{T_k}$ as the transformation matrix of the k_{th} frame, which consists of a rotation matrix $\mathbf{R_k}$ and a translation vector $\mathbf{t_k}$. According to pinhole camera model and rigid transformation, we have:

$$s_i\mathbf{p_i} = \mathbf{K}(\mathbf{R_k}\mathbf{P_i} + \mathbf{t_k}) = \mathbf{K}\mathbf{T_k}\mathbf{P_i}. \quad (11)$$

Then it comes to solve an optimization problem to estimate transformation:

$$\mathbf{T_k} = \arg\min_{\mathbf{T_k}} e = \arg\min_{\mathbf{T_k}} \frac{1}{2}\sum_{i=1}^{n} \left\|\mathbf{p_i} - \frac{1}{s_i}\mathbf{K}\mathbf{T_k}\mathbf{P_i}\right\|_2^2. \quad (12)$$

To solve this problem, *iSAT* adopts EPnP [29] algorithm, an efficient $O(n)$ non-iterative solution of PnP to minimize the cost function e and acquire the optimal pose $\mathbf{T}_k$. Therefore, *iSAT* gets user's location and constructs trajectory in camera-coordinate.

Once *iSAT* gets the initial location, VO module turns on automatically, and gets user's real-time translations $\{\mathbf{t_0}, \mathbf{t_1}, \mathbf{t_2}, \ldots\}$ in camera coordinate as the user moves forward. Denote $A^{(i)}$ as the real-time location corresponding to the i_{th} frame in floorplan-coordinate. In order to locate users, *iSAT* only needs to project the translations to the floorplan:

$$\mathbf{t_{OA^{(i)}}} = \mathbf{t_{OA}} + \mathbf{t_{AA^{(i)}}} = \mathbf{t_{OA}} + r\mathbf{R_f}\mathbf{M_p}\mathbf{t_i}. \quad (13)$$

5.2 Navigation Methodology

During the initialization phase, *iSAT* reads semantic information, including the names of multiple landmarks and their connectivity, from the configuration files. Then *iSAT* constructs a connected graph $G = \langle V, E\rangle$, where the node $v_i \in V$ represent the location of the i_{th} landmark and edge $e_{ij} \in E$ represent the distance from v_i to v_j in real world. After that, the Dijkstra algorithm is applied to each node to obtain the shortest path and stores the result. When a user requests navigation service, *iSAT* will first get the user's current location and take the nearest landmark as the starting point, then give navigation instructions to the user. During the navigation process, the system will continuously get the user's real-time location from the visual odometry and lead user to the next landmark in planned path, until user arrives at the destination. Since VO is only based on the mobile camera, even in the blind area of the surveillance camera, user's location and trajectory can be reported correctly.

It's worth mentioning that *iSAT* introduces the relocation function, which significantly improves the navigation success rate in long-path navigation. During the navigation process, the POI recognition function mentioned in Section 3.1 runs continuously. Once the mobile camera successfully recognizes the POI, the system will automatically perform surveillance camera assisted localization again to correct user's current location.

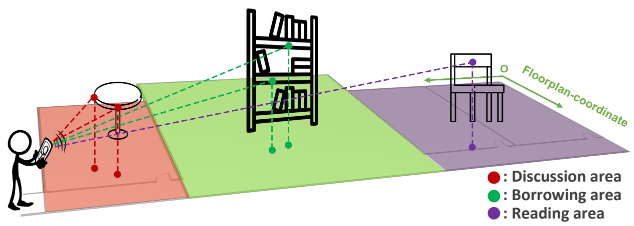


Fig. 5. *iSAT* transforms 3D points from camera-coordinate to floorplan-coordinate and then determines its semantic attribute according to which functional area its projection belongs to.

5.3 Semantic Map Construction

In our system, a 3D semantic map can: 1) Tell users/robots about the surrounding environment with room-level semantic information, which is important in certain scenarios (*e.g.* robot automatic package delivery). 2) Record the trajectory that the user/robot goes through, which is a convenient solution for some specific problems (*e.g.* reverse car-searching in indoor parking). 3) Have great potential to be combined with AR/MR technology. Compared with a fixed 2D map, 3D semantic map is built from users' perspective (*i.e.* mobile camera coordinate), which means instructions and descriptions can be easily rendered on users' views.

As mentioned in Section 5.2, once *iSAT* gets user's initial location, the visual odometry (VO) module will continuously track and navigate the user. During this process, VO module creates hundreds of new 3D map points via triangulation. Denote the set of map point as $\{\mathbf{P_1}, \mathbf{P_2}, \mathbf{P_3}, \ldots, \mathbf{P_N}\}$, each point $\mathbf{P_i}$ correspond to a certain location $(X_i, Y_i, Z_i)^{\mathrm{T}}$ in world coordinate.

To construct a 3D semantic map from map points, *iSAT* needs to transform map points from camera-coordinate to floorplan-coordinate and then determines its semantic attribute according to which functional area its projection belongs to, as shown in Figure 5. Concretely, *iSAT* first calculates the projection for each map point $\mathbf{P_i}$ and gets its projection:

$$\mathbf{p_i} = r\mathbf{R_f}\mathbf{M_p}\mathbf{P_i} + \mathbf{t_{OA}}, \tag{14}$$

and then gives label to each 3D point. Suppose we divide the floor into M disjoint areas, $A = \{A_1, A_2, A_3, \ldots, A_M\}$, we get $l_i = j$ if $\mathbf{p_i} \in A_j$.

6 Implementations and Evaluation

We implement *iSAT* on the server and conduct experiments using different mobile devices over various scenarios. In this section, we first introduce experimental settings and then present the detailed evaluation.

6.1 Experimental Methodology

6.1.1 Experimental Scenarios

We carry out extensive experiments in four typical public areas, including a library, an office building, a teaching building and a shopping mall. As shown in Figure 6, these areas have different floor layouts. Besides, user behaviors appear to be unique in these areas as well. For instance, office buildings are often filled with people during the daytime while almost empty at night. The teaching building is crowded or empty to different extents depending on the course schedule. The shopping mall is the most crowded place, while there are only a few people in the library.

