ImPos: An Image-Based Indoor Positioning System

Yunzhi Li*, Rajeswari Hita Kambhamettu†, Yidan Hu‡, and Rui Zhang*
*Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716
†Information Systems, Carnegie Mellon University, Pittsburgh, PA 15213
‡ Department of Computing Security, Rochester Institute of Technology, Rochester, NY 14623
liyunzhi@udel.edu, rkambham@andrew.cmu.edu, yidan.hu@rit.edu, ruizhang@udel.edu

***Abstract*—Recent years have witnessed growing interests from both academia and industry in developing effective Indoor Positioning Systems (IPSes). An IPS allows users to learn their locations and navigate in large unfamiliar indoor venues such as shopping malls, hospitals, and airports, where GPS signals are often absent or unreliable. Among different types of IPSes, images-based IPSes estimate a user's location from one or more pictures the user took of nearby landmarks, which explore the deep penetration of smartphones into people's everyday life and do not require any costly infrastructure upgrade. While several image-based IPSes have been proposed in the literature, most of them suffer from large processing delay due to computationally intensive 3D reconstructing or low positioning accuracy caused by inaccurate angle estimation. In this paper, we introduce the design and evaluation of ImPos, a novel image-based IPS that achieves high positioning accuracy by improved angle estimation and fully utilizing all recognized landmarks. Detailed experiment studies confirm the significant advantages of ImPos over prior image-based solutions.**

I. Introduction

Indoor positioning systems (IPSes) are indispensable for users to navigate in large and unfamiliar indoor venues such as airports, hospital complexes, and shopping malls. The lack of reliable GPS signals in indoor venues makes accurate indoor positioning a non-trivial task and has prompt the design and development of various types of IPSes based on different techniques over the past two decades. Existing IPSes include WiFi fingerprint-based IPS [1], [2], WiFi Channel State Information (CSI)-based IPS [3], Bluetooth-based IPS [4], image-based IPS [5], [6], Radio Frequency Identification (RFID)-based IPS [7], sound-based IPS [8], [9], visible light-based IPS [10], and so on. The global IPS market is expected to reach 256.59 billions in 2028 [11].

As a promising type of IPS, image-based IPSes are particularly suitable for venues without WiFi infrastructure. In a typical image-based IPS, the IPS server identifies a number of landmarks in the indoor venue. A user only needs to take a few pictures of his surrounding using his smartphone and submits the images to the IPS server. The landmarks in the query images are recognized and matched with the ones stored at the server. The IPS server can then estimate the user's position based on the locations of the recognized landmarks via 3D model reconstruction [12]–[15], triangulation [5], [6], or trilateration [16].

We observe that existing image-based IPSes suffer from two main limitations. First, prior solutions based on 3D model reconstruction [12]–[15] incur expensive setup costs and long processing delays for users' location queries. Second, existing solutions based on triangulation [5], [6] and trilateration [16] suffer from low positioning accuracy. For example, the state-of-art image-based IPS Sextant [5] reports a median positioning error of 3.6 m largely due to inaccurate angle measurements smartphone compass readings. Third, since triangulation and trilateration only require three recognized landmarks but query images may capture more than three landmarks, different combinations of three landmarks may result in different positioning accuracy. How to choose the best three landmarks remains unclear despite some heuristic guidelines. These situations call for efficient image-based IPSes with high positioning accuracy.

To tackle this challenge, we introduce the design and evaluation of ImPos, a novel image-based IPS based on triangulation with much improved positioning accuracy. The key idea behind the ImPos is a novel method for accurate angle estimation that jointly considers smartphone compass readings and the positions of the landmarks in the query image. Moreover, instead of trying to select the best three landmarks, we formulate the position estimation as an optimization problem to fully utilize all the recognized landmarks to improve positioning accuarcy. By doing so, ImPos achieves a much higher positioning accuracy than the state-of-art image-based IPS. Our contributions in this paper can be summarized as follows.

- We introduce ImPos, a novel image-based IPS with much improved positioning accuracy.
- We propose a novel method for angle estimation that jointly consider smartphone compass readings and image processing.
- Experiment studies based on a prototype confirm the advantages of ImPos over prior image-based IPSes. For example, our experiment results show that ImPos achieves a median distance error at 1.4 m in contrast to the 3.6 m reported in [5].

The rest of this paper is structured as follows. Section II discusses the related work. Section III presents the design of ImPos. Section IV reports our experiment results. This paper is finally concluded in Section V.

II. Related Work

In this section, we review some works that are most germane to our work.