Details of evaluations are summarized in Table 1. We utilize four different types of smartphones for data collection, including Samsung Galaxy S10, HUAWEI Honor 20, Google Nexus 6p and Google Pixel, which are equipped with mobile cameras with different camera intrinsics (*i.e.* focal length, lens center and distortion).

6.1.2 Experimental Setup

The client side of *iSAT* is implemented on the Android platform based on ROS [30] Android Platform with all of the mobile devices mentioned above, which record video at 30fps or 60fps. The size of each frame recorded by mobile cameras is either 1920 × 1080 or 1280 × 720 pixels, depending on the complexity of experimental scenarios. The HIKIVISION-C3A is used as the surveillance camera, which continuously records and streams videos to the server. The resolution of surveillance cameras is 1920 × 1080. There are 2-6 surveillance cameras deployed in each scenario. The server is equipped with i7-9700 CPU of 4.7GHz main frequency and 16G RAM, it runs Ubuntu 16.0.4 operating system. The communication between the client and the server is based on ROS, which integrates easy-to-use communication modules to transport text, images and videos between different devices. So that we don't need to focus on the detailed design of the communication module.

We modify ORB-SLAM [31] and use it as visual odometry. And we use Pangolin, a lightweight portable rapid development library for managing OpenGL display, to visualize constructed 3D point cloud map and validate results.

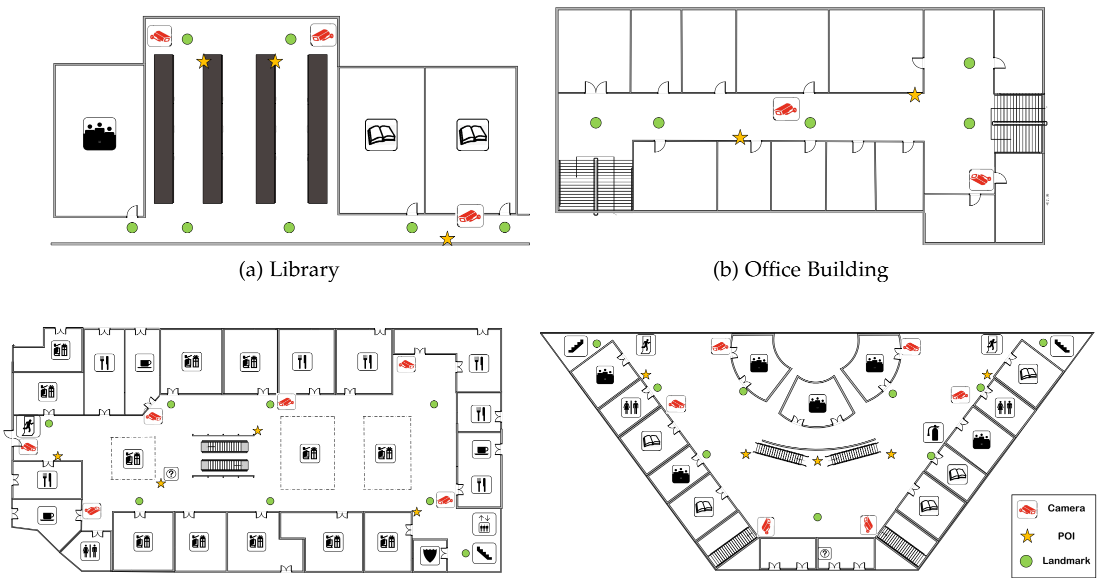
The *iSAT* system needs to be deployed before evaluation. With a centimeter-level-accuracy floorplan, 3D locations of all POI and surveillance cameras, as well as the division of functional areas, can be acquired with little effort. Then, all the data is put into the configuration file, *iSAT* automatically completes the initialization phase.

6.1.3 Evaluation Metrics and Ground Truth Acquisition

In experiments, 3 volunteers of different heights with different smartphone holding gestures are recruited. We principally test three aspects of performance about *iSAT*: localization accuracy, navigation success rate and 3D semantic map construction accuracy. To evaluate the localization performance, we focus on the initial location estimation bias, just like related works [21], [32]. We invite volunteers to take photos within surveillance cameras' view casually. Then, *iSAT* will calculate the volunteers' location by leveraging images captured from both smartphones and surveillance cameras; To evaluate the navigation performance, volunteers can choose any location, wherever POI can be captured, as the navigation starting point, and then select any location in the experimental area as the endpoint. Same as recent works [14], [33], we also set checkpoints at turns, escalators and some landmarks on each trajectory. The navigation success rate is defined as the rate of the users' successful arrival at the destination within a radius of 2 meters; And to evaluate the semantic map construction performance, as mentioned above, all the information about floorplan including 2D segmentation is predefined. After each set of experiments, we export all the 3D point cloud

TABLE 1
Evaluations in different scenarios

#	Scenarios	Size(m^2)	Cameras	Smartphones	Frames	Duration
1	Library	320	3	2	10.2k	3h in 2 days
2	Office Building	600	2	4	22.8k	6h in 3 days
3	Teaching Building	1,360	6	4	27.6k	4h in 3 days
4	Shopping mall	2,130	6	4	54.5k	6h in 3 days



(a) Library (b) Office Building
(c) Shopping Mall (d) Teaching Building

Fig. 6. Experimental areas

data and manually calculate the accuracy of semantic map construction.

6.1.4 Comparative Methods

To extensively evaluate the performance of *iSAT*, we additionally implement five different state-of-the-art indoor localization and navigation systems for comparison. All these systems are mainly based on mobile camera or surveillance camera.

1) **iVR** [21]: iVR is the most recent vision+radio+sensor tracking framework, which combines observations from surveillance cameras, WiFi radio signals and IMU data and outperforms the state-of-the-art system.

2) **PHADE** [20]: PHADE is a recent vision+sensor localization and tracking framework, which extracts human motion features from video and IMU sensors, and fuses both patterns to locate and identify different users.

3) **ClickLoc** [12]: ClickLoc is a typical high accurate localization system integrating mobile vision and IMU signals from smartphone.

4) **Pair-Navi** [14]: Pair-Navi is a real-time P2P navigation system based on a mobile camera, requiring no pre-installed infrastructure or pre-deployed localization services.

5) **Travi-Navi** [33]: Travi-Navi is a vision-guided P2P navigation system that enables a self-motivated user to deploy indoor navigation services without comprehensive indoor localization systems.

iSAT cannot only locate users but also navigate them to a certain destination with the help of a pre-known 2D floorplan. Our experiment with comparative systems includes two parts: localization and navigation. In the first part, *iSAT* is compared with iVR, PHADE, and ClickLoc; In the second part, *iSAT* is compared with Pair-Navi and Travi-Navi.

6.2 Component Study

6.2.1 Comparison of Different Features

In this subsection, we compare different feature points and tested their performance in different experimental scenarios. As shown in Table 2, we recorded the performance of SIFT, ORB, and BRISK, including location error, matching error, and time cost. All these experiments are conducted under different disparities (*i.e.* angle at POI) of 15°, 30°, and 45° respectively. Compared with ORB and SIFT, BRISK achieves better performance with an angle of over 30°, proving its effectiveness when processing images with large disparity. In addition, BRISK feature point extraction and matching speed are satisfactory and will not significantly impact the system latency.

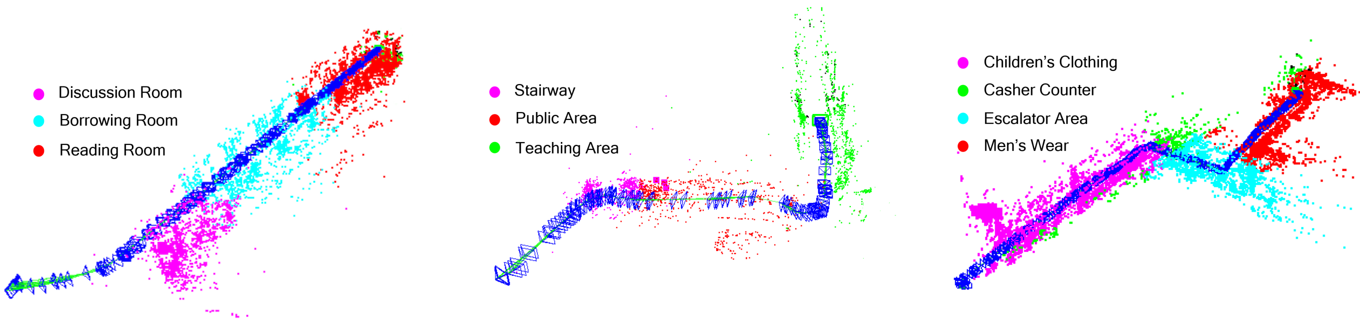
6.2.2 Performance of Real Scale Acquisition

Real scale acquisition module determines the localization accuracy, since it associates pixel changes with physical distance. Inaccurate scale will incur incorrect moving distance.

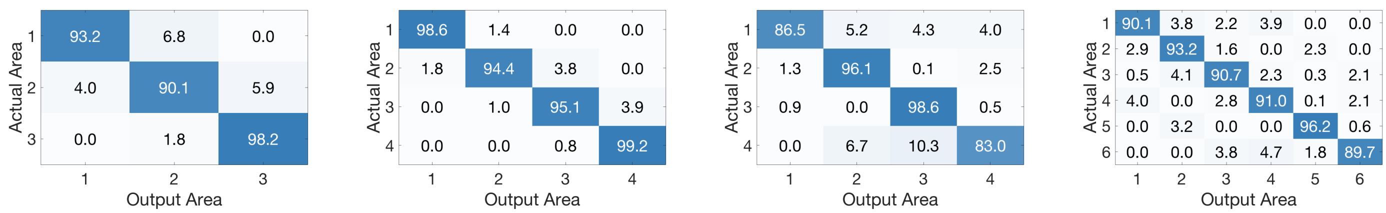
Figure 9 shows the real scale error rate in four different scenarios. We noticed that in the first three scenarios, the error rate is under 3%. While in shopping mall, due to the complex NLOS environment, the real scale error reached nearly 5%, which further leads to a degradation of localization and navigation performance. The result also reveals that using higher definition videos in complex environments can effectively improve system performance.

TABLE 2
Comparision of different features under different disparities

Name	Time (ms)	Matching Error (pixel)			Location Error (m)		
		15°	30°	45°	15°	30°	45°
SIFT	195.09	7.30×10^{-5}	1.93×10^{-2}	1.05×10^{-1}	**0.45**	1.47	1.98
ORB	18.49	3.42×10^{-4}	6.44×10^{-3}	5.83×10^{-2}	0.52	1.26	1.54
BRISK	42.42	2.35×10^{-4}	4.82×10^{-4}	2.44×10^{-3}	0.48	**0.66**	**1.01**



(a) Library (b) Teaching Building (c) Shopping Mall
Fig. 7. Semantic map constructed in different scenarios



(a) Library (b) Office Building (c) Teaching Building (d) Shopping Mall
Fig. 8. Confusion matrices of 3D point cloud map semantic segmentation in different scenarios

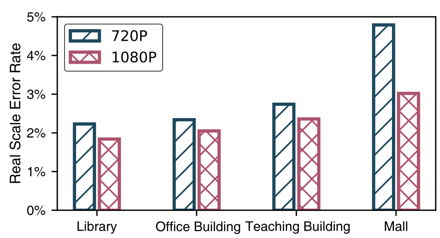


Fig. 9. Real scale error rate

6.2.3 Performance of 3D Semantic Map Construction

As discussed above, semantic 3D map construction, which improves the quality of indoor navigation services, is a key function of *iSAT*.