Image-based IPSes have attracted significant attentions in the past two decades because they do not require any costly infrastructure upgrade but explore the ubiquitous presence of smartphones in people's daily life. Existing image-based IPSes can be broadly divided into two categories. The first category utilizes 3D modeling and locates a user by constructing the 3D model of the landmark and projecting the model into query images through machine learning techniques. Arnold *et al.* [12] proposed a compressed 3D representation for landmarks then project the 3D model to the query image to locate the user. Kawaji *et al.* [17] used omnidirectional panoramic images and KNN methods for comparison to recognize the landmarks. Ke *et al.* [18] achieved high positioning accuracy by improving the accuracy of image matching through an advanced RANdom SAmple Consensus (RANSAC) algorithm. In [13], Tommaso *et al.* trained a regression forest model for 2D image to 3D model correspondences whereby to predict the camera pose. InLoc [14] explores dense matching rather than local features to calculate the similarities between the query image and landmark images and then infer the user's location according to landmark orientation depth on images. Lu *et al.* [15] coped WiFi with the 3D model of the landmark to recognize the landmark quickly. However, these solutions suffer from costly 3D model construction and large processing latency for location queries because the server must search the matching feature points among million of feature descriptors on the landmark model to construct the 3D model of the landmark then project the 3D model to the query image.

The second category estimates a user's location from recognized landmarks using geometric approaches. MoVips [19] calculates the ratio of the distance between the same pair of feature points shown on the query and landmark image and infers the user's distance to the landmark based on the focal length. Hamed *et al.* [20] used a weighted KNN with Epipolar Geometry to infer the user's location from recognized the landmarks. Sextant [5] obtains the angles between each capturing gesture from the smartphone interior sensors then calculates the user's location by triangulation. Guan *et al.* [21] considered both user's height and sight region to infer the user's relative location to the landmark. Yuanqing *et al.* [22] obtained the user's trace from the pedestrian dead reckoning(PDR) system and then paired the trace with the stored guider's trace HAIL [23] infers a possible region with a predefined compass error and then searches the user's accurate position in the region with the landmarks cope with compass and gyroscope readings. Jiang *et al.* [24] uses images to identifies users' rough position and facing direction, then locate the user with the help of dead reckoning systems. However, these solutions suffer low positioning accuracy due to inaccurate angle or distance estimation.

Besides image-based IPSes, there are also several other types of IPSes based on different technologies. For example, WiFi received signal strength (RSS)-based IPSes [1], [2] explore the distinguishable WiFi received signal strengths to serve as the location fingerprints to estimate a user's location. FILA [3] utilizes WiFi CSI signals as location fingerprints. In [25], Bose *et al.* used the path loss model to calculate the distance between the user and WiFi access point then uses trilateration to locate the user. As another example, Fischer *et al.* [4] utilize the angle of arrival (AOA) and time of arrival (TOA) of Bluetooth signal to infer the distance between the Bluetooth access point and a user whereby to locate the user through trilateration. Similar to RADAR, Wang *et al.* used Bluetooth RSS signals as fingerprints in [26]. There are also IPSes that explore acoustic signals [8], [9], visible light [10], or RFID signals [7] to locate a user. However, these solutions typically require extra infrastructures which may not be readily available in many indoor venues.

III. Design of ImPos

In this section, we first give an overview of ImPos and then detail its design.

A. Overview

ImPos works in two phases, the offline phase and online localization phase. In the offline phase, the server constructs a landmark database by taking images of all landmarks and recording their physical coordinates on the floor. In the online localization phase, the server answers location queries from users based on the received query images and the constructed landmark database. Specifically, different from prior image-based IPSes [5], [23], which estimate angles only considering camera headings and identify user's location using angle-based localization techniques, ImPos jointly considers the camera headings and the positions of the landmarks in the query images to estimate angles with high accuracy. Moreover, we formulate localization as an optimization problem instead of directly applying the naive triangulation techniques.

In what follows, we first introduce how to construct the landmark database offline. Next, we introduce how the server determines a users location which mainly includes an algorithm for angle estimation, and an localization mechanism by solving the formulated optimization problem.

B. Offline Landmark Database Construction

We first select N landmarks $\mathcal{L} = \{L_1, L_2, \cdots, L_N\}$ in the indoor venue. Next, we take one image for each landmark $L_i \in \mathcal{L}$ and record its physical coordinate, $(x_i, y_i) \in \mathcal{D}$, where $\mathcal{D}$ is the domain of the physical coordinate. For each landmark image, we extract a feature vector, $F_i = \{\langle f_{i,1}, (p_{i,1}, q_{i,1})\rangle, \cdots, \langle f_{i,m_i}, (p_{i,m_i}, q_{i,m_i})\rangle\}$ using the classical Speeded Up Robust Features (SURF) algorithm [27], where $f_{i,j}, (1 \le j \le m_i)$, is a feature point, $(p_{i,j}, q_{i,j})$ is the pixel position of the feature point, and m_i is the number of feature points in F_i. The landmark database can be represented by $\{\langle (x_i, y_i), F_i\rangle | 1 \le i \le N\}$.