Figure 7 shows the 3D semantic map in different scenarios built by *iSAT*. To evaluate the effectiveness and accuracy of the map construction function, we mainly focus on the ability of *iSAT* to correctly classify 3D points into their real-world areas. As shown in Figure 8, the average accuracy of point cloud classification is 90.8%.

We notice that *iSAT* performs slightly worse in large-scale scenarios (*i.e.* teaching building and shopping mall) than it does in small-scale scenarios (*i.e.* library and office building). The reason is that errors in location and map construction accumulates as the user moves forward. Although the relocation function can effectively alleviate this problem, the accumulative errors cannot be completely avoided. Even though, the accuracy in all these four scenarios are above 88%, which proves the effectiveness of 3D semantic map construction. Unlike traditional methods to manually build a 3D semantic map which is labor-intensive and requires expensive devices, our method needs zero human start-up effort and can work automatically.

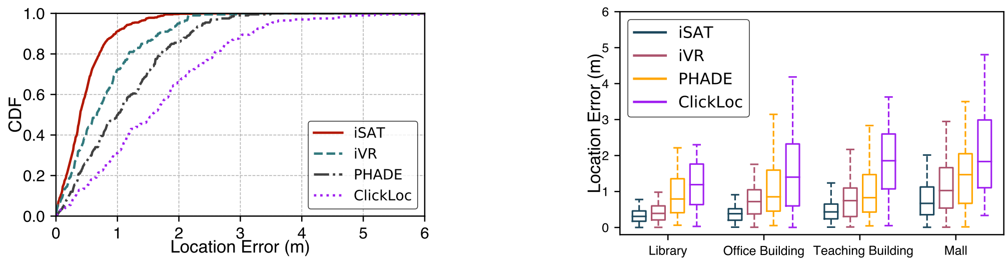
6.3 Performance Evaluation

6.3.1 Overall Performance Comparison

Figure 10a depicts the performance of the proposed *iSAT* as well as three other comparative systems in indoor localization scenarios. We observe that *iSAT* achieves the best performance among all the systems. The average localization accuracy of *iSAT* is $0.48m$ which outperforms iVR by 29.1%, PHADE by 56.3% and exceeds ClickLoc by 71.2%. As for the performance of navigation, the average navigation success rate of *iSAT* is 90.5%, which outperforms Pair-Navi by 2.8% and Travi-Navi by 5.7%.

We further evaluate the performance of *iSAT* and other comparative systems in different experimental scenarios. As shown in Figure 10b, *iSAT* outperforms iVR, PHADE and ClickLoc by at least 15%, 35% and 50% respectively in all experimental scenarios. As shown in Figure 12, the navigation success rate of *iSAT* exceeds Pair-Navi in three of four scenarios and outperforms Travi-Navi by at least 1.5% in all experimental scenarios.

The results demonstrate that *iSAT* achieves remarkable performance among state-of-the-art vision-based localization and navigation systems. The reason for this performance gain is two-fold: 1) Innate metric advantages of vision-based methods. As far as we know, *iSAT* is the first system that integrates mobile smartphone cameras



(a) Overall localization accuracy (b) Different methods in different scenarios

Fig. 10. Overall performance comparison with state-of-the-art systems

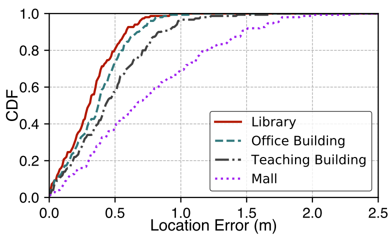


Fig. 11. Localization accuracy in different areas

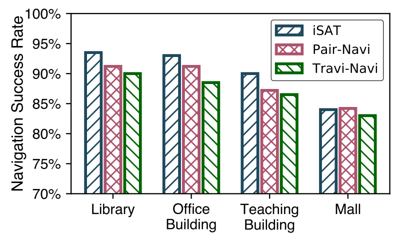


Fig. 12. Navigation success rate in different areas

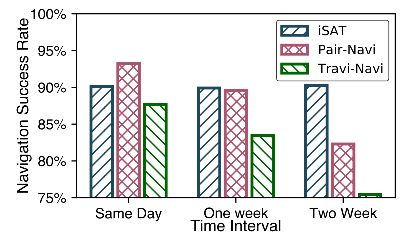


Fig. 13. Navigation success rate with different interval

and ambient surveillance cameras. Compared with previous radio-frequency-assisted or inertial-sensor-assisted systems which suffer from the metric error of localization typically averaging 3-5m due to intrinsic defects such as fluctuation of signal strength or inaccurate pedestrian dead-reckoning (PDR), vision-based methods enjoy more robustness and finer granularity. 2) Enhanced data fusion algorithm. Although existing works also take heterogeneous data (*e.g.* IMU, RF signal, images) as input, the fusing methods are loosely coupled: localization results are directly generated by individual systems independently, the localization bias introduced by each module will be accumulated and further reduce the performance [21]. In contrast, the data fusion algorithm of *iSAT* is tightly coupled: *iSAT* does not treat mobile vision-based and surveillance camera-based localization as two standalone sub-systems: the pixel-level information of frames captured by above different cameras are fused by leveraging BRISK feature points and multiview geometry algorithms, which will reduce the bias introduced by each sub-system and fusion procedure.

6.3.2 Performance in Different Areas

We evaluate the performance in four different experimental scenarios as illustrated in Figure 6, including a library, an office building, a teaching building and a large shopping mall. Figure 11 shows the performance of *iSAT* in different areas. We can see *iSAT* yields an average localization accuracy of 0.32m in the library, 0.38m in the office building, 0.47m in the teaching building, and 0.76m in the shopping mall. The corresponding 95th percentile location errors in these four scenarios are 0.69m, 0.75m, 0.98m, and 1.71m respectively. The result shows *iSAT* achieves sub-meter localization accuracy in all these four scenarios. As shown in Figure 12, *iSAT* achieves a navigation success rate of 93% in the library, 92% in the office building, 90% in the teaching building, while only 84% in the shopping mall.