C. Online Positioning

In the online positioning phase, on receiving a location query from a user, the server estimates the user's location via image processing and triangulation. Assume that a user issues a location query $Q = \{(I_1, c_1), \cdots, (I_n, c_n)\}$, where each I_i,

$1 \leq i \leq n$, is a query image taken by the users, and c_i is the corresponding compass reading recorded when taking I_i. We assume that every query image I_i includes at least one landmark.

Upon receiving the location query Q, the server processes the query in three steps. First, the server identifies the landmarks includes in each query image and obtain their physical coordinate according to the landmark database. Next, the server estimates the *intersection angle* between the line from the user to a landmark and the line from the user to another landmark. Finally, the server uses the recognized landmark coordinate and the estimated intersection angles to estimate the user's location. In what follows, we detail these steps.

1) Landmark Recognition: Given a query $Q = \{(I_1, c_1), \cdots, (I_n, c_n)\}$, the server first extracts the feature vector from each image. For every image $I_i, 1 \leq j \leq n$, the server converts it into $I_i = \{\langle f_{i,1}, (p_{i,1}, q_{i,1})\rangle, \cdots, \langle f_{i,k_i}, (p_{i,k_i}, q_{i,k_i})\rangle\}$, where $f_{i,j}$ is a feature point and $(p_{i,j}, q_{i,j})$ is the pixel position of the feature point, and k_i is the number of feature points in image I_i for all $1 \leq j \leq k_i$.

For each query image I_i, the server identifies the landmarks it captures as follows. First, we compare the feature vector of the query image I_i with the feature vector of every landmark $L_j \in \mathcal{L}$. Specifically, for each landmark L_j with feature vector F_j, we calculate its matching degree with the query image I_i as follows. Assume that F_j has m_j feature points, we then have $k_i m_j$ possible feature point pairs. We adopt the standard feature matching techniques [27] to determine whether every pair of feature points are a match. The total number of the matched pairs $n_{i,j}$ is deemed as the matching degree with respect to query image I_i and landmark L_j. If the matching degree is larger than a threshold θ, we consider that the landmark L_j matches the query image I_i, and the query image I_i captures landmark L_j with high probability.

Second, we refine the matched landmarks by filtering out possible duplicate matches. Since some landmarks may share similar shapes and features, it is possible that the same set of features in the query image are matched to multiple landmarks. Let $\mathcal{L}_i \subseteq \mathcal{L}$ be the set of matched landmarks in image I_i identified the first step. For each landmark $l_j \in \mathcal{L}_i$ with a set of matched feature points $\{\langle f_{j,1}, (p_{j,1}, q_{j,1})\rangle, \cdots, \langle f_{j,n_{i,j}}, (p_{j,n_{i,j}}, q_{j,n_{i,j}})\rangle\}$, we compute the pixel position of the landmark l_j in image I_i as

$$\begin{cases} p_{i,j} & = \frac{1}{n_{i,j}} \sum_{x=1}^{n_{i,j}} p_{j,x} \\ q_{i,j} & = \frac{1}{n_{i,j}} \sum_{x=1}^{n_{i,j}} q_{j,x} \end{cases}. \quad (1)$$

For every pair of landmarks $L_j, L_k \in \mathcal{L}_i$, we compute their pixel distance as

$$d(j,k) = \sqrt{(p_{i,j} - p_{i,k})^2 + (q_{i,j} - q_{i,k})^2}\ .$$

If $d(j,k)$ is below a predetermined threshold, we consider L_j and L_k are matched to the same set of feature points and remove the one with lower matching degree from $\mathcal{L}_i$.

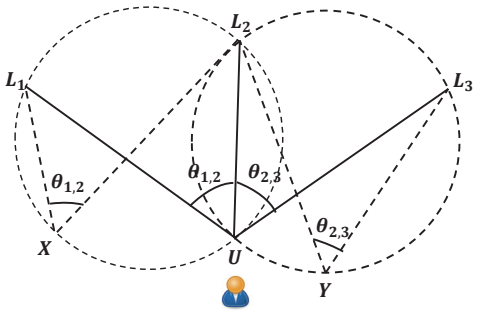

Fig. 1: Illustration of localization via triangulation.

2) Intersecting Angle Estimation: In this subsection, we first introduce the classical triangulation technique and then present a novel method for estimating the intersecting angles.