Most existing pattern-matching and data-driven approaches rely on a domain-specific model, which means when applying these systems to different scenarios, the distribution of objects will significantly reduce the system performance. In contrast, *iSAT* is based on the general multi-view geometry principle which is environment-irrelevant. Once the surveillance camera and the mobile camera capture overlapping scenes (regardless of what the captured scene is), *iSAT* will calculate their geometric relationship and accurately get the user's location.

The degradation of localization and navigation performance in the shopping mall is mainly caused by a complex and crowded indoor environment, where the occlusion of POI inevitably occurs. Besides, the performance of visual odometry also decreases in a high-dynamic indoor environment. Overall, *iSAT* outperforms compared systems in all of these scenarios, proving its relatively high performance regardless of the environmental difference.

6.3.3 Performance at Different Time Intervals

A major drawback of the P2P navigation systems is that their performance degrades significantly as time flows. Since the indoor environments are dynamically and gradually changing over time, these systems may fail to match current video frames with the pre-recorded video frames perfectly. Fortunately, the navigation service of *iSAT* is location-based thus having no need for any pre-recorded data. With the help of surveillance cameras, *iSAT* performs well in any dynamic scenarios. We evaluated the navigation success rate of *iSAT* and comparative systems at different time intervals in teaching building. As shown in Figure 13, the navigation success rate of *iSAT* is almost time-invariant and stable at 90%. While the navigation success rate of Pair-Navi and Travi-navi decline 3% and 5% respectively after one week, and 10% and 13% respectively after two weeks. The results demonstrate that the performance of *iSAT* has robustness against time changes.

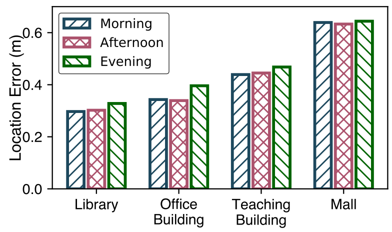


Fig. 14. Localization accuracy in different time

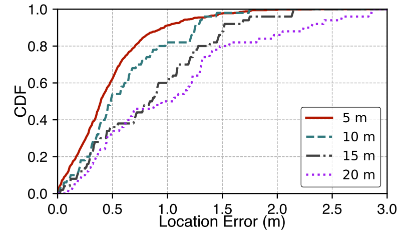


Fig. 15. Distance to POI

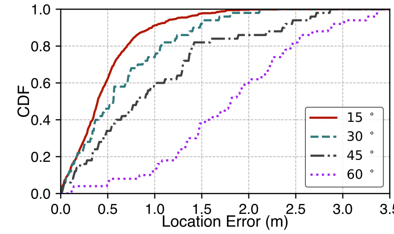


Fig. 16. Angle at POI

6.3.4 Performance at Different Times of the Day

We further test our system at different times of the day in four experimental scenarios. We conduct experiments at 10 a.m. in the morning, 3 p.m. in the afternoon and 8 p.m. in the evening. Figure 14 shows that there is a slight difference between morning and afternoon, while the localization errors are relatively large at night. Especially in the office building which undergoes the most drastic illumination change among the four areas, the localization error increases 15.2% from 0.33m to 0.38m. Generally, in the office building, most of the lights are usually turned down in the evening, which leads to a reduction in the number of feature points. So the image registration between surveillance camera and mobile camera may not be accurate, which could incur a slightly larger location error. For the same reason, the navigation success rate of the *iSAT* at night also decline to 88%, about 2.5% lower than that in the daytime.

6.3.5 Impact of the Relative Location of Cameras and POI

As mentioned in Section 2, POI triangulation is of vital importance to the localization accuracy of *iSAT*. The result of the triangulation is highly dependent on the relative location of the mobile camera, surveillance camera and POI. For ease of notion, we assume the surveillance cameras are fixed, and test the localization performance with different user-POI distance and different user-POI-surveillance camera angle respectively.

We first set the angle to 15° and evaluate the localization accuracy of *iSAT* at different distances. The experimental results shown in Figure 15 demonstrate that the average location errors of *iSAT* are 0.48m, 0.62m, 0.86m and 1.06m at distance 5m, 10m, 15m and 20m respectively. Longer distances result in higher localization biases since the depth estimation deviation of the POI in triangulation is proportional to the distance.

Then, we set the distance between the mobile camera and POI to 5m and test the localization accuracy of *iSAT* at different angles. Figure 16 shows that the average location errors of *iSAT* are 0.48m, 0.66m, 1.01m and 2.20m at 15°, 30°, 45° and 60° respectively, which indicates the average location error scales with increasing disparity between frames captured from mobile and surveillance cameras. According to the experimental results, *iSAT* achieves a satisfactory localization accuracy at the user-POI distance $< 15m$ and the user-POI-surveillance camera angle $< 45°$.

6.3.6 Impact of Video Resolution and Frame Rate

iSAT is a vision-based system, which means its localization and navigation performance is closely related to the quality of video captured by mobile camera. Thus we conduct multiple experiments using videos of 4 combinations of resolution and frame detection rate to examine the impacts of two main factors.

As shown in Figure 17, the location error of *iSAT* using 1080P video decreases 28% comparing with that using 720P video. *iSAT* using 60fps video reduces 9% average location error comparing with *iSAT* using 30fps video. Figure 18 shows the navigation success rate of *iSAT* with different video quality. The average navigation success rate of four types of videos are 95%, 93%, 88% and 85% respectively. Generally speaking, videos with higher resolution and frame detection rate enables higher localization accuracy and navigation success rate. Because high-resolution video frames usually contain more feature points, which can not only improve the registration accuracy between surveillance camera and mobile camera, but also reduce the error of inter-frame motion estimation during the navigation process. Video frames with a higher detection rate are less prone to be blurred, which contributes to fewer errors during the localization and navigation process.