Triangulation is a classical method to estimate a user's position from the positions of three landmarks and two intersecting angles. Consider Fig. 1 as an example, in which three landmarks L_1, L_2 and L_3 are placed in a clockwise order. Suppose that we want to estimate a user's position U. Assume that we know the two intersecting angles $\angle L_1 U L_2 = \theta_{1,2}$ and $\angle L_2 U L_3 = \theta_{2,3}$. We can find a point X such that $\angle L_1 X L_2 = \theta_{1,2}$ and draw a circumscribed circle that passes through L_1, L_2 and X. It is easy to see that any point on the arc $\overset{\frown}{L_1 X L_2}$ has a circle angle of $\theta_{1,2}$. Similarly, we can find another point Y such that $\angle L_2 Y L_3 = \theta_{2,3}$ and draw a circumscribed circle that passes through L_2, L_3, and Y so that any point on the arc $\overset{\frown}{L_2 Y L_3}$ has a circle angle of $\theta_{1,2}$. Therefore, the user's position U must be the intersection point of the two circles.

Accurate measurement of the intersecting angles from users to the recognized landmarks is key for accurate location estimation via triangulation. Prior works [5], [23] all directly use the compass readings recorded by the smartphone when taking the picture as the direction from the user to the landmarks. We find that such angle measurements are inaccurate because the camera may not be perfectly facing the landmark when picture is taken. The difference between the true direction towards the landmark and the reported direction from the compass reading is commonly referred to as *angle drift*. Large angle drifts would introduce large errors to the estimated intersecting angles and lead to large positioning error. In what follows, we introduce an effective method to estimate such angle drifts.

Consider Fig. 2 as an example in which two reference lines pass X and Y, respectively. Suppose that the user is at position U when taking a picture of landmark L and that the camera's position is at point F. The true direction from the user to the landmark can be represented by $\angle LUY$. Now suppose that the compass reading is c which is the angle $\angle HFX = \angle HUY$. The angle drift is thus $\angle HUL = \alpha$, which is difficult to directly estimate.

Our key idea is that the angle drift α can be approximated by angle $\angle HFL = \gamma$. The reason is that the length of the line

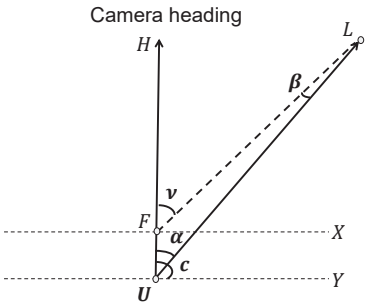

Fig. 2: Illustration of angle drift.

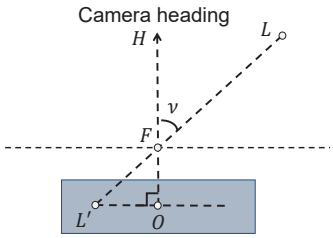

Fig. 3: Angle drift estimation

segment FU is approximately the user's arm length, which is about tens of centimeter and much smaller than the distance between the user and the landmark LU, which is usually several meters or more in practice. Therefore, the angle $\angle FLU$ is very small in practice. Since $\gamma = \alpha + \beta$, we have $\gamma \approx \alpha$.

We now discuss how to estimate angle γ from the landmark L's pixel position in the image. Consider Fig. 3 as an example, in which F is the camera's focal point, O is the the optical center of the camera, and L' is the position of the landmark L in the image. We can see that $\angle L'FO = \gamma$ and that triangle $\triangle L'OF$ is a right triangle. It follows that

$$\gamma = \arctan \frac{\overline{OL'}}{\overline{FO}} \ , \tag{2}$$

where $\overline{FO}$ is the focal length of the camera and $\overline{OL'}$ can be computed from the pixel size and the pixel distance between L' and O.

For every image I_i and every landmark $L_j \in \mathcal{L}_i$, we apply the above method to find angle drift $\gamma_{i,j}$. Denote by $\theta_{i,j}$ the direction from the user to landmark L_j according to image I_i for all $1 \le i \le n$ and $j \in \mathcal{L}_i$. We compute each $\theta_{i,j}$ as

$$\theta_{i,j} = \begin{cases} c_i - \gamma_{i,j} & \text{if landmark } L_j \text{ is on the left,} \\ c_i + \gamma_{i,j} & \text{if landmark } L_j \text{ is on the right,} \end{cases} \tag{3}$$

where c_i is the compass reading recorded when taking image I_i.

Since the same landmark may be captured in multiple images, we further estimate the direction to each recognized landmarks by taking the average of multiple estimations. Let $\mathcal{L}_q = \bigcup_{i=1}^{n} \mathcal{L}_i$ be the set of all the recognized landmarks. For each landmark $L_j \in \mathcal{L}_q$, we define $\mathcal{I}_j = \{i | L_j \in \mathcal{L}_i, 1 \le i \le n\}$ as the set of images that contain L_j and denote by θ_j as the direction from user towards L_j. We compute each θ_j as

$$\theta_j = \frac{\sum_{j \in \mathcal{I}_j} \theta_{i,j}}{|\mathcal{I}_j|} \tag{4}$$

3) User Positioning: Given a set of recognized landmarks $\mathcal{L}_q$, where each landmark $L_j \in \mathcal{L}_q$ is at position (x_j, y_j) and estimation θ_j, we formulate the location estimation as an optimization problem. Specifically, for every pair of landmarks $L_j, L_k \in \mathcal{L}_q$, we define their estimated intersecting angle as

$$\phi_{j,k} = |\theta_j - \theta_k| \ , \tag{5}$$

where θ_j and θ_k are the estimated directions from the user to L_j and L_k, respectively. Assume the user's location is at (x, y). We can draw a triangle formed by the user's location and two landmarks. Consider Fig. 4 as an example. The intersecting angle can be computed as

$$\phi_{j,k}(x, y) = \arccos \frac{d_j^2 + d_k^2 - d_{j,k}^2}{2 d_j d_k} \ , \tag{6}$$

where

$$d_j = \sqrt{(x_j - x)^2 + (y_j - y)^2},$$
$$d_k = \sqrt{(x_k - x)^2 + (y_k - y)^2},$$
$$d_{j,k} = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \ .$$

Ideally, the user's true location (x, y) should minimize the difference between the measured intersecting angle $\phi_{j,k}$ and the computed intersecting angle $\phi_{j,k}(x, y)$.

We now introduce two optimization problem formulation for estimating the user's location based on the above observation. Our first optimization problem formulation seeks to minimize the total square difference between the measured and computed intersecting angles for all pairs of recognized landmarks, which is given by

$$\begin{aligned} \text{Minimize} \quad & \sum_{j \in \mathcal{L}_q} \sum_{k \in \mathcal{L}_q} \left(\phi_{j,k}(x, y) - \phi_{j,k}\right)^2 \ , \\ \text{Subject to} \quad & (x, y) \in \mathcal{D} \ , \end{aligned} \tag{7}$$

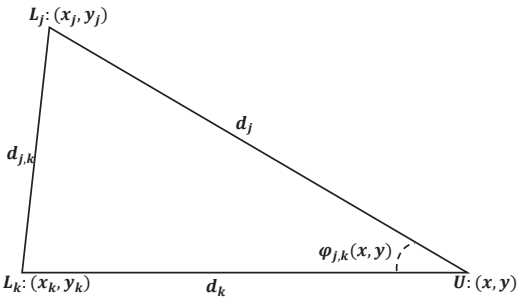

Fig. 4: An example of the triangle formed by the user's location and two landmarks

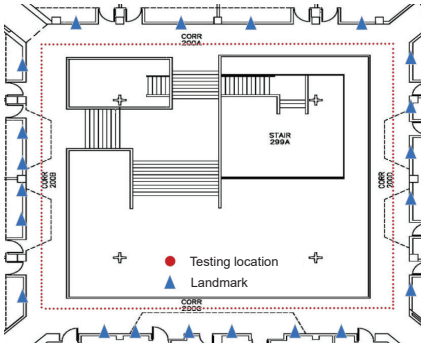

Fig. 5: The floor plan of an office building

where $\mathcal{D}$ denotes the set of possible locations in the indoor venue after proper discretization.

While the first formulation is intuitive, it suffers from one limitation. In particular, we notice that the change in the user's location has different impact on the computed intersecting angles of different pairs of recognized landmarks. Specifically, if the user is far away from both landmarks, then the change in the user's location has a relatively small impact on the computed intersecting angle, so minimizing the square difference between the measured and computed intersecting angles may require a large change in the user's location and thus should be given a smaller weight. On the other hand, if the user is very close to one of the two landmarks, then a small change in user's location would have a very large impact on the computed intersecting angle, which should be given a higher weight. While we don't know the user's location in advance, we find that the smaller the measured intersecting angle and the larger the distance between the two landmarks, the more likely that the user is far away from both landmarks, and vice versa. Based on this observation, we define a weight for every pair of landmarks L_j and L_k as

$$w_{j,k} = \frac{\phi_{j,k}}{\sqrt{(x_j - x_k)^2 + (y_j - y_k)^2}} \tag{8}$$

and define the cost function as

$$f(x,y) = \sum_{j \in \mathcal{L}_q} \sum_{k \in \mathcal{L}_q} w_{j,k} (\phi_{j,k}(x,y) - \phi_{j,k})^2 \ . \tag{9}$$

Our second optimization problem formulation seeks to minimize the total weighted square difference between the measured and computed intersecting angles for all pairs of recognized landmarks, which is given by

$$\begin{aligned} \text{Minimize} \quad & f(x,y) \\ \text{Subject to} \quad & (x,y) \in \mathcal{D} \ . \end{aligned} \tag{10}$$

Both optimization problems can be solve via exhaustive search.

IV. Performance Evaluation

In this section, we evaluate the performance of ImPos through experimental studies.