6.3.7 Performance of Relocalization

As mentioned above, relocalization can reduce the probability of deviation caused by accumulative errors. To demonstrate the effectiveness of the relocalization function, we evaluate the navigation success rate of *iSAT* with and without relocalization. We select 4 paths of different lengths in the teaching building and conduct experiments. Figure 19 shows that navigation service with relocalization achieves a relatively high success rate compared with that without relocalization. *iSAT* with relocalization outperforms that without relocalization by 1%, 4%, 8% and 12% in 4 paths respectively. It can be concluded that the longer the path, the more effective relocalization is. We summarize the reason as that a relatively long path usually incurs a greater accumulative errors, and relocalization can eliminate the errors at one time, relocating the user to the correct path. The result also shows that when the relocalization function is disabled (means during the navigation process, no surveillance cameras and POI will be used) the navigation success rate in 50-meter-path still achieves over 93%, which proves *iSAT* can still achieve a good navigation performance only using one surveillance camera and POI at initial positioning stage.

6.3.8 System Latency

In *iSAT*, system end-to-end latency consists of communication latency and computation latency. The former depends on the quality of wireless links and resolution of transmitted frames; and the latter is introduced by the proposed localization and navigation algorithms. In this experiment, we first evaluate the communication latency of *iSAT*. Furthermore,

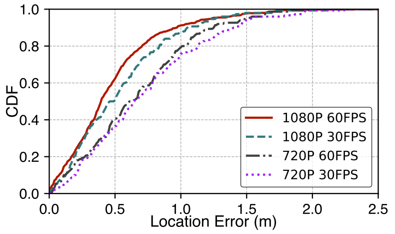


Fig. 17. Localization accuracy under different resolution and fps

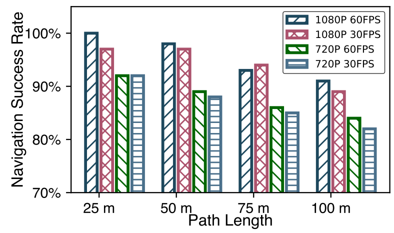


Fig. 18. Navigation success rate under different resolution and fps

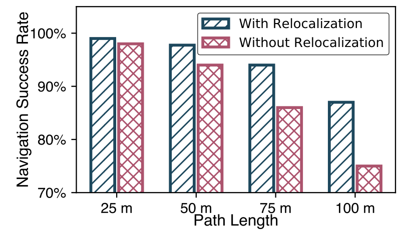


Fig. 19. Effectiveness of relocalization

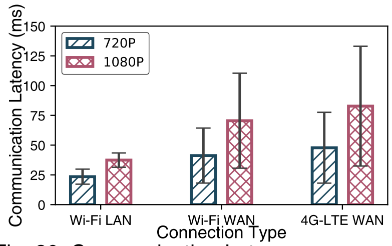


Fig. 20. Communication Latency

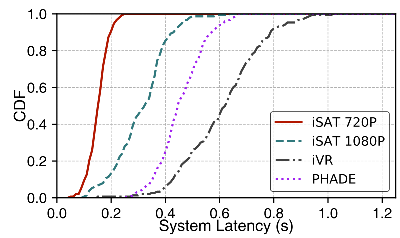


Fig. 21. Different System Latency

we compare the end-to-end latency of *iSAT* with iVR and PHADE by using 720P videos.

As shown in the following Figure 20, we choose three typical network connections: Wi-Fi based LAN, Wi-Fi based WAN, and 4G-LTE based WAN. Wi-Fi and 4G-LTE are two different wireless access methods that are commonly used by mobile devices. LAN connection means the client and server are in the same local network. The result shows the average communication latency is within $100ms$ in any case.

After receiving the data, the server extracts BRISK feature points of each frame, which takes $23ms$ for 720P frame and $50ms$ for 1080P frame on average. Then the POI detection module and VO module run in parallel, which takes $72ms$ for 720P frame and $183ms$ for 1080P frame, respectively. Finally, *iSAT* estimates the pose of the mobile camera and locates the user, which takes 28ms on average. As shown in Figure 21, the system latency of *iSAT* is about $0.16s$ and $0.23s$ respectively using 720P and 1080P video. Comparing with iVR and PHADE, *iSAT* reduces system latency by over 30% and 60% respectively.

In a nutshell, *iSAT* can achieve sub-meter or even decimeter level localization in no more than 0.5s, which meets the real-time requirement for being applied as an indoor localization system.

6.3.9 Application Power Consumption

To evaluate the power consumption of *iSAT* on the mobile device, we run our application on different types of smartphones for dozens of times, and the power consumption is recorded by the Android battery manager. On average, running our application on Galaxy S10 and Google Pixel for 15 minutes incurs a power consumption of 4.7% and 5.6% respectively. Such a delightful result benefits from the client-server design of *iSAT*, which means the mobile device only need to record the video frames and stream them to the server-side, and all the computational intensive tasks (*e.g.*, feature matching, coordinate transformation, *etc.*) are performed on the server.

7 Related Work

In this section, we briefly summarize the most related works in the following categories.

Mobile-vision-based Localization Compared with indoor localization techniques based on wireless or inertial information, prior works based on SfM (Structure from Motion) [34] take high-resolution images as input data thus achieving sub-meter accuracy [35]. Overlay [36] combines data retrieved from the camera and sensors of a smartphone to construct a geometric representation of environments. Apart from works mentioned above, ORB-SLAM [31], with DBoW2 [23] for place recognition and g2o [37] for optimization achieves excellent performance. Among other technique advancements, works of semanticSLAM aims to adopt semantic information into the framework [38], [39], [40], while WiFiSLAM [41] relates RSS fingerprints, which is further improved by GraphSLAM [42]. One significant limitation of SLAM framework is the scale ambiguity, thus requiring data from other sources such as IMU sensors and WiFi signals. However, our solution exploits SLAM's advantages of efficiency and map generation, and by introducing surveillance camera's real-world location, *iSAT* can get the scale of the real world, thus doesn't need other forms of data. Besides, with the real position of 3D points known, semantic information can be attached more easily, which allows the system to offer more detailed instructions during navigation. Furthermore, compared with SLAM-based navigation systems [14] and image database methods [12], using surveillance cameras means that even in time-variant scenarios, localization and navigation function can perform well, since the frames captured by the surveillance camera and the mobile camera are always time-synchronized.