A. Data Collection and Experiment Settings

We implement a prototype of the ImPos in Android Studio/Java and test it on a Huawei P30 smartphone, which is equipped with a 2.8 mm physical focal length camera and SONY IMX600 camera sensor. The resolution of image is 2736×3648, and the pixel size for SONY IMX600 $1\mu m$.

We deploy the prototype system in an office building of 400 m^2, and Fig. 5 shows the floor plan. As we can see, we take $N = 21$ posters on the wall as landmarks. For every landmark, we record its coordinate, take a picture of it, and extract the SURF features from the image. We then choose 216 testing locations and take five images and record the corresponding compass readings at each testing location.

We compare the performance of ImPos with the Sextant system [5], which is the state of art image-based IPS that uses triangulation. In Sextant, the IPS server recognizes the landmarks from images and obtains the intersecting angles directly from the compass readings. If more than three landmarks are recognized, Sextant first estimates a rough location of the user and then chooses the three landmarks that are closest to the rough location to estimate the user's location via triangulation.

We use the following two performance metrics for our evaluation.

- **Angle error**: For every image I_i taken at testing location $l = (x, y)$ and every recognized landmark L_j at location (x_j, y_j), the true direction from the testing location l to landmark L_j is given by

  $$\bar{\theta}_{i,j} = \arctan \frac{y_j - y}{x_j - x} \ .$$

  The corresponding angle error is defined as

  $$\Delta\theta_{i,j} = |\theta_{i,j} - \bar{\theta}_{i,j}| \ ,$$

  where $\theta_{i,j}$ is given by Eq. (3).
- **Error distance**: For each testing location $l = (x, y)$ and corresponding estimated location $\hat{l} = (\hat{x}, \hat{y})$, the error distance is defined as

  $$\Delta d = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2} \ .$$

B. Experiment Results

1) Angle Estimation Accuracy: Fig. 6 shows the CDFs of angle error under the Sextant and the ImPos. We can see that the ImPos achieves a median angle error of 3.5 degrees, which is much smaller than the median angle error of 15.1 degrees achieved by the Sextant. In addition, the angle error is below 6.2 degrees for 80% of the cases under the ImPos which is significantly smaller than 19.8 degree under Sextant. These results clearly show that the ImPos can measure the angle much more accurately than the Sextant that directly uses compass readings as the directions to landmarks. We can also see that the maximum angle error under ImPos is 11.91 degree. This is because there is always some difference between the true angle drift α and the approximated angle γ in Fig. 2. In addition, the position of a landmark shown on an image is not always the center of all the matched features. We can also

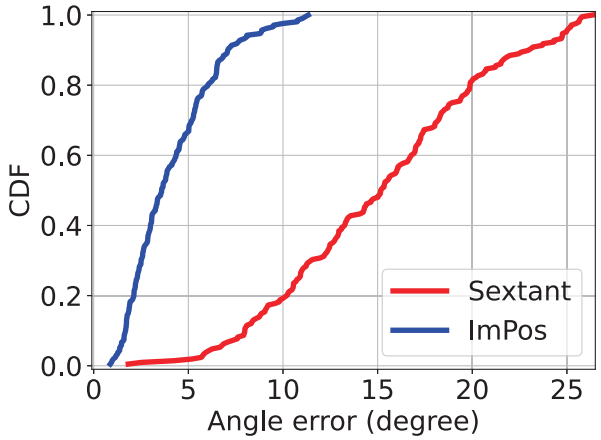

Fig. 6: Comparison of Sextant and ImPos in terms of angle error

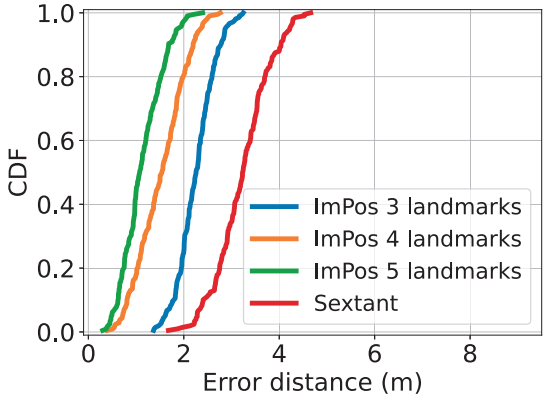

Fig. 7: The impact of the number of recognized landmarks on localization accuracy

see that the maximum angle error under the Sextant is 25.9 degree. This is also reasonable, because the angle of view of HUAWEI P30 is about 60 degrees. If a landmark appears on the edge of a query image, it would result in an angle error of up to 30 degree.

2) Positioning Accuracy : We now compare the ImPos and the Sextant in terms of error distance. Since the ImPos makes use of all recognized landmarks to estimate the user's location and the number of recognized landmarks affects its localization accuracy, we consider different numbers of landmarks for the ImPos. In particular, since we took five images at each testing location, the number of recognized landmarks is at least five. For each testing location, we enumerate all the combinations of 3, 4, and 5 recognized landmarks. For every combination of landmarks, we estimate the user's location and compute the corresponding error distance. As we can see from Fig. 7, ImPos achieves a median error distance of 2.33 m, 1.56 m, and 1.09 m with 3, 4, and 5 recognized landmarks, respectively, which is much lower than the 3.23 m under the Sextant system. These results indicate that the ImPos achieves a much

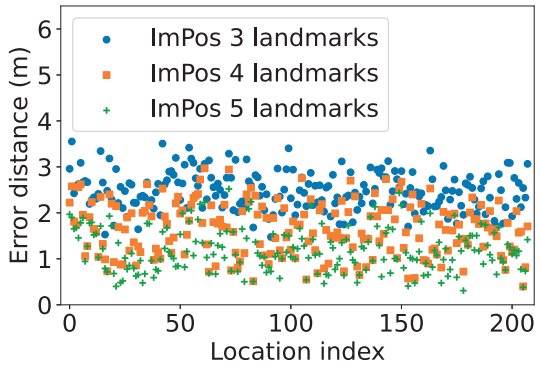

Fig. 8: Error distance at difference testing locations

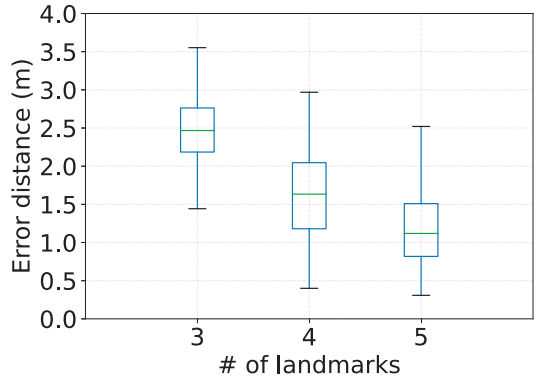

Fig. 9: Box-plot of error distance with different number of recognized landmarks

higher positioning accuracy than the Sextant because it can effectively make use of all recognized landmarks to improve its positioning accuracy.

Fig. 8 shows the average error distance for at every testing location with the number of recognized landmarks varying from 3 to 5. We can see that different testing locations have different error distances, which is expected. In addition, the average error distance decreases as the number of recognized landmarks increases for every testing location. Fig. 9 shows the box-plot of error distance with the number of recognized landmarks varying from 3 to 5. Once again, we can see that the ImPos achieves an median error distance of 2.33 m, 1.56 m, and 1.09 m with 3, 4, and 5 recognized landmarks respectively. These results further confirms the advantage of ImPos in fully utilizing all recognized landmarks to improve its localization accuracy.

3) Comparison of Two Optimization Problem Formulations: We also compare the error distances produced by the two optimization problem formulations introduced in Section III-C3. Fig. 10 shows CDFs of error distance produced by the two optimization problem formulations where the number of recognized landmarks varying from 4 to 5. We ignore the case of 3 recognized landmarks here because both formulations produce the same estimated location. We can see from Fig. 10 that

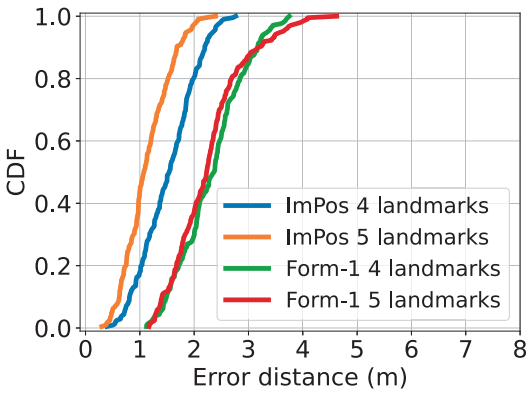

Fig. 10: The impact of different cost functions on localization accuracy

the error distance produced by the first problem formulation with unweighted cost function has a larger error distance than ImPos (i.e., the second formulation) in both cases. In addition, as the number of recognized landmarks increases from 4 to 5, the median error distance produced by the first problem formulation only reduces from 2.4 m to 2.3 m. In contrast, the median error distance under the ImPos reduces from 1.6 m to 1.2 m. These results show that the first formulation is unable to make full use of additional recognized landmarks to improve the positioning accuracy.

V. Conclusion

In this paper, we have introduced the design and evaluation of ImPos, a novel image-based IPS that achieves high positioning accuracy by accurately estimating the direction from a user to recognized landmarks through image analysis and fully utilizing all the recognized landmarks. Detailed experiment results confirm the significant advantages of ImPos over prior image-based solutions based on triangulation.