Surveillance-camera-assisted Localization Places where indoor localization is needed generally provide extensive coverage of surveillance cameras and routers. Given these circumstances, pioneer works have integrated ambient information to achieve higher accuracy. RAVEL [43] leverages the fusion of visual data and radio data to realize high-accuracy localization. TAR [32] combines visual patterns from a surveillance camera with BLE signals to identify and track users. Another work [44] makes use of IMU tracking and video to realize stable pedestrian tracking. Similarly, PHADE [20] combines a unique pattern of pedestrian trajectories extracted from a surveillance camera and IMU data to identify and track different users. The latest work of iVR [21] explores the effect of fusing WiFi, IMU and visual signals and outperforms previous systems in the aspect of accuracy. Compared to prior works, our system doesn't need IMU data from the mobile device or any information from the WiFi signal. Another significant limitation of the above systems is that surveillance cameras can only work within a limited LOS range. Instead, our system is the first indoor localization system which exploits synchronized images from surveillance cameras and mobile camera. The VO module based on mobile camera ensures *iSAT* to navigate users successfully even if they are not in any surveillance cameras' view. Therefore, a full-coverage navigation service can be provided.

Easing Deployment Existing indoor navigation systems generally require a time-consuming site survey to obtain prior information. Previous research has made efforts to rid their systems of heavy human labor. [45] leverages WiFi fingerprints and motion information to achieve automatic construction of floorplan. Walkie-Markie [46] exploits user trajectories and WiFi-Marks calculated from RSS trends to label landmarks and reconstruct internal pathway maps. Some pioneer works make use of visual messages to gain information. Jigsaw [47] extracts geometric data and uses SfM to help construct a 2D floorplan. iVR [21] relies on images from a surveillance camera to construct the indoor map, and hence doesn't require the fingerprint database or digital floorplan. The relevant works design effective methods to construct an indoor map.

Still, most of them require either manual measurement of the external parameters (*i.e.* 3D position and 3D rotation) of the surveillance camera, or a constructed large image database which needs to be maintained and manually updated. In contrast, our system leverages the combination of surveillance and mobile camera to calculate projection, in which the surveillance camera serves as a robust and real-time image source. With floorplan which comes with the building itself, it takes little human effort to get the locations of POI and surveillance camera. Our system is based on the surveillance cameras that have already been deployed for security purposes, which means minimal additional overhead is required. Once deployed, our system doesn't need to be updated until the building structure changes.

8 Conclusion

In this paper, we present *iSAT*, an indoor navigation system that enables public cameras to 1) locate users with absolute locations on the floorplan, 2) tell users with semantic information about the environment, and 3) guide users with navigation instructions. We implement *iSAT* on commodity smartphones and conduct experiments in 4 different scenarios. Experiment results show that our system outperforms existing systems in localization accuracy and navigation success rate. We believe *iSAT* takes a promising step forwards to a practical navigation system. By leveraging *iSAT*, all areas with public cameras can upgrade to smart space with visual navigation services.

Acknowledgments

This work is supported in part by the NSFC under grant 61832010, 61872081, 61632013, 61972131.

References

[1] Z. Yang, Z. Zhou, and Y. Liu, "From rssi to csi: Indoor localization via channel response," *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, pp. 1–32, 2013.

[2] C. Wu, J. Xu, Z. Yang, N. D. Lane, and Z. Yin, "Gain without pain: Accurate wifi-based localization with fingerprint spatial gradient," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Sep 11-15 2017.

[3] D. Li, J. Xu, Z. Yang, Y. Lu, Q. Zhang, and X. Zhang, "Train once, locate anytime for anyone: Adversarial learning based wireless localization," in *Proceedings of the IEEE INFOCOM*, May 10-13 2021.

[4] J. Wang and D. Katabi, "Dude, where's my card? RFID positioning that works with multipath and non-line of sight," in *Proceedings of the ACM SIGCOMM*, 2013.

[5] L. Shangguan, Z. Yang, A. X. Liu, Z. Zhou, and Y. Liu, "Stpp: Spatial-temporal phase profiling-based method for relative rfid tag localization," *IEEE/ACM Transactions on Networking*, 2017.

[6] Z. Yang, C. Wu, Z. Zhou, X. Zhang, X. Wang, and Y. Liu, "Mobility increases localizability: A survey on wireless indoor localization using inertial sensors," *ACM Computing Surveys*, 2015.

[7] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," in *Proceedings of ACM Mobicom*, 2014.

[8] P. Dollár, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *Computer Vision–ECCV 2012*. Springer, 2012.

[9] B. Liu, J. Huang, L. Yang, and C. Kulikowsk, "Robust tracking using local sparse appearance model and k-selection," in *Proceedings of the IEEE CVPR*, 2011.

[10] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part i: The first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, 2011.

[11] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, 2016.

[12] H. Xu, Z. Yang, Z. Zhou, L. Shangguan, K. Yi, and Y. Liu, "Indoor localization via multi-modal sensing on smartphones," in *Proceedings of the ACM UbiComp*, 2016.

[13] Z. Yin, C. Wu, Z. Yang, and Y. Liu, "Peer-to-peer indoor navigation using smartphones," *IEEE Journal on Selected Areas in Communications*, 2017.

[14] E. Dong, J. Xu, C. Wu, Y. Liu, and Z. Yang, "Pair-navi: Peer-to-peer indoor navigation with mobile visual slam," in *Proceedings of the IEEE INFOCOM*, 2019.

[15] J. Xu, G. Chi, Z. Yang, D. Li, Q. Zhang, Q. Ma, and X. Miao, "Followupar: Enabling follow-up effects in mobile ar applications," in *Proceedings of the ACM MobiSys*, June 24-July 2 2021.