References

[1] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *IEEE International Conference on Computer Communications (INFOCOM'00)*, vol. 2, Tel Aviv, Israel, March 2000, pp. 775–784.

[2] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *International Conference on Mobile Systems, Applications, and Services (MobiSys'05)*, Seattle, WA, June 2005, pp. 205–218.

[3] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "Csi-based indoor localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, 2013.

[4] G. Fischer, B. Dietrich, and F. Winkler, "Bluetooth indoor localization system," 01 2004.

[5] G. Ruipeng, T. Yang, Y. Fan, L. Guojie, B. Kaigui, W. Yizhou, W. Tao, and L. Xiaoming, "Sextant: Towards ubiquitous indoor localization service by photo-taking of the environment," *IEEE Transactions on Mobile Computing*, vol. 15, no. 2, pp. 460–474, 2016.

[6] M. Liu, J. Du, Q. Zhou, Z. Cao, and Y. Liu, "Eyeloc: Smartphone vision-enabled plug-n-play indoor localization in large shopping malls," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5585–5598, 2021.

[7] S. S. Saab and Z. S. Nakad, "A standalone rfid indoor positioning system using passive tags," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1961–1970, 2011.

[8] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services(MobiSys'11)*, 2011, p. 155168.

[9] F. Hflinger, J. Hoppe, R. Zhang, A. Ens, L. Reindl, J. Wendeberg, and C. Schindelhauer, "Acoustic indoor-localization system for smart phones," in *IEEE 11th International Multi-Conference on Systems, Signals Devices (SSD'14)*, 2014, pp. 1–4.

[10] S. Zhu and X. Zhang, "Enabling high-precision visible light localization in today's buildings," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys'17, New York, NY, USA, 2017, p. 96108.

[11] "Global indoor positioning and navigation system market industry trends and forecast to 2028," 2021. [Online]. Available: https://www.databridgemarketresearch.com/reports/global-indoor-positioning-and-navigation-system-market

[12] I. Arnold, Z. Christopher, F. Jan-Michael, and B. Horst, "From structure-from-motion point clouds to fast location recognition," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2599–2606.

[13] C. Tommaso, G. Stuart, L. Nicholas A., V. Julien, S. Luigi Di, and T. Philip H. S., "On-the-fly adaptation of regression forests for online camera relocalisation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017, pp. 218–227.

[14] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "Inloc: Indoor visual localization with dense matching and view synthesis," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018.

[15] G. Lu and J. Song, "3d image-based indoor localization joint with wifi positioning," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval(ICMR'18)*, 2018, p. 465472.

[16] L. Nam Tuan and J. Yeong Min, "Photography trilateration indoor localization with image sensor communication," *Sensors*, 2019.

[17] H. Kawaji, K. Hatada, T. Yamasaki, and K. Aizawa, "Image-based indoor positioning system: Fast image matching using omnidirectional panoramic images," 01 2010.

[18] W. Ke, M. Lin, and T. Xuezhi, "An improvement algorithm on ransac for image-based indoor localization," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC'16)*, 2016, pp. 842–845.

[19] W. Martin, K. Moritz, and M. Chadly, "Indoor positioning using smartphone camera," in *2011 International Conference on Indoor Positioning and Indoor Navigation*, 2011, pp. 1–6.

[20] S. Hamed, V. Shahrokh, and S. Shahram, "A weighted knn epipolar geometry-based approach for vision-based indoor localization using smartphone cameras," in *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM'14)*, 2014, pp. 37–40.

[21] K. Guan, L. Ma, T. Xuezhi, and G. Shizeng, "Vision-based indoor localization approach based on surf and landmark," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC'16)*, 2016, pp. 655–659.

[22] Z. Yuanqing, S. Guobin, L. Liqun, Z. Chunshui, L. Mo, and Z. Feng, "Travi-navi: Self-deployable indoor navigation system," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2655–2669, 2017.

[23] N. Qun, L. Mingkuan, H. Suining, G. Chengying, S.-H. G. Chan, and L. Xiaonan, "Resource-efficient and automated image-based indoor localization," vol. 15, no. 2, 2019.

[24] D. Jiang, N. Marius, X. Yu, and Y.-J. Antti, "Vinav: A vision-based indoor navigation system for smartphones," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1461–1475, 2019.

[25] B. Atreyi and C. Heng Foh, "A practical path loss model for indoor wifi positioning enhancement," in *6th International Conference on Information, Communications Signal Processing*, 2007, pp. 1–5.

[26] Y. Wang, Q. Ye, J. Cheng, and L. Wang, "Rssi-based bluetooth indoor localization," in *11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN'15)*, 2015, pp. 165–171.

[27] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf:speeded up robust features," in *Computer Vision(ECCV'06*, 2006, pp. 404–417.