[16] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao, "Travi-navi: Self-deployable indoor navigation system," in *Proceedings of the ACM MobiCom*, 2014.

[17] Y. Shu, K. G. Shin, T. He, and J. Chen, "Last-mile navigation using smartphones," in *Proceedings of ACM MobiCom*, 2015.

[18] N. Jenkins, "245 million video surveillance cameras installed globally in 2014," *IHS Technology*, 2015.

[19] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[20] S. Cao and H. Wang, "Enabling public cameras to talk to the public," in *PACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Sep 11-15 2018.

[21] J. Xu, H. Chen, K. Qian, E. Dong, M. Sun, C. Wu, L. Zhang, and Z. Yang, "ivr: Integrated vision and radio localization with zero human effort," *Proceedings of the ACM UbiComp*, 2019.

[22] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International conference on computer vision*. IEEE, 2011.

[23] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, 2012.

[24] L. S. Shapiro, A. Zisserman, and M. Brady, "3d motion recovery via affine epipolar geometry," *International Journal of Computer Vision*, 1995.

[25] G. Xu and Z. Zhang, *Epipolar geometry in stereo, motion and object recognition: a unified approach*. Springer Science & Business Media, 2013, vol. 6.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981.

[27] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual slam algorithms: a survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, 2017.

[28] S. Li, C. Xu, and M. Xie, "A robust o (n) solution to the perspective-n-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

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

[30] "Ros kinetic kame," http://wiki.ros.org/kinetic.

[31] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, 2015.

[32] X. Liu, Y. Jiang, P. Jain, and K.-H. Kim, "Tar: Enabling fine-grained targeted advertising in retail stores," in *Proceedings of the ACM Mobisys*, 2018.

[33] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao, "Travi-navi: Self-deployable indoor navigation system," *IEEE/ACM Transactions on Networking*, 2017.

[34] J. J. Koenderink and A. J. Van Doorn, "Affine structure from motion," *JOSA A*, 1991.

[35] R. Mautz and S. Tilch, "Survey of optical indoor positioning systems," in *International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2011.

[36] P. Jain, J. Manweiler, and R. Roy Choudhury, "Overlay: Practical mobile augmented reality," in *Proceedings of the ACM MobiSys*, 2015.

[37] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation*. IEEE, 2011.

[38] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. Montiel, "Towards semantic slam using a monocular camera," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011.

[39] H. Abdelnasser, R. Mohamed, A. Elgohary, M. F. Alzantot, H. Wang, S. Sen, R. R. Choudhury, and M. Youssef, "Semanticslam: Using environment landmarks for unsupervised indoor localization," *IEEE Transactions on Mobile Computing*, 2015.

[40] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *IEEE International Conference on Robotics and Automation*. IEEE, 2017.

[41] B. Ferris, D. Fox, and N. D. Lawrence, "Wifi-slam using gaussian process latent variable models." in *International Joint Conferences on Artificial Intelligence Organization*, 2007.

[42] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor wifi graphslam," in *IEEE International Conference on Robotics and Automation*. IEEE, 2011.

[43] S. Papaioannou, H. Wen, A. Markham, and N. Trigoni, "Fusion of radio and camera sensor data for accurate indoor positioning," in *Proceedings of the IEEE MASS*, 2014.

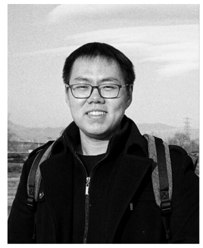
[44] W. Jiang and Z. Yin, "Combining passive visual cameras and active imu sensors for persistent pedestrian tracking," *Journal of Visual Communication and Image Representation*, 2017.

[45] Y. Jiang, Y. Xiang, X. Pan, K. Li, Q. Lv, R. P. Dick, L. Shang, and M. Hannigan, "Hallway based automatic indoor floorplan construction using room fingerprints," in *Proceedings of the ACM UbiComp*, 2013.

[46] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-markie: Indoor pathway mapping made easy," in *Proceedings of the USENIX NSDI*, 2013.

[47] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li, "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *Proceedings of the ACM MobiCom*, 2014.

Guoxuan Chi received his B.E. degree in School of Information and Communication Engineering from Beijing University of Posts and Telecommunications in 2019. He is now a PhD student in School of Software, Tsinghua University. His research interests include Internet of Things and mobile computing.

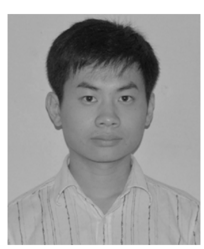


Jingao Xu received his B.E. degree in School of Software from Tsinghua University in 2017. He is now a PhD student in School of Software, Tsinghua University. His research interests include Internet of Things and mobile computing.

Jialin Zhang received his B.E. degree in Software Engineering from Tsinghua University in 2020. He is now a graduate student for a Master's degree in School of Software, Tsinghua University. His research interests include Internet of Things, indoor navigation and mobile computing.

Qian Zhang is a Postdoc researcher in School of Software, Tsinghua University, China. She received her Ph.D. degree in School of Computer Science, Beijing Institute of Technology, China, her MS degree in Computer Science, Illinois Institute of Technology. Her research interests include mobile computing, smart sensing, and mobile crowd sensing.

Qiang Ma received his BS degree in Department of Computer Science and Technology from Tsinghua University, China, in 2009, and Ph.D. degree in Department of Computer Science and Engineering at the Hong Kong University of Science and Technology in 2013. He is now an assistant researcher in the Software School of Tsinghua University. His research interests include wireless sensor networks, mobile computing and privacy.



Zheng Yang is an associate professor at Tsinghua University. He received a B.E. degree in computer science from Tsinghua University in 2006 and a Ph.D. degree in computer science from Hong Kong University of Science and Technology in 2010. His main research interests include Internet of Things and mobile computing. He is the PI of National Natural Science Fund for Excellent Young Scientist and has been awarded the State Natural Science Award (second class).